Lattice-based Signcryption with Equality Test in Standard Model

Huy Quoc Le[a,*], Dung Hoang Duong[a,*], Partha Sarathi Roy[a], Willy Susilo[a], Kazuhide Fukushima[b], Shinsaku Kiyomoto[b]

[a] *Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong NSW 2522, Australia*

[b] *Information Security Laboratory, KDDI Research, Inc.,*
2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502, Japan

Abstract

A signcryption, which is an integration of a public key encryption and a digital signature, can provide confidentiality and authenticity simultaneously. Additionally, a signcryption associated with equality test allows a third party (e.g., a cloud server) to check whether or not two ciphertexts are encrypted from the same message without knowing the message. This application plays an important role especially in computing on encrypted data. In this paper, we propose the first lattice-based signcryption scheme equipped with a solution to testing the message equality in the standard model. The proposed signcryption scheme is proven to be secure against insider attacks under the learning with errors assumption and the intractability of the short integer solution problem. As a by-product, we also show that some existing lattice-based signcryptions either is insecure or does not work correctly.

Keywords: Signcryption, equality test, standard model, learning with errors problem, short integer solution problem, insider attacks

1. Introduction

A signcryption scheme, first proposed by Zheng [1], simulatneously plays the roles of public key encryption and digital signature. A signcryption scheme therefore guarantees the confidentiality and the authenticity at the same time. On the other hand, signcryptions are designed aiming to be more efficient than the signing-then-encrypting approach in terms of cost.

[*] Corresponding author

Email addresses: `qhl576@uowmail.edu.au` (Huy Quoc Le), `hduong@uow.edu.au` (Dung Hoang Duong), `partha@uow.edu.au` (Partha Sarathi Roy), `wsusilo@uow.edu.au` (Willy Susilo), `ka-fukushima@kddi-research.jp` (Kazuhide Fukushima), `kiyomoto@kddi-research.jp` (Shinsaku Kiyomoto)

With a rapid increasing of amount of data, there are more and more personal users as well as organizations transferring their data to third-party service providers for outsourcing. In order to protect their data's sensitive information, data are usually encrypted. This fact requires service providers to have efficient methods of encrypted data management. Among such tools, equality test (ET) allows service providers to check whether or not different ciphertexts are generated on the same message even though service providers do not know what the message actually is. This augmented property has been realized in various public key encryption schemes (which are called PKEET), e.g., [2], [3], [4], [5], [6] with applications to internet-based personal health record systems [7], secure outsourced database managements [2] or in the identity-based encrytion schemes, e.g., [8], [9]. Of course, the equality test mechanism can also be realized in signcryptions which we call signcryption with equality test (SCET). SCET has also found some applications such as in securing messaging services [10], industrial Internet of Things [11].

Quantum computers are proven to be able to successfully break number-theoretic assumptions, such as the integer factorization problem or the discrete logarithm problem, which are currently the underlying hard problems for a plenty of cryptographic primitives [12]. Under the threat, research community has recently been paying more and more attention as well as resources to the so-called lattice-based cryptography, which based on hard lattice problems. With some advantages of easy implementation, provable hardness, lattice problems (e.g., learning with errors problem (LWE), short integer solution problem (SIS) are playing the role of underlying hard problems for numerous cryptographic primitives.

Related works. Malone-Lee and Mao [13] in 2003 presented a RSA-based signcryption scheme in the random oracle model. In the same year, Boyen [14] proposed a stringent security model for the schemes which he call *a joint identity-based signature/encryption* (IBSE), and presented an efficient IBSE construction, based on bilinear pairings. The work [4] by Lin et al. gave a generic SCET construction, extended from a generic PKEET construction. Wang et al. [10] presented a concrete SCET construction, which the authors call public key signcryption scheme with designated equality test on ciphertexts. However, the primitive in [10] is based on bilinear groups. More recently, Xiong et al. [11] propose the so-called heterogeneous SCET based on pairings which is claimed to be suitable for the sophisticated heterogeneous network of industrial Internet of Things. *So far, there has been no any SCET construction in the lattice setting.*

Regarding lattice-based signcryption constructions, there have been some works such as [15], [16], [17], [18] and [19]. Li et al. [15] built a lattice-based signcryption scheme that are only secure in the random oracle model (ROM). Gérard and Merckx [16] proposed a lattice-based signcryption scheme in the random oracle model (ROM) based on the ring learning with errors (RLWE) and a special version of the short integer solution (SIS) offering the indistinguishability under chosen plaintext attacks (IND-CPA) and existential unforgeability under chosen message attacks (EUF-CMA) security. Yang et al. [17] proposed an efficient lattice-based signcryption scheme in the standard model based on RLWE and the ideal short integer solution (ideal-SIS) assumption. The signcryption of [17] offers the IND-CCA2 security and and existential unforgeability under an adaptive chosen-message attack (EUF-aCMA). Lu et al. [18] proposed a lattice-based signcryption scheme without random oracles which is claimed to achieve the IND-CCA2 security and EUF-CMA security basing on the hardness of LWE and SIS. Sato and Shikata [19] constructed a lattice-based signcryption in the standard model of which security also based on LWE and SIS. We claim that, the IND-CCA2 security of the construction by Lu et al. [18] can be easily broken even by a CPA attack (see Section 7 below) while the work [19] has some serious flaws described as follows.

Description and flaws of the signcryption proposed by [19]. The work [19] exploits the gadget-based trapdoor mechanism proposed by [20], namely, the algorithms $\mathsf{GenTrap}$, Invert, $\mathsf{SampleD}$ (see Definition 5 and Lemma 7 below for further details). Then, for each receiver we generate the public key $pk_r = \mathbf{A}_r$, and the private key $sk_r = \mathbf{T}_r$ which is a $\mathbf{G}$-trapdoor for $\mathbf{A}_r$ with tag $\mathbf{H} = \mathbf{0}$, i.e., $\mathbf{A}_r\left[\begin{smallmatrix}\mathbf{T}_r\\ \mathbf{I}\end{smallmatrix}\right] = \mathbf{0} \pmod q$, where $\mathbf{G}$ is the gadget matrix specified in [20, Section 4]. The same thing $pk_s = \mathbf{A}_s, sk_r = \mathbf{T}_s$ is done for each sender except that $\mathbf{H} = \mathbf{I}$, i.e., $\mathbf{A}_s\left[\begin{smallmatrix}\mathbf{T}_s\\ \mathbf{I}\end{smallmatrix}\right] = \mathbf{G} \pmod q$.

For signcrypting a plaintext μ, one utilizes the Dual-Regev framework based on the LWE problem, i.e., one computes $(\overline{\mathbf{c}}_0)^t := \mathbf{s}^t\mathbf{A}_{r,\mathbf{t}} + (\mathbf{x}_0)^t$ and $(\overline{\mathbf{c}}_1)^t := \mathbf{s}^t\mathbf{U} + (\mathbf{x}_1)^t$. Here $\mathbf{U}$ is a uniform matrix while the matrix $\mathbf{A}_{r,\mathbf{t}}$ depends on $\mathbf{A}_r$ and some vector tag $\mathbf{t}$ which relates to the public key of the receiver pk_r, to the public key of the sender and to a random small vector $\mathbf{r}_e$. Then one signs on the tuple $(\mu|pk_r|\overline{ct})$, where $\overline{ct} = (\overline{\mathbf{c}}_0, \overline{\mathbf{c}}_1, \mathbf{r}_e)$, to get the signature $(\mathbf{e}, \mathbf{r}_s)$. To do that, one calculates a vector tag $\mathbf{h}$ using pk_s, pk_r, $(\mu|pk_r|\overline{ct})$ and a random small vector $\mathbf{r}_s$ and then signs on $\mathbf{h}$ using the same way as the signature scheme in [20, Section 6.2]. Finally, the signcryption finishes by computing $\mathbf{c}_0 := \overline{\mathbf{c}}_0 + \mathbf{r}_s$, $\mathbf{c}_1 := \overline{\mathbf{c}}_1 + \mu \cdot \lfloor q/2 \rfloor$ and outputting the ciphertext $ct = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{r}_e, \mathbf{e})$.

In the unsigncryption algorithm, one can use $\mathsf{SampleD}$ to obtain a matrix $\mathbf{E}$ such that $\mathbf{A}_{r,\mathbf{t}}\mathbf{E} = \mathbf{U} \bmod q$ which helps us to recover μ. However, in this algorithm, one also needs to verify whether the ciphertext is valid or not which needs to recover $\overline{ct} = (\overline{\mathbf{c}}_0, \overline{\mathbf{c}}_1, \mathbf{r}_e)$ from ct first. The way of [19] to do that is to run Invert on input $\mathbf{c}_0$ to get $\mathbf{r}_s$ and then compute $\overline{\mathbf{c}}_0 = \mathbf{c}_0 - \mathbf{r}_s$, $\overline{\mathbf{c}}_1 = \mathbf{c}_1 - \mu \cdot \lfloor q/2 \rfloor$. Unfortunately, we can see that this is not correct since Invert will actually output the sum $\mathbf{x}_0 + \mathbf{r}_s$ instead of $\mathbf{r}_s$.

One more flaw is that the dimensions of some matrices and some vectors in the signcryption of [19] do not match. Furthermore, the security proofs are quite unclear. For example, in the security proof for the strong unforgeability against insider attacks (i.e., MU-sUF-iCMA) in [19, Theorem 2], after showing that $\mathbf{A}_S \cdot \mathbf{z} = 0 \bmod q$, the authors do not prove why $\mathbf{z} \neq \mathbf{0}$.

Our contribution and technical overview. In this paper, we propose, for the first time, a lattice-based signcryption scheme possessing a capacity of equality test provably secure in the standard model. Moreover, since both [18] and [19] do not work correctly then our work without the equality test part can be considered as a lattice-based signcryption alternative to them. For our construction, we consider the multi-user setting and the insider security model in which there are multiple users and some of them could adversarially behave. We call such users the *internal users* or *insider attackers*. Such kind of attacker is stronger than external adversaries since they can know private information of other users in the setting. We show that our proposed scheme offers OW-iCCA1, IND-iCCA1 and UF-iCMA security against insider attacks relying on the hardness of decisional-LWE and SIS problems.

Our scheme is basically inspired from the work of Sato and Shikata [19] and the recent method of Duong et al. [5] for equality test. We have shown above that [19] does not work correctly. Fortunately, we successfully fix this error simply by in the signcryption algorithm setting $\mathbf{c}_0 := \overline{\mathbf{c}}_0$ instead of $\mathbf{c}_0 := \overline{\mathbf{c}}_0 + \mathbf{r}_s$ and outputting $ct = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{r}_e, \mathbf{r}_s, \mathbf{e})$ instead of $ct = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{r}_e, \mathbf{e})$. Also, to fix the dimension-related flaw in [19], we use the hash functions named H_1, H_3 to make dimensions match.

For equality test, we use an one-way hash function H and encrypt $H(\mu)$ (but do not sign) in the same way described above for the plaintext μ. This releases some things named $\mathbf{c}'_0$, $\overline{\mathbf{c}}'_1, \mathbf{r}'_e$, corresponding to $\mathbf{c}_0$, $\overline{\mathbf{c}}_1, \mathbf{r}_e$, for μ. Note that, the signing phase (which produces the signature $(\mathbf{e}, \mathbf{r}_s)$) now runs on input $(\mu | pk_r | \overline{ct})$ with $\overline{ct} = (\mathbf{c}_0, \overline{\mathbf{c}}_1, \mathbf{r}_e, \mathbf{c}'_0, \overline{\mathbf{c}}'_1, \mathbf{r}'_e)$ instead of $\overline{ct} = (\mathbf{c}_0, \overline{\mathbf{c}}_1, \mathbf{r}_e)$. Therefore, the final ciphertext for the proposed SCET is $ct = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{r}_e, \mathbf{r}_s, \mathbf{c}'_0, \overline{\mathbf{c}}'_1, \mathbf{r}'_e, \mathbf{e})$. Finally, two ciphertexts are proven to come from the same plaintext if we can recover the same hash value $H(\mu)$ from them without knowing the plaintext μ.

Furthermore, for the security proof, we also utilize the so-called abort-resistant hash functions defined by [21, Section 7] provided in Lemma 12. Also note that, the presence of $\mathbf{B}$, $\mathbf{r}_e, \mathbf{r}'_e, \mathbf{r}_s$ helps us to simulate the responses to the adversary's queries.

Organisation. In Section 2, we give a background of lattices. The framework of signcryption with equality test will be provided in Section 3. Section 4 is for our lattice-based signcryption construction. The security of the proposed scheme will be given in Section 5. Parameter setting will be done in Section 6. We demonstrate an attack against the IND-CPA of the signcryp-

Table 1: Some SC and SCET constructions based on hard lattice problems in the literature.

Works	Assumption	Security Level	Security Model	with ET	Insider Attacks
Li [15]	LWE & SIS	IND-CCA2 SUF-CMA	ROM	×	×
Lu [18]	Not secure even with IND-CPA (see Section 7)				
Sato [19]	Does not work correctly				
Gérard [16]	RLWE & SIS	IND-CPA EUF-CMA	ROM	×	×
Yang [17]	RLWE & ideal-SIS	IND-CCA2 EUF-aCMA	SDM	×	×
This work	LWE & SIS	IND-iCCA1 OW-iCCA1 SUF-iCMA	SDM	✓	✓

Category	Size
Public key per user	$2mn \cdot \mathbb{Z}_q$
Secret key per user	$2\overline{m}nk \cdot D_{\sigma_1}$
Ciphertext	$3m \cdot D_{\alpha q} + (m + nk) \cdot D_{\sigma_2} + 2(m + \ell) \cdot \mathbb{Z}_q$

Table 2: Sizes of our SCET. Here $a \cdot S$ means a elements in the domain S. For example $3m \cdot D_{\alpha q}$ indicates that there are $3m$ elements each of which sampled from $D_{\alpha q}$.

tion construction proposed by Lu et al. [18] in Section 7. In Section 8, we make some conclusions on our work.

2. Preliminaries

Throughout this work, the norm $\|\mathbf{S}\|$ of a set of vectors $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_n\}$ is computed as $\max_{i \in [n]} \|\mathbf{s}_i\|$.

Lattices. Given a matrix $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_m] \in \mathbb{R}^{n \times m}$ of m linearly independent vectors, the set $\mathcal{L}(\mathbf{B}) := \{\sum_{i \in [m]} \mathbf{b}_i z_i : z_i \in \mathbb{Z}\}$ is called a lattice of basis $\mathbf{B}$. In this work, we focus on the so-called q-ary lattices: $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{0} \pmod{q}\},$, and $\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathrm{A}\mathbf{e} = \mathbf{u} \text{ (mod q)}\}$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}^{n \times m}$ is a random matrix. Note that, if $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$ then $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \mathbf{t} + \Lambda_q^{\perp}(\mathbf{A})$.

The *first minimum* of a lattice $\mathcal{L}$ is defined as $\lambda_1(\mathcal{L}) := \min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{v}\|$. The *i-th minimum* of a lattice $\mathcal{L}$ of dimension n is denoted by and defined as $\lambda_i(\mathcal{L}) := \min\{r : \dim(\mathrm{span}(\mathcal{L} \cap \mathcal{B}_n(0, r))) \geq i\}$, where $\mathcal{B}_n(0, r) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq r\}$. The SIVP_γ and GapSVP_γ are assumed to be the worst case hard problems in lattices. Given $\mathbf{A}$ to be a basis of a lattice $\mathcal{L}(\mathbf{A})$ and a positive real number d, the first problem requires to find a set of n linearly independent lattice vectors $\mathbf{S} \subset \mathcal{L}(\mathbf{A})$ such that $\|\mathbf{S}\| \leq \gamma\lambda_n(\mathbf{A})$, while the second one asks to decide if $\lambda_1(\mathcal{L}(\mathbf{A})) \leq d$ or $\lambda_1(\mathcal{L}(\mathbf{A})) > \gamma d$.

Gaussians. Let $m \geq 1$, a vector $\mathbf{c} \in \mathbb{R}^m$ and a positive parameter s, for $\mathbf{x} \in \mathbb{R}^m$ define $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{c}\|^2/s^2)$. The continuous Gaussian distribution on $\mathbb{R}^m$ with mean $\mathbf{c}$ and with width parameter s is proportional to $\rho_{s,\mathbf{c}}(\mathbf{x})$. Let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ be the additive group of real numbers modulo 1. Given $\alpha > 0$ and $m = 1$, we denote by Ψ_α the continuous Gaussian distribution on $\mathbb{T}$ of mean 0 and width parameter $s := \alpha$. Remind that, this Gaussian distribution has standard deviation of $\sigma = \alpha/\sqrt{2\pi}$.

Definition 1 (Discretized Gaussian). The discretized Gaussian distribution $\widetilde{\Psi}_{\alpha q}$ is defined by sampling $X \leftarrow \Psi_\alpha$ then outputting $\lfloor q \cdot X \rceil \bmod q$.

In particular, we can define a Gaussian distribution over a subset of $\mathbb{Z}^m$, hence over any integer lattices in $\mathbb{Z}^m$.

Definition 2 (Discrete Gaussian). Let m be a positive integer, $\Lambda \subset \mathbb{Z}^m$ be any subset, a vector $\mathbf{c} \in \mathbb{R}^m$ and a positive parameter s, define $\rho_{s,\mathbf{c}}(\Lambda) := \sum_{\mathbf{x}\in\Lambda} \rho_{s,\mathbf{c}}(\mathbf{x})$. The discrete Gaussian distribution over Λ centered at $\mathbf{c} \in \mathbb{Z}^m$ with width parameter s is defined by: $\forall \mathbf{x} \in \Lambda$, $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) := \rho_{s,\mathbf{c}}(\mathbf{x})/\rho_{s,\mathbf{c}}(\Lambda)$. If $\mathbf{c} = \mathbf{0}$, we just simply write ρ_s, $D_{\Lambda,s}$. If $\Lambda = \mathbb{Z}$, we can write $D_{\Lambda,s}$ as D_s.

Note that in Definition 2, Λ can be a lattice over $\mathbb{Z}^m$. The following lemma shows the min-entropy of a discrete Gaussian.

Lemma 1. *[22, Lemma 2.1] Let $\Lambda \subset \mathbb{R}^n$ be a lattice and $s \geq 2\eta_\epsilon(\Lambda)$ for some $\epsilon \in (0,1)$. Then for any $\mathbf{c} \in \mathbb{R}^n$ and any $\mathbf{y} \in \Lambda + \mathbf{c}$, $\Pr[\mathbf{x} \leftarrow D_{\Lambda+\mathbf{c},s} : \mathbf{x} = \mathbf{y}] \leq 2^{-n} \cdot \frac{1+\epsilon}{1-\epsilon}$.*

Lemma 2. *[23, Lemma 4.4] Let $q > 2$ and let $\mathbf{A}$ be a matrix in $\mathbb{Z}_q^{n\times m}$ with $m > n$. Let $\mathbf{T}_A$ be a basis for $\Lambda_q^\perp(\mathbf{A})$. Then, for $s \geq \|\widetilde{\mathbf{T}_A}\| \cdot \omega(\sqrt{\log n})$,*

$$\Pr[\mathbf{x} \leftarrow \mathcal{D}_{\Lambda_q^{\mathbf{u}}(A),s} \;:\; \|\mathbf{x}\| > s\sqrt{m}] \leq \mathsf{negl}(n).$$

Lerning with Errors problem (LWE) and Short interger Solutions problem (SIS). Let n and $q \geq 2$ be positive integers and χ be a distribution on $\mathbb{Z}_q$. Given a vector $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, we define an LWE distribution $\mathcal{L}_{\mathbf{s},\chi}$ on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ as follows: first sample uniformly at random $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$, then draw according to χ an error term e, and finally output the pair $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s}\rangle + e \pmod q)$.

Definition 3 (LWE, [24]). The decisional-LWE problem ($\mathsf{dLWE}_{n,q,\chi}$) asks to distinguish a pair $(\mathbf{a}, b) \leftarrow \mathcal{L}_{\mathbf{s},\chi}$ from a pair $(\mathbf{a}, b) \xleftarrow{\$} \mathbb{Z}_q^n \times \mathbb{Z}_q$.

In the case that $\chi = \widetilde{\Psi}_{\alpha q}$, we instead use the notations $\mathsf{dLWE}_{n,q,\alpha}$ and $\mathsf{sLWE}_{n,q,\alpha}$, and generally mention them as the $\mathsf{LWE}_{n,q,\alpha}$. Regarding the hardness of LWE, we have the following result:

Theorem 3 ([25, Theorem 2.16]). *Let $n, q \geq 1$ be integers and let $\alpha \in (0,1)$ be such that $\alpha q \geq 2\sqrt{n}$. Then there exists a quantum reduction from worst-case $\mathsf{GapSVP}_{\widetilde{O}(n/\alpha)}$ to $\mathsf{LWE}_{n,q,\alpha}$. If in addition $q \geq 2^{n/2}$ then there is also a classical reduction between those problems.*

Definition 4 (SIS). For an integer q, a random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and a positive real number β, the short integer problem $\mathsf{SIS}_{q,n,m,\beta}$ is to find a non-zero vector $\mathbf{z} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ satisfying that $\mathbf{Az} = \mathbf{0} \pmod{q}$ and $\|\mathbf{z}\| \leq \beta$.

It is shown in [23] and then in [26] that for large enough q, solving SIS is as hard as solving SIVP problem. Formally,

Lemma 4 ([26, Proposition 5.7]). *For any poly-bounded m, and $\beta = \mathsf{poly}(n)$, and for any prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$ the average-case problem $\mathsf{SIS}_{q,n,m,\beta}$ is as hard as SIVP_γ in the worst-case to within certain $\gamma = \widetilde{O}(\beta\sqrt{n})$ factor.*

The following lemma gives a condition for which the $\mathsf{SIS}_{n,m,q,\beta}$ problem has a solution.

Lemma 5 ([23, Lemma 5.2]). *For any q, $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, and $\beta \geq \sqrt{m} q^{n/m}$, the $\mathsf{SIS}_{n,m,q,\beta}$ admits a solution.*

Gadget-based Trapdoor. We will recall the notion of $\mathbf{G}$-trapdoor and its related algorithms.

Definition 5 (G-trapdoors,[20, Definittion 5.2]). Let n, q, m, k be positive integers and $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{G} \in \mathbb{Z}_q^{n\times nk}$ be matrices with $m \geq nk$. Let $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ be some invertible matrix. The $\mathbf{G}$-trapdoor for $\mathbf{A}$ with tag $\mathbf{H}$ is a matrix $\mathbf{R} \in \mathbb{Z}^{(m-nk)\times nk}$ such that $\mathbf{A}\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}_{nk}\end{smallmatrix}\right] = \mathbf{HG} \pmod{q}$.

The largest singular value $s_1(\mathbf{R})$ is used to measure the quality of a $\mathbf{G}$-trapdoor $\mathbf{R}$ by its. Note that, $s_1(\mathbf{R})$ is essentially small as claimed in the following lemma.

Lemma 6 ([20, Lemma 2.9]). *Let $D_\sigma^{n\times m}$ be a discrete Gaussian distribution with parameter σ and $\mathbf{R} \leftarrow D_\sigma^{n\times m}$. Then with overwhelming probability $s_1(\mathbf{R}) \leq \sigma \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{n} + \sqrt{m})$.*

Let $k = \lceil \log_2 q \rceil$, and $\mathbf{g}^t = (1, 2, 4, ..., 2^{k-1}) \in \mathbf{Z}_q^k$. We will be working with $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^t \in \mathbb{Z}_q^{n\times nk}$, where $\otimes$ denotes the tensor product. Further details can be found in [20]. We will exploit the following algorithms for the proposed SCET construction.

Lemma 7. *Let $q \geq 2, \overline{m} \geq 1$, $k = \lceil \log_2 q \rceil$, and $m = \overline{m} + nk = O(n \log q)$.*

1. $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{GenTrap}(n, \overline{m}, q, \sigma)$ *[20, Algorithm 1]: On input integer $n, \overline{m}, q, \sigma$,* $\mathsf{GenTrap}$ *chooses a uniform matrix $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n \times \overline{m}}$ and a matrix $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$, then outputs a random matrix $\mathbf{A} = \left[\overline{\mathbf{A}} | \mathbf{H}\mathbf{G} - \overline{\mathbf{A}}\mathbf{R}\right]$ and a $\mathbf{G}$-trapdoor $\mathbf{R} \sim D_\sigma^{\overline{m} \times nk}$ with tag $\mathbf{H}$. The condition for Gaussian parameter σ is that for any $\epsilon \in (0, 1)$, $\sigma \geq \eta_\epsilon(\mathbb{Z})$, i.e., $\sigma \geq \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}}$. Moreover, there exists $\epsilon = \epsilon(n)$ negligible for which $\sigma \geq \omega(\sqrt{\log n})$. Note also that, $s_1(\mathbf{R}) \leq \sigma \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk})$ by Lemma 6.*

2. $\mathbf{e} \leftarrow \mathsf{SampleD}(\mathbf{A}, \mathbf{R}, \mathbf{H}, \mathbf{u}, \sigma)$ *[20, Algorithm 3]: On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times (\overline{m} + nk)}$ and its $\mathbf{G}$-trapdoor $\mathbf{R} \in \mathbb{Z}^{\overline{m} \times nk}$, an invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$, a uniform vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$ and a Gaussian parameter σ,* $\mathsf{SampleD}$ *outputs a vector $\mathbf{e} \in \mathbb{Z}^{m+nk} \sim D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}), \sigma}$. The condition for σ is that $\sigma \geq \sqrt{7(s_1(\mathbf{R})^2 + 1)} \cdot \omega(\sqrt{\log n})$ (see [20, Section 5.4]).*

3. $(\mathbf{s}, \mathbf{e}) \leftarrow \mathsf{Invert}(\mathbf{R}, \mathbf{A}, \mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t)$ *[20, Algorithm 2]: On input a uniform matrix $\mathbf{A}$ and its $\mathbf{G}$-trapdoor $\mathbf{R}$, and a vector $\mathbf{b}$ such that $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$,* Invert *returns ($\mathbf{s}$ and $\mathbf{e}$). Note that if $\boldsymbol{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$ and $1/\alpha \geq 2\sqrt{5(s_1(\mathbf{R})^2 + 1)} \cdot \omega(\sqrt{\log n})$ then* Invert *succeeds with overwhelming probability over the choice of $\mathbf{e}$ (see [20, Theorem 5.4]).*

We adapt Lemma 6 in [27] for scalar tags (i.e., $\mathbf{H} = x \cdot \mathbf{I}_n$ for some $x \in \mathbb{Z}_q \setminus \{0\}$) to get the following lemma which will be helpful for the security analysis in Section 5:

Lemma 8 (Adapted from [27, Lemma 6]). *For $i = 0, \cdots, n$, let $\mathbf{T}^{(i)}$ be $\mathbf{G}$-trapdoor for $[\mathbf{A}|\mathbf{A}^{(i)}] \in \mathbb{Z}_q^{n \times (m-k)} \times \mathbb{Z}_q^{n \times k}$ with tag $\mathbf{H}^{(i)} = x_i\mathbf{I}_n$ for some $x_i \in \mathbb{Z}_q \setminus \{0\}$. Then any linear combination $\mathbf{T} = \sum_{i=1}^n h_i\mathbf{T}^{(i)}$ with $h_i \in \mathbb{Z}_q$ is a $\mathbf{G}$-trapdoor for $[\mathbf{A}| \sum_{i=1}^n h_i\mathbf{A}^{(i)}]$ with tag $\mathbf{H} = \sum_{i=1}^n h_i\mathbf{H}^{(i)} = (\sum_{i=1}^n h_i x_i)\mathbf{I}_n \neq \mathbf{0}$.*

3. Framework of Signcryption Scheme with Equality Test

There are two settings for an SCET scheme depending on the number of users joining the scheme [28]. While in two-user setting, there are only one receiver and only one sender, the multi-user setting involves with multiple receivers and senders. In this setting, it is supposed that the attacker knows all public keys of all receivers and of all senders when he accesses the communication channel between the target sender and the target receiver. See [29, Subsection 1.3] for more details.

From now on, we suppose that in a SCET scheme, there are N receivers and M senders. We also use r (resp., s) to represent the index of a receiver (resp., a sender).

3.1. Syntax

A SCET is a tuple of algorithms Setup, KGr, KGs, SC and USC, Tag, and Test which is described as follows:

- $\mathsf{Setup}(1^\lambda)$ is a probabilistic polynomial time (PPT) algorithm that takes as input a security parameter λ to output a set of public parameters pp.

- $\mathsf{KGr}(pp)$ (resp., $\mathsf{KGs}(pp)$) is a PPT algorithm that on input the set of public parameters pp, outputs a public key pk_r and a private key sk_r for a receiver $\mathcal{R}$ (resp., a public key pk_s and a private key sk_s for a sender $\mathcal{S}$).

- $\mathsf{SC}(pk_r, sk_s, \mu)$ is a PPT algorithm takes as input a public key pk_r of a receiver, a private key sk_s of a sender and a message μ in the message space $\mathcal{M}$ to output a ciphertext ct.

- $\mathsf{USC}(sk_r, pk_s, ct)$ is a deterministic polynomial time (DPT) algorithm takes as input the private key sk_r of a receiver, a public key pk_s of a sender and a ciphertext ct to output a message μ or an invalid $\perp$.

- $\mathsf{Tag}(sk_r)$ is a DPT algorithm that on input a private key sk_r of a receiver to output a tag tg_r.

- $\mathsf{Test}(tg_1, ct_1, tg_2, ct_2)$ is a DPT algorithm that takes as input two pairs of tag/ciphertext (tg_1, ct_1), (tg_2, ct_2) to output 1 if ct_1 and ct_2 are generated on the same message or 0 otherwise.

3.2. Correctness

Let λ be any security parameter. For any $pp \leftarrow \mathsf{Setup}(1^\lambda)$, $(pk_r, sk_r) \leftarrow \mathsf{KGr}(pp)$, $(pk_s, sk_s) \leftarrow \mathsf{KGs}(pp)$, $(pk_{r_1}, sk_{r_1}) \leftarrow \mathsf{KGr}(pp)$, $(pk_{s_1}, sk_{s_1}) \leftarrow \mathsf{KGs}(pp)$, $(pk_{r_2}, sk_{r_2}) \leftarrow \mathsf{KGr}(pp)$, $(pk_{s_2}, sk_{s_2}) \leftarrow \mathsf{KGs}(pp)$, $tg_1 \leftarrow \mathsf{Tag}(sk_{r_1})$ and $tg_2 \leftarrow \mathsf{Tag}(sk_{r_2})$, any ciphertexts ct_1 and ct_2, and any message $\mu \in \mathcal{M}$, the correctness for an SCET scheme requires all the following to hold:

1. $\Pr[\mu = \mathsf{USC}(sk_r, pk_s, \mathsf{SC}(pk_r, sk_s, \mu))] = 1 - \mathsf{negl}(\lambda)$.
   This says that given a valid ciphertext on a message, the unsigncryption algorithm succeeds in recovering that message with overwhelming probability.

2. If $\mathsf{USC}(sk_{r_1}, pk_{s_1}, ct_1) = \mathsf{USC}(sk_{r_2}, pk_{s_2}, ct_2) \neq \perp$, then

   $$\Pr[\mathsf{Test}(tg_1, ct_1, tg_2, ct_2) = 1] = 1 - \mathsf{negl}(\lambda).$$

   This says that if two ciphertexts are from the same message then the equality test algorithm returns 1 with overwhelming probability.

3. If $\mathsf{USC}(sk_{r_1}, pk_{s_1}, ct_1) \neq \mathsf{USC}(sk_{r_2}, pk_{s_2}, ct_2)$, then

$$\Pr[\mathsf{Test}(tg_1, ct_1, tg_2, ct_2) = 1] = \mathsf{negl}(\lambda).$$

This says that if two ciphertexts are generated on two different messages then the equality test algorithm returns 1 with negligible probability.

3.3. Security

We categorize the security for SCET into the outsider and insider securities. In the outsider security setting, an external adversary cannot know private information of users but public information (e.g., public system parameters and public keys). In contrast, in the insider security setting, an internal adversary can know some private keys of other users hence he is stronger than any external adversaries.

We also consider three types of adversary against a SCET scheme. Remark that, all of them can be internal adavesaries, i.e., insider attackers. These types of adversary and their behaviors will be detailed in the following definitions and games.

- *Type 1 adversary* is supposed to know the target receiver, but does not have the tag of the target receiver and his goal is to guess which message between two options that is used in the signcryption algorithm to produce the challenge ciphertext. As an insider attacker, he can also know the target sender's public and private keys. The Type 1 adversary corresponds to the IND-iCCA1 game.

- *Type 2 adversar*y is supposed to know the target receiver and can perform equality tests on any ciphertexts and his goal is to recover the message corresponding to the challenge ciphertext. As an insider attacker, he can also know the target sender's public and private keys. The Type 2 adversary corresponds to the OW-iCCA1 game.

- *Type 3 adversary* is supposed to know the target sender and his goal is to try to forge at least one valid ciphertext. As an insider attacker, he can also know the target receiver's public and private keys. The Type 3 adversary corresponds to the UF-iCMA game.

Definition 6 (IND-iCCA1). An SCET scheme is IND-iCCA1 secure if the advantage of any PPT adversary $\mathcal{A}_1$ playing the $\mathsf{INDCCA1}_{SCET}^{\mathcal{A}_1}$ game is negligible: $\mathsf{Adv}_{\mathcal{A}_1}^{\text{IND-iCCA1}}(\lambda) := |\Pr[\mathsf{INDCCA1}_{SCET}^{\mathcal{A}_1} \Rightarrow 1] - 1/2| \leq \mathsf{negl}(\lambda)$.

The $\mathsf{INDCCA1}_{SCET}^{\mathcal{A}_1}$ game is defined as follows:

Setup. The challenger $\mathcal{C}$ first runs $\mathsf{Setup}(1^\lambda)$ to have the set of public parameters pp. Then $\mathcal{C}$ runs $\mathsf{KGr}(pp)$ to get (pk_r, sk_r) for $r \in [N]$, and

$\mathsf{KGs}(pp)$ to get (pk_s, sk_s) for $s \in [M]$, then sends $(pp, \{pk_r\}_{r\in[N]}), \{pk_s\}_{s\in[M]})$ to the adversary $\mathcal{A}_1$. Let $r^* \in [N]$ be the index of the target receiver.

Phase 1. $\mathcal{A}_1$ adaptively makes a polynomially bounded number of the following queries:

- Private key query PKQ(r): If $r = r^*$, the challenger $\mathcal{C}$ rejects the query. Otherwise, $\mathcal{C}$ returns the private key sk_r of the receiver $\mathcal{R}_r$ to $\mathcal{A}_1$.

- Signcryption query SCQ(r, s, μ): The challenger $\mathcal{C}$ sends the output ct of $\mathsf{SC}(pk_r,\ sk_s, \mu)$ back to $\mathcal{A}_1$.

- Unsigncryption query USQ(r, s, ct): $\mathcal{C}$ sends the output of $\mathsf{USC}(sk_r, pk_s,$ $ct)$ back to $\mathcal{A}_1$.

- Tag query TGQ(r): If $r = r^*$, the challenger $\mathcal{C}$ rejects the query. Otherwise, $\mathcal{C}$ in turn sends the output tg_r of $\mathsf{Tag}(sk_r)$ back to $\mathcal{A}_1$.

Challenge. $\mathcal{A}_1$ submits two messages μ_0^*, μ_1^* together with the target sender's keys (pk_{s^*}, sk_{s^*}). The challenger $\mathcal{C}$ then chooses uniformly at random a bit $b \in \{0, 1\}$ and returns the challenge ciphertext $ct^* \leftarrow \mathsf{SC}(pk_{r^*}, sk_{s^*}, \mu_b^*)$ to $\mathcal{A}_1$.

Phase 2. $\mathcal{A}_1$ queries the oracles again as in **Phase 1** with a restriction that $\mathcal{A}_1$ is not allowed to make the query PKQ(r^*) and all unsigncryption queries USQ($r, s,\ ct$).

Output. $\mathcal{A}_1$ outputs a bit $b' \in \{0, 1\}$. He wins the game if $b' = b$.

Definition 7 (OW-iCCA1). The scheme SCET is OW-iCCA1 secure if the advantage of any PPT adversary $\mathcal{A}_2$ playing the $\mathsf{OWCCA1}^{\mathcal{A}_2}_{SCET}$ game is negligible: $\mathsf{Adv}^{\text{OW-iCCA1}}_{\mathcal{A}_2}(\lambda) := \Pr[\mathsf{OWCCA1}^{\mathcal{A}_2}_{SCET} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$.

The $\mathsf{OWCCA1}^{\mathcal{A}_2}_{SCET}$ game is defined as follows:

Setup. The challenger $\mathcal{C}$ first runs $\mathsf{Setup}(1^\lambda)$ to have the set of public parameters pp and then runs $\mathsf{KGr}(pp)$ to get (pk_r, sk_r) for $r \in [N]$, and $\mathsf{KGs}(pp)$ to get (pk_s, sk_s) for $s \in [M]$, then sends $(pp, \{pk_r\}_{r\in[N]}), \{pk_s\}_{s\in[M]})$ to the adversary $\mathcal{A}_2$. Let $r^* \in [N]$ be the index of the target receiver.

Phase 1. $\mathcal{A}_2$ adaptively makes polynomially bounded number of the following queries:

- Private key query PKQ(r): If $r = r^*$, $\mathcal{C}$ rejects the query. Otherwise, $\mathcal{C}$ returns the private key sk_r of the receiver $\mathcal{R}_r$.

- Signcryption query SCQ(r, s, μ): $\mathcal{C}$ returns the output ct of $\mathsf{SC}(pk_r, sk_s, \mu)$.

- Unsigncryption query USQ(r, s, ct): $\mathcal{C}$ sends the output of $\mathsf{USC}(sk_r, pk_s,$ $ct)$ back to $\mathcal{A}_2$.

- Tag query TGQ(r): $\mathcal{C}$ returns the output tg_r of $\mathsf{Tag}(sk_r)$ (even when $r = r^*$).

Challenge. $\mathcal{A}_2$ submits the target sender's keys (pk_{s^*}, sk_{s^*}), $\mathcal{C}$ chooses a random message $\mu^* \in \mathcal{M}$ and returns the challenge ciphertext $ct^* \leftarrow \mathsf{SC}(pk_{r^*}, sk_{s^*}, \mu^*)$ to $\mathcal{A}_2$.

Phase 2. $\mathcal{A}_2$ queries the oracles again as in **Phase 1** with a restriction that $\mathcal{A}$ is not allowed to make the query PKQ(r^*) and all unsigncryption queries USQ(r, s, ct).

Output. $\mathcal{A}_2$ outputs μ'^*. He wins the game if $\mu'^* = \mu^*$.

Remark 1. One should be aware that there is no any reduction from IND-iCCA1 to OW-iCCA1 because the OW-iCCA1 adversary is allowed to know the tag of the target receiver, whilst the IND-iCCA1 is not.

Definition 8 (UF-iCMA). The scheme SCET is UF-iCMA secure if the advantage of any PPT adversary $\mathcal{A}_3$ playing the $\mathsf{UFCMA}_{SCET}^{\mathcal{A}_3}$ game is negligible: $\mathsf{Adv}_{\mathcal{A}_3}^{\text{UF-iCMA}}(\lambda) := \Pr[\mathsf{UFCMA}_{SCET}^{\mathcal{A}_3} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$.

The $\mathsf{UFCMA}_{SCET}^{\mathcal{A}_3}$ game is defined as follows:

Setup. The challenger $\mathcal{C}$ first runs $\mathsf{Setup}(1^\lambda)$ to have the set of public parameters pp and then runs $\mathsf{KGr}(pp)$ to get (pk_r, sk_r) for $r \in [N]$, and $\mathsf{KGs}(pp)$ to get (pk_s, sk_s) for $s \in [M]$, then sends $(pp, \{pk_r\}_{r\in[N]}), \{pk_s\}_{s\in[M]})$ to the adversary $\mathcal{A}_3$. Let $s^* \in [M]$ be the index of the target sender.

Queries. $\mathcal{A}_3$ adaptively makes polynomially bounded number of the following queries:

- Private key query PKQ(s): If $s = s^*$, $\mathcal{C}$ rejects the query. Otherwise, $\mathcal{C}$ returns the private key sk_s of the sender $\mathcal{S}_s$ to $\mathcal{A}_3$.

- Signcryption query SCQ(r, s, μ): $\mathcal{C}$ returns the output ct of $\mathsf{SC}(pk_r, sk_s, \mu)$ back to $\mathcal{A}_3$.

- Unsigncryption query USQ(r, s, ct): $\mathcal{C}$ returns the output of $\mathsf{USC}(sk_r, pk_s, ct)$.

- Tag query TGQ(r): $\mathcal{C}$ returns the output tg_r of $\mathsf{Tag}(sk_r)$.

Forge. $\mathcal{A}_3$ outputs an index r^* of some receiver and a ciphertext ct^* on a message μ^*, where ct^* must not be the output of any query $SCQ(r, s, \mu)$ in the query phase. He wins the game if $\mathsf{USC}(sk_{r^*}, pk_{s^*}, ct^*) \neq \perp$. Note that, if μ^* is not the same as the messages queried previously, the SCET scheme is called EUF-iCMA (i.e., existential unforgeability). If μ^* is one of the messages queried previously but $(\mu^*, ct^*) \neq (\mu, ct)$ for all (μ, ct) that was queried previously, then the SCET scheme is called SUF-iCMA (i.e., strong unforgeability).

4. Our Construction

In this section, we describe a lattice-based signcryption with equality test, named SCET. The proposed SCET signcryption consists of algorithms Setup, KG, SC, USC, Tag and Test. We also consider lattice-based collision-resistant hash functions indicated by a uniform matrix $\mathbf{W}$ defined as $f_{\mathbf{W}}(\mathbf{x}) := \mathbf{Wx} \bmod q$ (cf. [23]).

Setup(1^n): On input a security parameter n, perform the following:

1. Set parameters $n, q, \overline{m}$, ℓ, N, M, α, σ_1, σ_2, $k = \lceil \log q \rceil$, $m = \overline{m} + nk$ as in Section 6.
2. Samples randomly and independently matrices $\mathbf{C}_0, \cdots, \mathbf{C}_n, \mathbf{C}'_0, \cdots, \mathbf{C}'_n \in \mathbb{Z}_q^{n\times nk}$, $\mathbf{B}, \mathbf{B}' \in \mathbb{Z}_q^{n\times m}$, $\mathbf{U}, \mathbf{U}' \in \mathbb{Z}_q^{n\times \ell}$.
3. Samples randomly vector $\mathbf{u} \in \mathbb{Z}_q^n$.
4. One-way hash function $H : \{0,1\}^\ell \to \{0,1\}^\ell$, collision-resistant hash functions $H_1 : \mathbb{Z}_q^{n\times m} \to \mathbb{Z}_q^{\overline{m}}$, H_2 is a full-rank differences (FRD) encoding [1] and a universal hash function $H_3 : \{0,1\}^* \to \mathbb{Z}_q^{\overline{m}}$.
5. A plaintext (message) space $\mathcal{M} = \{0,1\}^\ell$.
6. Return $pp = \{n, q, k, \overline{m}, m, \ell, \alpha, \sigma_1, \sigma_2, N, M, \mathcal{M}, (\mathbf{C}_i, \mathbf{C}'_i)_{i=0}^n, H, H_1, H_2, H_3\}$ as the set of public parameters.

KG(pp): On input the public parameters pp, do the following:

1. For each receiver $r \in [N]$, generate $\overline{\mathbf{A}}_r, \overline{\mathbf{A}}'_r \xleftarrow{\$} \mathbb{Z}_q^{n\times\overline{m}}$, $\mathbf{T}_r, \mathbf{T}'_r \leftarrow D_{\mathbb{Z}^{\overline{m}\times nk},\sigma_1}$ and then set $\mathbf{A}_r = [\overline{\mathbf{A}}_r | -\overline{\mathbf{A}}_r \cdot \mathbf{T}_r] \in \mathbb{Z}_q^{n\times m}$, $\mathbf{A}'_r = [\overline{\mathbf{A}}'_r | -\overline{\mathbf{A}}'_r \cdot \mathbf{T}'_r] \in \mathbb{Z}_q^{n\times m}$
2. Similarly, for each sender $s \in [M]$, generate $\overline{\mathbf{A}}_s, \overline{\mathbf{A}}'_s \xleftarrow{\$} \mathbb{Z}_q^{n\times\overline{m}}$, $\mathbf{T}_s, \mathbf{T}'_s \leftarrow D_{\mathbb{Z}^{\overline{m}\times nk},\sigma_1}$ and then set $\mathbf{A}_s = [\overline{\mathbf{A}}_s | \mathbf{G} - \overline{\mathbf{A}}_s \cdot \mathbf{T}_s] \in \mathbb{Z}_q^{n\times m}$, $\mathbf{A}'_s = [\overline{\mathbf{A}}'_s | \mathbf{G} - \overline{\mathbf{A}}'_s \cdot \mathbf{T}'_s] \in \mathbb{Z}_q^{n\times m}$
3. Return $pk_r = (\mathbf{A}_r, \mathbf{A}'_r)$, and $sk_r = (\mathbf{T}_r, \mathbf{T}'_r)$ as public key and private key for a receiver $\mathcal{R}$ of index r, $pk_s = (\mathbf{A}_s, \mathbf{A}'_s)$, and $sk_s = (\mathbf{T}_s, \mathbf{T}'_s)$ as public key and private key for a sender $\mathcal{S}$ of index s.

SC(pk_r, sk_s, μ): On input a receiver's public key $pk_r = (\mathbf{A}_r, \mathbf{A}'_r)$, a sender's private key $sk_s = (\mathbf{T}_s, \mathbf{T}'_s)$, a plaintext $\mu \in \mathcal{M}$, perform the following:

1. $\mathbf{r}_e, \mathbf{r}'_e \leftarrow D_{\mathbb{Z}^m,\alpha q}$, $\mathbf{t} = f_{\overline{\mathbf{A}}_r}(H_1(\mathbf{A}_s)) + f_{\mathbf{B}}(\mathbf{r}_e) \in \mathbb{Z}_q^n$,
   $\mathbf{t}' = f_{\overline{\mathbf{A}}'_r}(H_1(\mathbf{A}'_s)) + f_{\mathbf{B}'}(\mathbf{r}'_e) \in \mathbb{Z}_q^n$.

[1] See [21, Section 5] for details on FRD.

2. $\mathbf{A}_{r,\mathbf{t}} = \mathbf{A}_r + [\mathbf{0}|H_2(\mathbf{t})\mathbf{G}] \in \mathbb{Z}_q^{n\times m} = [\overline{\mathbf{A}}_r|H_2(\mathbf{t})\mathbf{G}-\overline{\mathbf{A}}_r\cdot\mathbf{T}_r] \in \mathbb{Z}_q^{n\times m}$, $\mathbf{A}'_{r,\mathbf{t}} = \mathbf{A}'_r + [\mathbf{0}|H_2(\mathbf{t}')\mathbf{G}] \in \mathbb{Z}_q^{n\times m} = [\overline{\mathbf{A}}'_r|H_2(\mathbf{t}')\mathbf{G} - \overline{\mathbf{A}}'_r \cdot \mathbf{T}'_r] \in \mathbb{Z}_q^{n\times m}$.
3. $\mathbf{s}, \mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n, \quad \mathbf{x}_1, \mathbf{x}'_1 \leftarrow D_{\mathbb{Z}^\ell,\alpha q}$.
4. $\mathbf{c}_0 = \mathbf{s}^t\mathbf{A}_{r,\mathbf{t}} + \mathbf{x}_0^t \in \mathbb{Z}_q^m, \quad \overline{\mathbf{c}}_1 = \mathbf{s}^t\mathbf{U} + \mathbf{x}_1^t \in \mathbb{Z}_q^\ell$,
   $\mathbf{c}'_0 = (\mathbf{s}')^t\mathbf{A}'_{r,\mathbf{t}} + (\mathbf{x}'_0)^t \in \mathbb{Z}_q^m, \quad \overline{\mathbf{c}}'_1 = (\mathbf{s}')^t\mathbf{U}' + (\mathbf{x}'_1)^t \in \mathbb{Z}_q^\ell$.
5. Set $\overline{ct} = (\mathbf{c}_0, \overline{\mathbf{c}}_1, \mathbf{r}_e, \mathbf{c}'_0, \overline{\mathbf{c}}'_1, \mathbf{r}'_e)$.
6. Sign on $\mu|pk_r|\overline{ct}$ to get the signature $(\mathbf{e}, \mathbf{r}_s)$ as follows:
   (a) $\mathbf{r}_s \leftarrow D_{\mathbb{Z}^m,\alpha q}$.
   (b) $\mathbf{h} = (h_1, \cdots, h_n) = f_{\overline{\mathbf{A}}_s}(H_3(\mu|pk_r|\overline{ct})) + f_{\mathbf{B}}(\mathbf{r}_s) \in \mathbb{Z}_q^n$.
   (c) $\mathbf{A}_{s,\mathbf{h}} = [\mathbf{A}_s|\mathbf{C}_0 + \sum_{i=1}^n h_i \cdot \mathbf{C}_i] \in \mathbb{Z}_q^{n\times(m+nk)}$,
   (d) $\mathbf{e} \in \mathbb{Z}^{m+nk} \leftarrow \mathsf{SampleD}(\mathbf{T}_s.\mathbf{A}_{s,\mathbf{h}}, \mathbf{u}, \sigma_2)$
7. $\mathbf{c}_1 = \overline{\mathbf{c}}_1 + \mu \cdot \lfloor q/2 \rfloor \in \mathbb{Z}_q^\ell, \quad \mathbf{c}'_1 = \overline{\mathbf{c}}'_1 + H(\mu) \cdot \lfloor q/2 \rfloor \in \mathbb{Z}_q^\ell$.
8. Output the ciphertext $ct = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{r}_e, \mathbf{r}_s, \mathbf{c}'_0, \mathbf{c}'_1, \mathbf{r}'_e, \mathbf{e})$.

USC(sk_r, pk_s, ct): On input a sender's public key $pk_s := (\mathbf{A}_s, \mathbf{A}'_s)$, a receiver's private key $sk_r := (\mathbf{T}_r, \mathbf{T}'_r)$, a ciphertext $ct = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{r}_e, \mathbf{r}_s, \mathbf{c}'_0, \mathbf{c}'_1, \mathbf{r}'_e, \mathbf{e})$, do the following:

1. Compute $\mathbf{t} = f_{\overline{\mathbf{A}}_r}(H_1(\mathbf{A}_s)) + f_{\mathbf{B}}(\mathbf{r}_e) \in \mathbb{Z}_q^n$ and $\mathbf{A}_{r,\mathbf{t}} = [\overline{\mathbf{A}}_r|H_2(\mathbf{t})\mathbf{G} - \overline{\mathbf{A}}_r \cdot \mathbf{T}_r] \in \mathbb{Z}_q^{n\times m}$.
2. $(\mathbf{s}, \mathbf{x}_0) \leftarrow \mathsf{Invert}(\mathbf{T}_r, \mathbf{A}_{r,\mathbf{t}}, \mathbf{c}_0)$.
3. Compute $\mathbf{E} \in \mathbb{Z}^{m\times\ell} \leftarrow \mathsf{SampleD}(\mathbf{T}_r, \mathbf{A}_{r,\mathbf{t}}, \mathbf{U}, \sigma_2)$.
4. Compute $\mathbf{v}^t = \mathbf{c}_1^t - (\mathbf{c}_0 - \mathbf{x}_0)^t\mathbf{E} = \mathbf{x}_1^t + \mu \cdot \lfloor q/2 \rfloor$.
5. Recover μ from $\mathbf{v}$ mod q.
6. $\overline{\mathbf{c}}_1 = \mathbf{c}_1 - \mu \cdot \lfloor q/2 \rfloor$ mod q, $\overline{\mathbf{c}}'_1 = \mathbf{c}'_1 - H(\mu) \cdot \lfloor q/2 \rfloor$ mod q, and let $\overline{ct} := (\mathbf{c}_0, \overline{\mathbf{c}}_1, \mathbf{r}_e, \mathbf{c}'_0, \overline{\mathbf{c}}'_1, \mathbf{r}'_e)$.
7. Compute $\mathbf{h} = (h_1, \cdots, h_n) = f_{\overline{\mathbf{A}}_s}(H_3(\mu|pk_r|\overline{ct})) + f_{\mathbf{B}}(\mathbf{r}_s) \in \mathbb{Z}_q^n$.
8. $\mathbf{A}_{s,\mathbf{h}} = [\mathbf{A}_s|\mathbf{C}_0 + \sum_{i=1}^n h_i \cdot \mathbf{C}_i] \in \mathbb{Z}_q^{n\times(m+nk)}$.
9. If $\mathbf{A}_{s,\mathbf{h}} \cdot \mathbf{e} = \mathbf{u} \bmod q$ and $\|\mathbf{e}\| \leq \sigma_2\sqrt{m+nk}$ then output μ; otherwise, output $\perp$.

Tag(sk_r): On input a receiver's private key $sk_r := (\mathbf{T}_r, \mathbf{T}'_r)$, return the tag $tg_r := \mathbf{T}'_r$.

Test($(tg_{r,i}, ct_i), (tg_{r,j}, ct_j)$): On input a tag $tg_{r,i} := \mathbf{T}'_{r,i}$, a ciphertext $ct_i = (\mathbf{c}_{0,i}, \mathbf{c}_{1,i}, \mathbf{r}_{e,i}, \mathbf{r}_{s,i}, \mathbf{c}'_{0,i}, \mathbf{c}'_{1,i}, \mathbf{r}'_{e,i}, \mathbf{e}_i)$ with respect to the receiver $\mathcal{R}_i$, and a tag $tg_{r,j} := \mathbf{T}'_{r,j}$, a ciphertext $ct_j = (\mathbf{c}_{0,j}, \mathbf{c}_{1,j}, \mathbf{r}_{e,j}, \mathbf{r}_{s,j}, \mathbf{c}'_{0,j}, \mathbf{c}'_{1,j}, \mathbf{r}'_{e,j}, \mathbf{e}_j)$ with respect to the receiver $\mathcal{R}_j$, do the following:

- For $\mathcal{R}_i$, do:
  1. Compute $\mathbf{t}'_i = f_{\overline{\mathbf{A}}'_{r,i}}(H_1(\mathbf{A}'_{s,i})) + f_{\mathbf{B}'}(\mathbf{r}'_{e,i}) \in \mathbb{Z}_q^n$ and $\mathbf{A}'_{r,\mathbf{t}_i} = [\overline{\mathbf{A}}'_{r,i} | H_2(\mathbf{t}'_i)\mathbf{G} - \overline{\mathbf{A}}'_{r,i} \cdot \mathbf{T}'_{r,i}] \in \mathbb{Z}_q^{n\times m}$,
  2. $(\mathbf{s}'_i, \mathbf{x}'_{0,i}) \leftarrow \mathsf{Invert}(\mathbf{T}'_{r,i}, \mathbf{A}'_{r,\mathbf{t}_i}, \mathbf{c}'_{0,i})$.
  3. Compute $\mathbf{E}' \in \mathbb{Z}^{m\times \ell} \leftarrow \mathsf{SampleD}(\mathbf{T}'_r, \mathbf{A}'_{r,\mathbf{t}}, \mathbf{U}', \sigma_2)$.
  4. Compute $(\mathbf{v}'_i)^t = (\mathbf{c}'_{1,i})^t - (\mathbf{c}'_{0,i} - \mathbf{x}'_{0,i})^t\mathbf{E}_i = (\mathbf{x}'_{1,i})^t + H(\mu_i) \cdot \lfloor q/2 \rfloor$.
  5. Recover $H(\mu_i)$ from $\mathbf{v}'_i$ mod q.
- For $\mathcal{R}_j$: Do the same steps as above for $\mathcal{R}_i$ to recover $H(\mu_j)$.
- Output 1 if $H(\mu_i) = H(\mu_j)$. Otherwise, output 0.

Theorem 9 (Correctness). *The proposed* SCET *scheme is correct following the conditions mentioned in Subsection 3.2 provided that H is collision-resistant.*

Proof. For any $pp \leftarrow \mathsf{Setup}(1^\lambda)$, $(pk_r, sk_r) \leftarrow \mathsf{KGr}(pp)$, $(pk_s, sk_s) \leftarrow \mathsf{KGs}(pp)$, $(pk_{r_1}, sk_{r_1}) \leftarrow \mathsf{KGr}(pp)$, $(pk_{s_1}, sk_{s_1}) \leftarrow \mathsf{KGs}(pp)$, $(pk_{r_2}, sk_{r_2}) \leftarrow \mathsf{KGr}(pp)$, $(pk_{s_2}, sk_{s_2}) \leftarrow \mathsf{KGs}(pp)$, $tg_1 \leftarrow \mathsf{Tag}(sk_{r_1})$ and $tg_2 \leftarrow \mathsf{Tag}(sk_{r_2})$, any ciphertexts ct_1 and ct_2, and any message $\mu \in \mathcal{M}$. We need to check the following:

- First, we will prove that $\Pr[\mu = \mathsf{USC}(sk_r, pk_s, \mathsf{SC}(pk_r, sk_s, \mu))] = 1 - \mathsf{negl}(\lambda)$. Indeed, let $ct = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{r}_e, \mathbf{r}_s, \mathbf{c}'_0,\ \mathbf{c}'_1, \mathbf{r}'_e, \mathbf{e})$ be a ciphertext outputted by $\mathsf{SC}(pk_r, sk_s, \mu)$. Now what we need to verify is Step 5 in the USC algorithm. To succesfully recover $\mu = (\mu_1, \cdots, \mu_\ell)$ from $\mathbf{v} = \mathbf{x}_1 + \mu \cdot \lfloor q/2 \rfloor$, we compare each component of $\mathbf{v} = (v_1, \cdots, v_\ell)$ to $q/2$. If $|v_i| < q/2$ then $\mu_i = 0$. Otherwise, $\mu_i = 1$. This is thanks to the smallness of $\mathbf{x}_1 \leftarrow D_{\mathbb{Z}^\ell, \alpha q}$.
- Second, we need to show that if $\mathsf{USC}(sk_{r_1}, pk_{s_1}, ct_1) = \mathsf{USC}(sk_{r_2}, pk_{s_2}, ct_2) = \mu \neq \perp$, then
$$\Pr[\mathsf{Test}(tg_1, ct_1, tg_2, ct_2) = 1] = 1 - \mathsf{negl}(\lambda).$$
This can be done in the same way as above.
- Finally, we show that if $\mathsf{USC}(sk_{r_1}, pk_{s_1}, ct_1) = \mu_1 \neq \mathsf{USC}(sk_{r_2}, pk_{s_2}, ct_2) = \mu_2$, then
$$\Pr[\mathsf{Test}(tg_1, ct_1, tg_2, ct_2) = 1] = \mathsf{negl}(\lambda).$$
This can be done in the same way as above with noting that if $H(\mu_1) = H(\mu_2)$ happens, then it must be that $\mu_1 = \mu_2$ due to the collision-resistance of H.

Note that, the negligibility in the above conditions comes from that of the trapdoor algotihms being used such as $\mathsf{Invert}, \mathsf{SampleD}$ with appropriately chosen parameters.

5. Security Analysis

Theorem 10 (IND-iCCA1). *The proposed* SCET *scheme is IND-iCCA1 secure under the hardness of the decisional-LWE* $\mathsf{dLWE}_{n,2(\overline{m}+\ell),q,\alpha q}$ *problem and the collision-resistance of the functions* $f_{\overline{\mathbf{A}}_r}(\cdot) + f_{\boldsymbol{B}}(\cdot)$ *for any* $\overline{\mathbf{A}}_r \xleftarrow{\$} \mathbb{Z}_q^{n\times\overline{m}}$ *and any* $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.

Proof. We consider a sequence of games in which the first game **Game IND0** is the original one. And the last game **Game IND5** is the "uniform-based" game. We will demnostrate that **Game IND**i and **Game IND**$(i+1)$ are indistinguishable for $i \in \{0, \cdots, 4\}$.

Game IND0. This is the original IND-iCCA1 game. Suppose that the target receiver is r^* and the target sender announced at **Challenge** phase by the adversary $\mathcal{A}_1$ is s^*. Also, let $\mathbf{t}^* \leftarrow f_{\overline{\mathbf{A}}_{r^*}}(H_1(\mathbf{A}_{s^*})) + f_{\mathbf{B}}(\mathbf{r}_e^*)$, and $\mathbf{t}'^* \leftarrow f_{\overline{\mathbf{A}}'_{r^*}}(H_1(\mathbf{A}'_{s^*})) + f_{\mathbf{B}'}(\mathbf{r}'^*_e)$.

Game IND1. This game is same as **Game IND0**, except that if $\mathcal{A}_1$ makes an unsigcryption query (r^*, s, ct) such that $f_{\overline{\mathbf{A}}_{r^*}}(H_1(\mathbf{A}_s)) + f_{\mathbf{B}}(\mathbf{r}_e) = \mathbf{t}^*$ or $f_{\overline{\mathbf{A}}'_{r^*}}(H_1(\mathbf{A}'_s)) + f_{\mathbf{B}'}(\mathbf{r}'_e) = \mathbf{t}'^*$, where $\mathbf{t}^*$ and $\mathbf{t}'^*$ are defined as in **Game IND0** (we name this event by Event_1), then the challenger outputs $\perp$.

Game IND1 and **Game IND0** are indistinguishable since the probability that the event Event_1 happens is negligible due to the collision resistance of $f_{\overline{\mathbf{A}}_{r^*}}(\cdot) + f_{\mathbf{B}}(\cdot)$, and $f_{\overline{\mathbf{A}}'_{r^*}}(\cdot) + f_{\mathbf{B}'}(\cdot)$.

Game IND2. This game is same as **Game IND1**, except that instead of $\mathbf{B}, \mathbf{B}'$ being uniform in $\mathbb{Z}_q^{n\times m}$, use $\mathsf{GenTrap}(n, \overline{m}, q, \sigma_1)$ to generate $(\mathbf{B}, \mathbf{T_B}), (\mathbf{B}', \mathbf{T}'_{\mathbf{B}}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{\overline{m}\times nk}$.

Game IND2 and **Game IND1** are indistinguishable due to the property of GenTrap algorithm. Namely, although being genereted using GenTrap, both $\mathbf{B}, \mathbf{B}'$ look uniform in $\mathbb{Z}_q^{n\times m}$.

Game IND3. This game is same as **Game IND2**, except that in the **Setup** phase, for the target receiver r^*, the challenger generates as follows:

1. Choose $\mathbf{t}^*, \mathbf{t}'^* \in \mathbb{Z}_q^n$ uniformly at random. The challenger uses $\mathbf{t}^*, \mathbf{t}'^*$ to build $\mathbf{A}_{r^*}, \mathbf{A}'_{r^*}$.
2. Choose $\mathbf{T}_{r^*}, \mathbf{T}'_{r^*} \leftarrow D_{\mathbb{Z}^{\overline{m}\times nk},\sigma_1}$ and then set $\mathbf{A}_{r^*} = [\overline{\mathbf{A}}_{r^*}|-H_2(\mathbf{t}^*)\mathbf{G} - \overline{\mathbf{A}}_{r^*} \cdot \mathbf{T}_{r^*}] \in \mathbb{Z}_q^{n\times m}$, $\mathbf{A}'_{r^*} = [\overline{\mathbf{A}}'_{r^*}| - H_2(\mathbf{t}'^*)\mathbf{G} - \overline{\mathbf{A}}'_{r^*} \cdot \mathbf{T}'_{r^*}] \in \mathbb{Z}_q^{n\times m}$.
3. The public key for r^* is $pk_{r^*} = (\mathbf{A}_{r^*}, \mathbf{A}'_{r^*})$ and the private key key for r^* is $sk_{r^*} = (\mathbf{T}_{r^*}, \mathbf{T}'_{r^*})$.

Game IND3 and **Game IND2** are indistinguishable since the distribution of $\mathbf{A}_{r^*}, \mathbf{A}'_{r^*}$ is the same as that of $\mathbf{A}_r, \mathbf{A}'_r$ for all $r \neq r^*$ which are generated in Step 1 of the KG algorithm.

Game IND4. This game is the same as **Game IND3**, except that in the **Challenge** phase, the challenger performs the following:

1. Choose randomly $b \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{A}_{r^*,\mathbf{t}^*} = \mathbf{A}_{r^*} + [\mathbf{0}|H_2(\mathbf{t}^*)\mathbf{G}] = [\overline{\mathbf{A}}_{r^*}| - \overline{\mathbf{A}}_{r^*} \cdot \mathbf{T}_{r^*}] \in \mathbb{Z}_q^{n\times m}$,
   $\mathbf{A}'_{r^*,\mathbf{t}'^*} = \mathbf{A}'_{r^*} + [\mathbf{0}|H_2(\mathbf{t}'^*)\mathbf{G}] = [\overline{\mathbf{A}}_{r^*}| - \overline{\mathbf{A}}'_{r^*} \cdot \mathbf{T}'_{r^*}] \in \mathbb{Z}_q^{n\times m}$.
3. Compute $\mathbf{r}_e^* \leftarrow \mathsf{SampleD}(\mathbf{T_B}, \mathbf{B}, (\mathbf{t}^* - f_{\overline{\mathbf{A}}_{r^*}}(H_1(\mathbf{A}_{s^*}))), \alpha q)$, ${\mathbf{r}'_e}^* \leftarrow \mathsf{SampleD}(\mathbf{T}'_\mathbf{B}, \mathbf{B}', (\mathbf{t}'^* - f_{\overline{\mathbf{A}}'_{r^*}}(H_1(\mathbf{A}'_{s^*}))), \alpha q)$.
4. Sample $\mathbf{s}, \mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n, \quad \hat{\mathbf{x}}_0, \hat{\mathbf{x}}'_0 \leftarrow D_{\mathbb{Z}^{\overline{m}},\alpha q}, \quad \mathbf{x}_1, \mathbf{x}'_1 \leftarrow D_{\mathbb{Z}^\ell,\alpha q}$.
5. Compute $\hat{\mathbf{c}}_0 = \mathbf{s}^t\overline{\mathbf{A}}_{r^*} + \hat{\mathbf{x}}_0^t \in \mathbb{Z}_q^{\overline{m}}$, $\overline{\mathbf{c}}_1 = \mathbf{s}^t\mathbf{U} + \mathbf{x}_1^t \in \mathbb{Z}_q^\ell$, $\hat{\mathbf{c}}'_0 = (\mathbf{s}')^t\overline{\mathbf{A}}'_{r^*} + (\hat{\mathbf{x}}'_0)^t \in \mathbb{Z}_q^{\overline{m}}$, $\overline{\mathbf{c}}'_1 = (\mathbf{s}')^t\mathbf{U}' + (\mathbf{x}'_1)^t \in \mathbb{Z}_q^\ell$,
6. Set $(\mathbf{c}_0^*)^t := (\hat{\mathbf{c}}_0^t|\hat{\mathbf{c}}_0^t\mathbf{T}_{r^*}), \quad (\mathbf{c}_0'^*)^t := ((\hat{\mathbf{c}}'_0)^t|(\hat{\mathbf{c}}'_0)^t\mathbf{T}'_{r^*})$.
7. Sign on $\mu_b^*|pk_{r^*}|\overline{ct}^*$ with $\overline{ct}^* = (\mathbf{c}_0^*, \overline{\mathbf{c}}_1^*, \mathbf{r}_e^*, \mathbf{c}_0'^*, \overline{\mathbf{c}}_1'^*, \mathbf{r}_e'^*)$ to get the signature $(\mathbf{e}^*, \mathbf{r}_s^*)$ as usuall.
8. $\mathbf{c}_1^* = \overline{\mathbf{c}}_1^* + \mu_b^* \cdot \lfloor q/2 \rfloor, \quad \mathbf{c}_1'^* = \overline{\mathbf{c}}_1'^* + H(\mu_b^*) \cdot \lfloor q/2 \rfloor$
9. Return $ct^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{r}_e^*, \mathbf{r}_s^*, \mathbf{c}_0'^*, \mathbf{c}_1'^*, \mathbf{r}_e'^*, \mathbf{e}^*)$ to $\mathcal{A}_1$.

Game IND4 and **Game IND3** are indistinguishable as the challenger ís just following the real signcryption algorithm SC with $\mathbf{A}_{r^*} = [\overline{\mathbf{A}}_{r^*}| - H_2(\mathbf{t}^*)\mathbf{G} - \overline{\mathbf{A}}_{r^*} \cdot \mathbf{T}_{r^*}] \in \mathbb{Z}_q^{n\times m}$, $\mathbf{A}'_{r^*} = [\overline{\mathbf{A}}'_{r^*}| - H_2(\mathbf{t}'^*)\mathbf{G} - \overline{\mathbf{A}}'_{r^*} \cdot \mathbf{T}'_{r^*}] \in \mathbb{Z}_q^{n\times m}$ and the distribution of $\mathbf{r}_e^*, \mathbf{r}_e'^*$ is still $D_{\mathbb{Z}^m,\alpha q}$ by the property of $\mathsf{SampleD}$.

Game IND5. This game is the same as **Game 4**, except that $\overline{\mathbf{c}}_0^*, \overline{\mathbf{c}}_1^*, \mathbf{r}_e^*, \overline{\mathbf{c}}_0'^*, \overline{\mathbf{c}}_1'^*, \mathbf{r}_e'^*$ are chosen uniformly at random.

Below, we are going to show that **Game IND5** and **Game IND4** are indistinguishable using a reduction from the hardness of the decision LWE problem.

Reduction from LWE. Suppose that $\mathcal{A}_1$ can distinguish **Game IND5** and **Game IND4**. Then we will construct an algorithm $\mathcal{B}_1$ that can solve an LWE instance.

LWE Instance. $\mathcal{B}_1$ is given a pair $(\mathbf{F}, \mathbf{c}^t) \in \mathbb{Z}_q^{n\times(2\overline{m}+2\ell)} \times \mathbb{Z}_q^{2(\overline{m}+\ell)}$ that can be parsed as $(\mathbf{A}|\mathbf{A}'|\mathbf{U}|\mathbf{U}', \hat{\mathbf{c}}_0^t|(\hat{\mathbf{c}}'_0)^t|\overline{\mathbf{c}}_1^t|(\overline{\mathbf{c}}'_1)^t) \in \mathbb{Z}_q^{n\times(\overline{m}+\overline{m}+\ell+\ell)} \times \mathbb{Z}_q^{\overline{m}+\overline{m}+\ell+\ell}$, and $\mathcal{B}_1$ has to decide whether

- (i) $(\mathbf{F}, \mathbf{c}^t)$ is an LWE instance: $\hat{\mathbf{c}}_0 = \mathbf{s}^t\mathbf{A} + \hat{\mathbf{x}}_0^t \in \mathbb{Z}_q^{\overline{m}}$, $\overline{\mathbf{c}}_1 = \mathbf{s}^t\mathbf{U} + \mathbf{x}_1^t \in \mathbb{Z}_q^{\ell}$, $\hat{\mathbf{c}}_0' = (\mathbf{s}')^t\mathbf{A}' + (\hat{\mathbf{x}}_0')^t \in \mathbb{Z}_q^{\overline{m}}$, $\overline{\mathbf{c}}_1' = (\mathbf{s}')^t\mathbf{U}' + (\mathbf{x}_1')^t \in \mathbb{Z}_q^{\ell}$, for some $\mathbf{s}, \mathbf{s}' \overset{\$}{\leftarrow} \mathbb{Z}_q^n$, $\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_0' \leftarrow D_{\mathbb{Z}^{\overline{m}}, \alpha q}$, $\mathbf{x}_1, \mathbf{x}_1' \leftarrow D_{\mathbb{Z}^{\ell}, \alpha q}$; or
- (ii) $(\mathbf{F}, \mathbf{c}^t)$ is uniform in $\mathbb{Z}_q^{n \times (2\overline{m}+2\ell)} \times \mathbb{Z}_q^{2(\overline{m}+\ell)}$.

The algorithms $\mathcal{B}_1$ and $\mathcal{A}_1$ play the following game:

Setup. $\mathcal{B}_1$ simulates public parameters pp, public keys for M senders and N receivers as follows:

- Pick $n, q, k, \overline{m}, m, \ell, n, N, M, \alpha, \sigma_1, \sigma_2$ and use hash functions H, H_1, H_2, H_3. The message space is $\mathcal{M}$.
- Randomly guess $r^* \overset{\$}{\leftarrow} \{1, \cdots, N\}$ to be the target receiver targeted by $\mathcal{A}_1$, and then set $\overline{\mathbf{A}}_{r^*} := \mathbf{A}$, $\overline{\mathbf{A}}_{r^*}' := \mathbf{A}'$.
- Choose $\mathbf{t}^*, \mathbf{t}'^* \in \mathbb{Z}_q^n$ uniformly at random and choose $\mathbf{T}_{r^*}, \mathbf{T}_{r^*}' \leftarrow D_{\mathbb{Z}^{\overline{m} \times nk}, \sigma_1}$ and then set $\mathbf{A}_{r^*} = [\overline{\mathbf{A}}_{r^*} | - H_2(\mathbf{t}^*)\mathbf{G} - \overline{\mathbf{A}}_{r^*} \cdot \mathbf{T}_{r^*}] \in \mathbb{Z}_q^{n \times m}$, $\mathbf{A}_{r^*}' = [\overline{\mathbf{A}}_{r^*}' | - H_2(\mathbf{t}'^*)\mathbf{G} - \overline{\mathbf{A}}_{r^*}' \cdot \mathbf{T}_{r^*}'] \in \mathbb{Z}_q^{n \times m}$.
- Also, use $\mathsf{GenTrap}(n, \overline{m}, q, \sigma_1)$ to generate $(\mathbf{B}, \mathbf{T_B}), (\mathbf{B}', \mathbf{T_B}') \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{\overline{m} \times nk}$.
- Sample $\mathbf{u} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$ and matrices $\mathbf{C}_0, \cdots, \mathbf{C}_n$, $\mathbf{C}_0', \cdots, \mathbf{C}_n' \overset{\$}{\leftarrow} \mathbb{Z}_q^{n \times m}$.
- For each receiver $r \in [N] \setminus \{r^*\}$ and each sender $s \in [M]$, use the algorithm KG to generate $(\mathbf{A}_r, \mathbf{T}_r)$, $(\mathbf{A}_r', \mathbf{T}_r')$, $(\mathbf{A}_s, \mathbf{T}_s)$, $(\mathbf{A}_s', \mathbf{T}_s') \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{\overline{m} \times nk}$.
- Set $pp = \{n, q, k, \overline{m}, , m, \ell, n, \alpha, \sigma_1, \sigma_2, N, M, \mathcal{M}, (\mathbf{C}_i, \mathbf{C}_i')_{i=0}^n, H, H_1$, H_2, $H_3\}$ as public parameters and $pk_r = (\mathbf{A}_r, \mathbf{A}_r')$, $pk_s = (\mathbf{A}_s, \mathbf{A}_s')$ as public keys corresponding to each receiver $r \in [N]$, and each sender $s \in [M]$.
- Send pp, pk_s's, pk_r's all to the adversary $\mathcal{A}_1$.

Phase 1. $\mathcal{A}_1$ adaptively makes a polynomially bounded number of the following queries:

- Private key query PKQ(r): If $r = r^*$, $\mathcal{B}_1$ rejects the query. Otherwise, $\mathcal{B}_1$ returns the private key $sk_r = (\mathbf{T}_r, \mathbf{T}_r')$ to $\mathcal{A}_1$.
- Signcryption query SCQ(r, s, μ): $\mathcal{B}_1$ sends the output ct of $\mathsf{SC}(pk_r$, $sk_s, \mu)$ back to $\mathcal{A}_1$.
- Unsigncryption query USQ(r, s, ct): If $\mathcal{A}_1$ makes an unsigcryption query (r, s, ct) such that $\mathbf{t}^* = f_{\overline{\mathbf{A}}_r}(H_1(\mathbf{A}_s)) + f_{\mathbf{B}}(\mathbf{r}_e)$ and $\mathbf{t}'^* = f_{\overline{\mathbf{A}}_r'}(H_1(\mathbf{A}_s')) + f_{\mathbf{B}'}(\mathbf{r}_e')$ then $\mathcal{B}_1$ outputs $\perp$. Otherwise, $\mathcal{B}_1$ sends the output $\mu/\perp$ of $\mathsf{USC}(sk_r, pk_s, ct)$ back to $\mathcal{A}_1$.
- Tag query TGQ(r): If $r = r^*$, $\mathcal{B}_1$ rejects the query. Otherwise, $\mathcal{B}_1$ sends the output $tg_r = \mathbf{T}_r'$ back to $\mathcal{A}_1$.

Challenge. $\mathcal{A}_1$ submits two messages μ_0^*, μ_1^* together with the target sender's keys (pk_{s^*}, sk_{s^*}). The adversary $\mathcal{B}_1$ does the following:

1. Choose randomly $b \xleftarrow{\$} \{0,1\}$.
2. Compute $\mathbf{r}_e^* \leftarrow \mathsf{SampleD}(\mathbf{T_B}, \mathbf{B}, (\mathbf{t}^* - f_{\overline{\mathbf{A}}_{r^*}}(H_1(\mathbf{A}_{s^*}))), \alpha q)$, $\mathbf{r'}_e^* \leftarrow \mathsf{SampleD}(\mathbf{T'_B}, \mathbf{B}', (\mathbf{t}'^* - f_{\overline{\mathbf{A}}'_{r^*}}(H_1(\mathbf{A}'_{s^*}))), \alpha q)$.
3. $\mathbf{A}_{r^*,\mathbf{t}^*} = \mathbf{A}_{r^*} + [\mathbf{0}|H_2(\mathbf{t}^*)\mathbf{G}] = [\overline{\mathbf{A}}_{r^*}| - \overline{\mathbf{A}}_{r^*} \cdot \mathbf{T}_{r^*}] \in \mathbb{Z}_q^{n\times m}$,
   $\mathbf{A}'_{r^*,\mathbf{t}'^*} = \mathbf{A}'_{r^*} + [\mathbf{0}|H_2(\mathbf{t}'^*)\mathbf{G}] = [\overline{\mathbf{A}}_{r^*}| - \overline{\mathbf{A}}'_{r^*} \cdot \mathbf{T}'_{r^*}] \in \mathbb{Z}_q^{n\times m}$.
4. Set $(\mathbf{c}_0^*)^t := (\hat{\mathbf{c}}_0^t | \hat{\mathbf{c}}_0^t \mathbf{T}_{r^*})$, $\quad (\mathbf{c}_0'^*)^t := ((\hat{\mathbf{c}}_0')^t | (\hat{\mathbf{c}}_0')^t \mathbf{T}'_{r^*})$.
5. Sign on $\mu_b^* | pk_{r^*} | \overline{ct}^*$ with $\overline{ct}^* = (\mathbf{c}_0^*, \overline{\mathbf{c}}_1^*, \mathbf{r}_e^*, \mathbf{c}_0'^*, \overline{\mathbf{c}}_1'^*, \mathbf{r}_e'^*)$ to get the signature $(\mathbf{e}^*, \mathbf{r}_s^*)$ as usuall.
6. $\mathbf{c}_1^* = \overline{\mathbf{c}}_1^* + \mu_b^* \cdot \lfloor q/2 \rfloor$, $\quad \mathbf{c}_1'^* = \overline{\mathbf{c}}_1'^* + H(\mu_b^*) \cdot \lfloor q/2 \rfloor$
7. Return $ct^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{r}_e^*, \mathbf{r}_s^*, \mathbf{c}_0'^*, \mathbf{c}_1'^*, \mathbf{r}_e'^*, \mathbf{e}^*)$ to $\mathcal{A}_1$.

Phase 2. $\mathcal{A}_1$ queries the oracles again as in **Phase 1** with a restriction that $\mathcal{A}_1$ is not allowed to make the queries $\mathrm{PKQ}(r^*)$ and $\mathrm{USQ}(r^*, s^*, ct^*)$. **Output.** $\mathcal{B}_1$ outputs whatever $\mathcal{A}_1$ outputs.

Analysis. The probability that an unsigncryption query (r, s, ct) makes $\mathbf{t} = \mathbf{t}^*$ and $\mathbf{t}' = \mathbf{t}'^*$ is negligible as $\mathbf{t}^*$ and $\mathbf{t}'^*$ are chosen randomly in **Setup** phase. Then, we have $H_2(\mathbf{t} - \mathbf{t}^*)$ and $H_2(\mathbf{t}' - \mathbf{t}'^*)$ are invertible then we can apply Invert as in the real unsigncryption algorithm USC. Obviously, if $(\mathbf{F}, \mathbf{c}^t)$ is the LWE instance then the view of $\mathcal{A}_1$ as in **Game IND4**; while if $(\mathbf{F}, \mathbf{c}^t)$ is uniform in $\mathbb{Z}_q^{n\times(2\overline{m}+2\ell)} \times \mathbb{Z}_q^{2(\overline{m}+\ell)}$ then the view of $\mathcal{A}_1$ as in **Game IND5**. Therefore, if $\mathcal{A}_1$ can distinguish **Game IND4** and **Game IND5** then $\mathcal{B}_1$ can solve the decision LWE problem. □

Theorem 11 (OW-iCCA1). *The proposed* SCET *scheme is OW-iCCA1 secure provided that H is an one-way hash function, the* $\mathsf{dLWE}_{n,2(\overline{m}+\ell),q,\alpha q}$ *problem is hard and the functions $f_{\overline{\mathbf{A}}_r}(\cdot) + f_{\mathbf{B}}(\cdot)$ are collision-resistant for any $\overline{\mathbf{A}}_r \xleftarrow{\$} \mathbb{Z}_q^{n\times\overline{m}}$ and any $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. In particular, the advantage of the OW-iCCA1 advesary is*

$$\epsilon \leq \epsilon_{H,OW} + \epsilon_{f,CR} + \epsilon_{LWE},$$

where $\epsilon_{H,OW}$ is the advantage of breaking the one-wayness of H, $\epsilon_{f,CR}$ is the advantage of finding collision for any functions $f_{\overline{\mathbf{A}}_r}(\cdot) + f_{\mathbf{B}}(\cdot)$ and ϵ_{LWE} is the advantage of solving the $\mathsf{dLWE}_{n,2(\overline{m}+\ell),q,\alpha q}$ *problem.*

Proof. We prove by giving a sequence of five games in which the first game is the original OW-iCCA1 one and in the last game, the ciphertext will be chosen randomly. Obviously, in the last game the advantage of the

OW-iCCA1 adversary is zero. For $i \in \{0, 1, 2, 3, 4\}$, let W_i be the event that the OW-iCCA1 adversary $\mathcal{A}_2$ wins **Game OW**i, we need to prove that $\Pr[W_0]$ is negligible. To do that we will show that for $i \in \{0, \cdots, 4\}$, $|\Pr[W_i] - \Pr[W_{i+1}]|$ is negligible, guaranteed by the one-wayness of hash functions and especially the hardness of the decision LWE problem.

Game OW0. This is the original OW-iCCA1 game. Suppose that the target receiver is r^* and the target sender announced at **Challenge** phase by the adversary $\mathcal{A}_2$ is s^*. Also, assume that $\mathbf{t}^* := f_{\overline{\mathbf{A}}_{r^*}}(H_1(\mathbf{A}_{s^*})) + f_{\mathbf{B}}(\mathbf{r}_e^*)$, and $\mathbf{t}'^* := f_{\overline{\mathbf{A}}'_{r^*}}(H_1(\mathbf{A}'_{s^*})) + f_{\mathbf{B}'}(\mathbf{r}_e'^*)$. Note that, the adversary $\mathcal{A}_2$ can get the trapdoor $\mathbf{T}'_{r^*}$ using the trapdoor query for the target receiver r^*.

Game OW1. This game is same as **Game OW0**, except that in the **Challenge** phase, on the challenge plaintext $\mu^* \xleftarrow{\$} \mathcal{M}$, the challenger first chooses $\mu' \xleftarrow{\$} \mathcal{M}$ then signcrypts μ^* in $\mathbf{c}_1^*$ and $H(\mu')$ instead of μ^* in $\mathbf{c'}_1^*$ of the challenge ciphertext ct^*, i.e., $\mathbf{c}_1^* = \overline{\mathbf{c}}_1^* + \mu^* \cdot \lfloor q/2 \rfloor$, $\mathbf{c}_1'^* = \overline{\mathbf{c}}_1'^* + H(\mu') \cdot \lfloor q/2 \rfloor$.

Since the view of the adverary $\mathcal{A}_2$ is the same in both **Game OW1** and **Game OW0**, except the case $\mathcal{A}_2$ can break the one-wayness of H, we have

$$|\Pr[W_1] - \Pr[W_0]| \le \epsilon_{H,OW}.$$

Game OW2. This game is same as **Game OW1**, except that if $\mathcal{A}_2$ makes an unsigcryption query (r^*, s, ct) such that $f_{\overline{\mathbf{A}}_{r^*}}(H_1(\mathbf{A}_s)) + f_{\mathbf{B}}(\mathbf{r}_e) = \mathbf{t}^*$ or $f_{\overline{\mathbf{A}}'_{r^*}}(H_1(\mathbf{A}'_s)) + f_{\mathbf{B}'}(\mathbf{r}'_e) = \mathbf{t}'^*$, where $\mathbf{t}^*$ and $\mathbf{t}'^*$ are defined as in **Game OW0** (we name this event by Event_1), then the challenger outputs $\perp$.

Since the view of the adverary $\mathcal{A}_2$ is the same, except once the event Event_1 happens, in both **Game OW2** and **Game OW1**, we have

$$|\Pr[W_2] - \Pr[W_1]| \le \epsilon_{f,CR}.$$

Game OW3. This game is same as **Game OW2**, except that instead of $\mathbf{B}, \mathbf{B}'$ being uniform in $\mathbb{Z}_q^{n \times m}$, the challenger uses $\mathsf{GenTrap}(n, \overline{m}, q, \sigma_1)$ to generate $(\mathbf{B}, \mathbf{T_B}), (\mathbf{B}', \mathbf{T}'_{\mathbf{B}}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{\overline{m} \times nk}$.

Due to the fact that $\mathbf{B}, \mathbf{B}'$ generated by $\mathsf{GenTrap}$ are close to uniform, then we have $\Pr[W_3] = \Pr[W_2]$.

Game OW4. This game is same as **Game OW3**, except that in the **Setup** phase, for the target receiver r^*, the challenger generates as follows:

1. Choose $\mathbf{t}^*, \mathbf{t}'^* \in \mathbb{Z}_q^n$ uniformly at random. The challenger uses $\mathbf{t}^*, \mathbf{t}'^*$ to build $\mathbf{A}_{r^*}, \mathbf{A}'_{r^*}$.
2. Choose $\mathbf{T}_{r^*}, \mathbf{T}'_{r^*} \leftarrow D_{\mathbb{Z}^{\overline{m} \times nk}, \sigma_1}$ and then set $\mathbf{A}_{r^*} = [\overline{\mathbf{A}}_{r^*} | -H_2(\mathbf{t}^*)\mathbf{G} - \overline{\mathbf{A}}_{r^*} \cdot \mathbf{T}_{r^*}] \in \mathbb{Z}_q^{n \times m}$, $\mathbf{A}'_{r^*} = [\overline{\mathbf{A}}'_{r^*} | - H_2(\mathbf{t}'^*)\mathbf{G} - \overline{\mathbf{A}}'_{r^*} \cdot \mathbf{T}'_{r^*}] \in \mathbb{Z}_q^{n \times m}$.
3. The public key for r^* is $pk_{r^*} = (\mathbf{A}_{r^*}, \mathbf{A}'_{r^*})$ and the private key key for r^* is $sk_{r^*} = (\mathbf{T}_{r^*}, \mathbf{T}'_{r^*})$.

In this game, once the adversary $\mathcal{A}_2$ makes a trapdoor query for r^*, the challenger still easily returns $\mathbf{T}'_{r^*}$ to $\mathcal{A}_2$. In **Game OW4** the view of $\mathcal{A}_2$ is unchanged in comparison with in **Game OW3** since the distribution of $\mathbf{A}_{r^*}, \mathbf{A}'_{r^*}$ is the same as that of $\mathbf{A}_r, \mathbf{A}'_r$ for all $r \neq r^*$ which are generated in Step 1 of the KG algorithm. Therefore,

$$\Pr[W_4] = \Pr[W_3].$$

Game OW5. This game is same as **Game OW4**, except that in the **Challenge** phase, on the challenge message μ^*, the challenger does the following:

1. Compute $\mathbf{r}_e^* \leftarrow \mathsf{SampleD}(\mathbf{T_B}, \mathbf{B}, (\mathbf{t}^* - f_{\overline{\mathbf{A}}_{r^*}}(H_1(\mathbf{A}_{s^*}))), \alpha q)$, $\mathbf{r}'^*_e \leftarrow \mathsf{SampleD}(\mathbf{T}'_\mathbf{B}, \mathbf{B}', (\mathbf{t}'^* - f_{\overline{\mathbf{A}}'_{r^*}}(H_1(\mathbf{A}'_{s^*}))), \alpha q)$.
2. Sample $\mathbf{s}, \mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n, \quad \hat{\mathbf{x}}_0, \hat{\mathbf{x}}'_0 \leftarrow D_{\mathbb{Z}^{\overline{m}}, \alpha q}, \quad \mathbf{x}_1, \mathbf{x}'_1 \leftarrow D_{\mathbb{Z}^\ell, \alpha q}$.
3. Compute $\hat{\mathbf{c}}_0 = \mathbf{s}^t \overline{\mathbf{A}}_{r^*} + \hat{\mathbf{x}}_0^t \in \mathbb{Z}_q^{\overline{m}}$, $\overline{\mathbf{c}}_1 = \mathbf{s}^t\mathbf{U} + \mathbf{x}_1^t \in \mathbb{Z}_q^\ell$, $\hat{\mathbf{c}}'_0 = (\mathbf{s}')^t \overline{\mathbf{A}}'_{r^*} + (\hat{\mathbf{x}}'_0)^t \in \mathbb{Z}_q^{\overline{m}}$, $\overline{\mathbf{c}}'_1 = (\mathbf{s}')^t\mathbf{U}' + (\mathbf{x}'_1)^t \in \mathbb{Z}_q^\ell$,
4. Set $(\mathbf{c}_0^*)^t := (\hat{\mathbf{c}}_0^t | \hat{\mathbf{c}}_0^t)\mathbf{T}_{r^*}, \quad (\mathbf{c}'^*_0)^t := ((\hat{\mathbf{c}}'_0)^t | (\hat{\mathbf{c}}'_0)^t)\mathbf{T}'_{r^*}$.
5. Sign on $\mu_b^* | pk_{r^*} | \overline{ct}^*$ with $\overline{ct}^* = (\mathbf{c}_0^*, \overline{\mathbf{c}}_1^*, \mathbf{r}_e^*, \mathbf{c}'^*_0, \overline{\mathbf{c}}'^*_1, \mathbf{r}'^*_e)$ to get the signature $(\mathbf{e}^*, \mathbf{r}_s^*)$ as usuall.
6. $\mu' \xleftarrow{\$} \mathcal{M}, \quad \mathbf{c}_1^* = \overline{\mathbf{c}}_1^* + \mu^* \cdot \lfloor q/2 \rfloor, \quad \mathbf{c}'^*_1 = \overline{\mathbf{c}}'^*_1 + H(\mu') \cdot \lfloor q/2 \rfloor$.
7. Return $ct^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{r}_e^*, \mathbf{r}_s^*, \mathbf{c}'^*_0, \mathbf{c}'^*_1, \mathbf{r}'^*_e, \mathbf{e}^*)$ to $\mathcal{A}_2$.

We have $\Pr[W_5] = \Pr[W_4]$ as the distributions of corresponding components in ct^* in **Game OW5** and **Game OW4** are the same.

Game OW6. This game is same as **Game OW5**, except that the challenge ciphertext $ct^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{r}_e^*, \mathbf{r}_s^*, \mathbf{c}'^*_0, \mathbf{c}'^*_1, \mathbf{r}'^*_e, \mathbf{e}^*)$ is chosen uniformly at random. The advantage of $\mathcal{A}_2$ in this game is obviously zero, i.e., $\Pr[W_6] = 0$.

At this point, we show that $|\Pr[W_6] - \Pr[W_5]| \leq \epsilon_{LWE}$ which is negligible by using a reduction from the LWE assumption as in Theorem 10. □

Before stating the SUF-iCMA security, we recap the so-called *abort-resistant hash functions*, presented in [21, Section 7.4.1]. We will exploit the hash functions in designing answers to the adversary's queries.

Definition 9 ([21, Definition 26]). Let $\mathcal{H} := \{H : X \to Y\}$ be a family of hash functions H from X to Y where $0 \in Y$. For a set of $Q+1$ inputs $\overline{\mathbf{h}} := (\mathbf{h}^*, \mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(Q)})$, the non-abort probability of $\overline{\mathbf{h}}$ is defined as

$$\alpha(\overline{\mathbf{h}}) := \Pr[H(\mathbf{h}^*) = 0 \text{ and } H(\mathbf{h}^{(1)}) \neq 0 \text{ and } \cdots \text{ and } H(\mathbf{h}^{(Q)}) \neq 0],$$

where the probability is over the random choice of H in $\mathcal{H}$. And $\mathcal{H}$ is called $(Q, \alpha_{\min}, \alpha_{\max})$ abort-resistant if for all $\overline{\mathbf{h}} := (\mathbf{h}^*, \mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(Q)})$ and $\mathbf{h}^* \notin \{\mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(Q)}\}$, we have $\alpha_{\min} \leq \alpha(\overline{\mathbf{h}}) \leq \alpha_{\max}$.

Particularly, we have the following result that will be applied to the security proof for the proposed signcryption construction.

Lemma 12 ([21, Lemma 27]). *let q be a prime and $0 < Q < q$. Consider the family $\mathcal{H}_{Wat} := \{H_{\mathbf{x}} : \mathbb{Z}_q^n \setminus \{\mathbf{0}\} \to \mathbb{Z}_q : \mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}\}$ defined as $H_{\mathbf{x}}(\mathbf{h}) = 1 + \sum_{i=1}^n x_i h_i \in \mathbb{Z}_q$ where $\mathbf{h} = (h_1, \cdots, h_n) \in \mathbb{Z}_q^n$. Then $\mathcal{H}_{Wat}$ is $(Q, \frac{1}{q}(1 - \frac{Q}{q}), \frac{1}{q})$ abort-resistant.*

Now, it is the time we state and prove the SUF-iCMA security for SCET.

Theorem 13 (SUF-iCMA). *Our SCET is SUF-iCMA secure in the standard model provided that the SIS problem is intractable. In particular, assume that there is an adversarial algorithm $\mathcal{F}$ who can win the SUF-iCMA game making at most $Q < q/2$ adaptive chosen message queries. Then, there is an algorithm $\mathcal{G}$ who is able to solve the $\mathsf{SIS}_{n,2\overline{m}+2nk+1,q,\beta}$ problem, with $\beta := 2\sigma_1 \cdot \sigma_2 \cdot \sqrt{n+1} \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{m} + \sqrt{nk}) \cdot \sqrt{m+nk}$..*

Proof. We assume by contradiction that if there exists a forger $\mathcal{F}$ who can break the SUF-iCMA security of the SCET scheme, then we can build from $\mathcal{F}$ an algorithm $\mathcal{G}$ that can find a solution to a given SIS problem.

We will give proof for the SUF-iCMA in two cases: **Case 1:** The forger $\mathcal{F}$ forges on an unqueried message and **Case 2:** The forger $\mathcal{F}$ who forges on a queried message. Suppose that, $\mathcal{F}$ makes at most Q adaptive signcryption queries on $(r^{(1)}, s^{(1)}, \mu^{(1)}), \cdots, (r^{(Q)}, s^{(Q)}, \mu^{(Q)}),$.

We consider **Case 1** first.

SIS Instance. Suppose that the algorithm $\mathcal{G}$ is given the following SIS problem below:

$$\mathbf{F} \cdot \mathbf{x} = \mathbf{0} \pmod{q}, \text{ where } \mathbf{F} \xleftarrow{\$} \mathbb{Z}_q^{n \times (2\overline{m}+2nk+1)}, \|\mathbf{x}\| \leq \beta_1, \tag{1}$$

where $\beta_1 = \sigma_2 \cdot \sqrt{n+1} \cdot (\sqrt{m} + \sqrt{nk})\sqrt{m+nk+1}$. Then, $\mathcal{G}$ parses $\mathbf{F}$ as $\mathbf{F} := [\overline{\mathbf{A}}|\mathbf{W}|\mathbf{u}|\overline{\mathbf{A}}'|\mathbf{W}']$, where $\overline{\mathbf{A}}, \overline{\mathbf{A}}' \in \mathbb{Z}_q^{n \times \overline{m}}$, $\mathbf{u} \in \mathbb{Z}_q^n$, and $\mathbf{W}, \mathbf{W}' \in \mathbb{Z}_q^{n \times nk}$. The algorithms $\mathcal{G}$ and $\mathcal{F}$ play the following game:

Setup. $\mathcal{G}$ simulates public parameters *pp*, public keys for M senders and N receivers as follows:

- Pick $n, q, k, m, \ell, n, \alpha, N, M, \sigma_1, \sigma_2, H, H_1, H_2, H_3, \mathcal{M}$ as system parameters.
- Guess $s^* \xleftarrow{\$} \{1, \cdots, M\}$ to be the target sender that $\mathcal{F}$ would like to forge, and then set $\overline{\mathbf{A}}_{s^*} := \overline{\mathbf{A}}$, $\overline{\mathbf{A}}'_{s^*} := \overline{\mathbf{A}}'$, $\mathbf{A}_{s^*} := [\overline{\mathbf{A}}_{s^*}|\mathbf{W}] \in \mathbb{Z}_q^{n\times m}$, $\mathbf{A}'_{s^*} := [\overline{\mathbf{A}}'_{s^*}|\mathbf{W}'] \in \mathbb{Z}_q^{n\times m}$. Recall that $m = \overline{m} + nk$.
- Also, use $\mathsf{GenTrap}(n, \overline{m}, q, \sigma_1)$ to generate $(\mathbf{B}, \mathbf{T}_{\mathbf{B}}), (\mathbf{B}', \mathbf{T}'_{\mathbf{B}}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{m\times m}$, and choose randomly $\mathbf{U}, \mathbf{U}' \xleftarrow{\$} \mathbb{Z}_q^{n\times \ell}$.
- For $i \in \{0, \cdots, n\}$, sample $\mathbf{T}_{s^*,i}, \mathbf{T}'_{s^*,i} \leftarrow D_{\mathbb{Z}^{\overline{m}\times nk},\sigma_1}$. Let $x_0 := 1$. Choose $\mathbf{x} := (x_1, \cdots, x_n) \xleftarrow{\$} \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$ and for $i \in \{0, \cdots, n\}$, let $\mathbf{H}_i = x_i\mathbf{I}_n$ and set $\mathbf{C}_i := \mathbf{H}_i\mathbf{G} - \overline{\mathbf{A}}_{s^*}\mathbf{T}_{s^*,i} \pmod{q} \in \mathbb{Z}_q^{n\times nk}$, $\mathbf{C}'_i := \mathbf{H}'_i\mathbf{G} - \overline{\mathbf{A}}'_{s^*}\mathbf{T}'_{s^*,i} \pmod{q} \in \mathbb{Z}_q^{n\times nk}$. Obviously, by Lemma 12, such an $\mathbf{x}$ will define an abort-resistant hash function belonging to $\mathcal{H}_{Wat}$.
- For each $j \in [Q]$: repeat choosing $\mathbf{h}^{(j)} = (h_1^{(j)}, \cdots, h_n^{(j)}) \in \mathbb{Z}_q^n$ uniformly at random until being such that $(1 + \sum_{i=1}^{n} x_i \cdot h_i^{(j)}) \neq 0$.
- For all $s \in [M] \setminus \{s^*\}$ and all $r \in [N]$, use $\mathsf{GenTrap}(n, \overline{m}, q, \sigma_1)$ to generate $(\mathbf{A}_r, \mathbf{T}_r)$, $(\mathbf{A}'_r, \mathbf{T}'_r)$, $(\mathbf{A}_s, \mathbf{T}_s)$, $(\mathbf{A}'_s, \mathbf{T}'_s) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{\overline{m}\times nk}$.
- Set $pp = \{n, q, k, m, \ell, n, N, M, \alpha, \sigma_1, \sigma_2, \mathcal{M}, (\mathbf{C}_i, \mathbf{C}'_i)_{i=0}^{n}, H, H_1, H_2, H_3\}$ as public parameters and $pk_r = (\mathbf{A}_r, \mathbf{A}'_r)$, $pk_s = (\mathbf{A}_s, \mathbf{A}'_s)$ as public keys corresponding to each receiver $r \in [N]$, and each sender $s \in [M]$.
- Send *pp*, pk_s's, pk_r's all to the forger $\mathcal{F}$.

Queries. $\mathcal{F}$ can adaptively make PKQ, SCQ and TGQ queries polynomially many times and in any order. Accordingly, $\mathcal{G}$ responds to the queries made by $\mathcal{F}$ as follows:

- For private key queries PKQ(s): if $s = s^*$, $\mathcal{G}$ rejects it. Otherwise, $\mathcal{G}$ returns the private key $\mathrm{sk}_s = (\mathbf{T}_s, \mathbf{T}'_s)$ of a sender $\mathcal{S}_s$ to $\mathcal{F}$.
- For the j-th signcryption query SCQ($r^{(j)}, s^{(j)}, \mu^{(j)}$) querie: If $s^{(j)} \neq s^*$, $\mathcal{G}$ sends the output *ct* of $\mathsf{SC}(pk_{r^{(j)}}, sk_{s^{(j)}}, \mu^{(j)})$ back to $\mathcal{F}$. Otherwise, $\mathcal{G}$ creates a ciphertext *ct* on input $(pk_{r^{(j)}}, sk_{s^*}, \mu)$ as follows:
  1. $\mathbf{r}_e, \mathbf{r}'_e \leftarrow D_{\mathbb{Z}^m,\alpha q}$, $\mathbf{t} = f_{\overline{\mathbf{A}}_r}(H_1(\mathbf{A}_{s^*})) + f_{\mathbf{B}}(\mathbf{r}_e) \in \mathbb{Z}_q^n$,
     $\mathbf{t}' = f_{\overline{\mathbf{A}}'_r}(H_1(\mathbf{A}'_{s^*})) + f_{\mathbf{B}'}(\mathbf{r}'_e) \in \mathbb{Z}_q^n$.
  2. $\mathbf{A}_{r,\mathbf{t}} = \mathbf{A}_r + [\mathbf{0}|H_2(\mathbf{t})\mathbf{G}] \in \mathbb{Z}_q^{n\times m} = [\overline{\mathbf{A}}_r|H_2(\mathbf{t})\mathbf{G} - \overline{\mathbf{A}}_r \cdot \mathbf{T}_r] \in \mathbb{Z}_q^{n\times m}$,
     $\mathbf{A}'_{r,\mathbf{t}} = \mathbf{A}'_r + [\mathbf{0}|H_2(\mathbf{t}')\mathbf{G}] \in \mathbb{Z}_q^{n\times m} = [\overline{\mathbf{A}}'_r|H_2(\mathbf{t}')\mathbf{G} - \overline{\mathbf{A}}'_r \cdot \mathbf{T}'_r] \in \mathbb{Z}_q^{n\times m}$.

3. $\mathbf{s}, \mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n, \quad \mathbf{x}_0, \mathbf{x}_0' \leftarrow D_{\mathbb{Z}^m, \alpha q}, \quad \mathbf{x}_1, \mathbf{x}_1' \leftarrow D_{\mathbb{Z}^\ell, \alpha q}$.
4. $\mathbf{c}_0 = \mathbf{s}^t \mathbf{A}_{r,\mathbf{t}} + \mathbf{x}_0^t \in \mathbb{Z}_q^m, \qquad \overline{\mathbf{c}}_1 = \mathbf{s}^t \mathbf{U} + \mathbf{x}_1^t \in \mathbb{Z}_q^\ell$,
   $\mathbf{c}_0' = (\mathbf{s}')^t \mathbf{A}_{r,\mathbf{t}}' + (\mathbf{x}_0')^t \in \mathbb{Z}_q^m, \quad \overline{\mathbf{c}}_1' = (\mathbf{s}')^t \mathbf{U}' + (\mathbf{x}_1')^t \in \mathbb{Z}_q^\ell$.
5. $\overline{ct} = (\mathbf{c}_0, \overline{\mathbf{c}}_1, \mathbf{r}_e, \mathbf{c}_0', \overline{\mathbf{c}}_1', \mathbf{r}_e')$.
6. Generate a signature on $\mu^{(j)} | pk_{r^{(j)}} | \overline{ct}$:
   (a) Compute $\mathbf{r}_s \leftarrow \mathsf{SampleD}(\mathbf{T}_\mathbf{B}, \mathbf{B}, (\mathbf{h}^{(j)} - f_{\overline{\mathbf{A}}_{s^*}}(H_3(\mu^{(j)} | pk_r | \overline{ct})))), \alpha q)$.
   (b) $\mathbf{A}_{s^*,\mathbf{h}} = [\mathbf{A}_{s^*} | \mathbf{C}_0 + \sum_{i=1}^n h_i^{(j)} \cdot \mathbf{C}_i] = [\mathbf{A}_{s^*} | \mathbf{H}\mathbf{G} - \overline{\mathbf{A}}_{s^*} \mathbf{T}_{s^*}] \in \mathbb{Z}_q^{n \times (m+nk)}$, with $\mathbf{H} := \mathbf{H}_0 + \sum_{i=1}^n h_i \cdot \mathbf{H}_i = (1 + \sum_{i=1}^n x_i \cdot h_i^{(j)}) \mathbf{I}_n \neq 0$ and $\mathbf{T}_{s^*} := \mathbf{T}_{s^*,0} + \sum_{i=1}^n h_i \cdot \mathbf{T}_{s^*,i}$. Note that, by Lemma 8, $\mathbf{T}_{s^*}$ is a $\mathbf{G}$-trapdoor for $\mathbf{A}_{s^*,\mathbf{h}}$ with tag $\mathbf{H}$.
   (c) $\mathbf{e} \in \mathbb{Z}^{m+nk} \leftarrow \mathsf{SampleD}(\mathbf{T}_{s^*}, \mathbf{A}_{s^*,\mathbf{h}}, \mathbf{u}, \sigma_2)$.
   (d) $(\mathbf{e}, \mathbf{r}_s)$ is the signature.
7. $\mathbf{c}_1 = \overline{\mathbf{c}}_1 + \mu^{(j)} \cdot \lfloor q/2 \rfloor, \quad \mathbf{c}_1' = \overline{\mathbf{c}}_1' + H(\mu^{(j)}) \cdot \lfloor q/2 \rfloor$.
8. Output $ct = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{r}_e, \mathbf{r}_s, \mathbf{c}_0', \mathbf{c}_1', \mathbf{r}_e', \mathbf{e})$.

- For unsigncryption query USQ(r, s, ct): $\mathcal{G}$ simply sends the output $\mu/\bot$ of $\mathsf{USC}(sk_r, pk_s, ct)$ back to $\mathcal{F}$ since $\mathcal{G}$ has $(\mathbf{T}_r, \mathbf{T}_r')$ for all $r \in [N]$.

- For tag query TGQ(r): $\mathcal{G}$ simply sends the output $tg_r := \mathbf{T}_r'$ back to $\mathcal{F}$.

Forge. The forger $\mathcal{F}$ outputs an index r^* of some receiver and its corresponding key-pair (pk_{r^*}, sk_{r^*}), and a new valid ciphertext $ct^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{r}_e^*, \mathbf{c}_0'^*, \mathbf{c}_1'^*, \mathbf{r}_e'^*, \mathbf{e}^*)$ on message μ^*, i.e., ct^* must be not the output of any previuos query SCQ(r, s, μ), and $\mathsf{USC}(sk_{r^*}, pk_{s^*}, ct^*) \neq \bot$.

Analysis. At the moment, $\mathcal{G}$ proceeds the following steps:

- Compute $\mathbf{h}^* = (h_1^*, \cdots, h_n^*) \leftarrow f_{\overline{\mathbf{A}}_{s^*}}(H_3(\mu^* | pk_{r^*} | \overline{ct}^*)) + f_\mathbf{B}(\mathbf{r}_{s^*}) \in \mathbb{Z}_q^n$ and check if $\mathbf{H} := \mathbf{H}_0 + \sum_{i=1}^n h_i^* \cdot \mathbf{H}_i = (1 + \sum_{i=1}^n x_i \cdot h_i^*) \mathbf{I}_n = 0$. If not, $\mathcal{G}$ aborts the game. Otherwise, it computes $\mathbf{A}_{s^*,\mathbf{h}^*} = [\mathbf{A}_{s^*} | \mathbf{C}_0 + \sum_{i=1}^n h_i^* \cdot \mathbf{C}_i] = [\overline{\mathbf{A}}_{s^*} | \mathbf{W} | -\overline{\mathbf{A}}_{s^*} \mathbf{T}_{s^*}] \in \mathbb{Z}_q^{n \times (m+nk)}$. Note that, the probability that $\mathcal{G}$ aborts the game is negligible by appropriately choosing parameters via Lemma 12.

- From $\mathbf{A}_{s^*,\mathbf{h}^*} \cdot \mathbf{e}^* = \mathbf{u} \bmod q$ and $\|\mathbf{e}^*\| \leq \sigma_2 \sqrt{m + nk}$, we have $[\overline{\mathbf{A}}_{s^*} | \mathbf{W} | - \overline{\mathbf{A}}_{s^*} \mathbf{T}_{s^*}] \cdot \mathbf{e}^* = \mathbf{u} \pmod q$, equivalently,
$$[\overline{\mathbf{A}}_{s^*} | \mathbf{W}] \cdot \left(\begin{bmatrix} \mathbf{I}_{\overline{m}} & \mathbf{0} & -\mathbf{T}_{s^*} \\ \mathbf{0} & \mathbf{I}_{nk} & \mathbf{0} \end{bmatrix} \cdot \mathbf{e}^* \right) = \mathbf{u} \pmod q.$$

- Let $\widehat{\mathbf{x}} := \begin{bmatrix} \mathbf{I}_{\overline{m}} & \mathbf{0} & -\mathbf{T}_{s^*} \\ \mathbf{0} & \mathbf{I}_{nk} & \mathbf{0} \end{bmatrix} \cdot \mathbf{e}^* \neq \mathbf{0}$, then $[\overline{\mathbf{A}} | \mathbf{W} | \mathbf{u}] \cdot \left(\begin{smallmatrix} \widehat{\mathbf{x}} \\ -1 \end{smallmatrix} \right) = \mathbf{0} \pmod q$. Hence, $\mathcal{G}$ gets a solution $\mathbf{x}$ to the SIS problem (1), i.e., $\mathbf{F} \cdot \mathbf{x} =$

$\mathbf{0} \pmod{q}$, with $\mathbf{x} = \left(\begin{smallmatrix} \widehat{\mathbf{x}} \\ -1 \\ \mathbf{0} \end{smallmatrix}\right)$, and by Lemma 2,

$$\begin{aligned} \|\mathbf{x}\| = \|\widehat{\mathbf{x}}\| + 1 &\leq s_1(\mathbf{T}_{s^*}) \cdot \|\mathbf{e}^*\| + 1 \\ &\leq \sigma_1 \cdot \sigma_2 \cdot \sqrt{n+1} \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk}) \cdot \sqrt{m+nk} + 1. \end{aligned}$$

Now, we consider **Case 2**. Suppose that the algorithm $\mathcal{G}$ is given the following SIS problem below:

$$\mathbf{F} \cdot \mathbf{x} = \mathbf{0} \pmod{q}, \text{ where } \mathbf{F} \xleftarrow{\$} \mathbb{Z}_q^{n \times (2\overline{m}+2nk)}, \|\mathbf{x}\| \leq \beta_2, \tag{2}$$

where $\beta_2 := 2\sigma_1 \cdot \sigma_2 \cdot \sqrt{n+1} \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk}) \cdot \sqrt{m+nk}$. Then, $\mathcal{G}$ parses $\mathbf{F}$ as $\mathbf{F} := [\overline{\mathbf{A}}|\mathbf{W}|\overline{\mathbf{A}}'|\mathbf{W}']$, where $\overline{\mathbf{A}}, \overline{\mathbf{A}}' \in \mathbb{Z}_q^{n \times \overline{m}}$, and $\mathbf{W}, \mathbf{W}' \in \mathbb{Z}_q^{n \times nk}$. The algorithms $\mathcal{G}$ and $\mathcal{F}$ play the following game:

Setup. This phase is the same as the **Setup** phase of **Case 1**, except that $\mathcal{G}$ does the following:

- Randomly guess $s^* \xleftarrow{\$} \{1, \cdots, M\}$ and $r^* \xleftarrow{\$} \{1, \cdots, N\}$ to be the target sender and the target receiver respectively, that $\mathcal{F}$ would like to forge, and then set $\overline{\mathbf{A}}_{s^*} := \overline{\mathbf{A}}$, $\overline{\mathbf{A}}'_{s^*} := \overline{\mathbf{A}}'$, $\mathbf{A}_{s^*} := [\overline{\mathbf{A}}_{s^*}|\mathbf{W}] \in \mathbb{Z}_q^{n \times m}$, $\mathbf{A}'_{s^*} := [\overline{\mathbf{A}}'_{s^*}|\mathbf{W}'] \in \mathbb{Z}_q^{n \times m}$. Recall that $m = \overline{m} + nk$.
- Repeat choosing $\mathbf{h}^{(j)} = (h_1^{(j)}, \cdots, h_n^{(j)}) \in \mathbb{Z}_q^n$ uniformly at random until $(1 + \sum_{i=1}^n x_i \cdot h_i^{(j_0)}) = 0$ for some $j_0 \in [Q]$ and $(1 + \sum_{i=1}^n x_i \cdot h_i^{(j)}) \neq 0$ for each $j \in [Q] \setminus \{j_0\}$. Let $\mathbf{h}^{(j_0)}$ to be the one corresponding to the target query (r^*, s^*, μ^*) and let $\mathbf{h}^* = (h_1^*, \cdots, h_n^*) \leftarrow \mathbf{h}^{(j_0)}$.
- Set $\mathbf{A}_{s^*,\mathbf{h}^*} = [\mathbf{A}_{s^*}|\mathbf{C}_0 + \sum_{i=1}^n h_i^* \cdot \mathbf{C}_i] = [\overline{\mathbf{A}}_{s^*}|\mathbf{W}| - \overline{\mathbf{A}}_{s^*}\mathbf{T}_{s^*}]$, where $\mathbf{T}_{s^*} := \mathbf{T}_{s^*,0} + \sum_{i=1}^n h_i^* \cdot \mathbf{T}_{s^*,i}$.
- Choose $\mathbf{e}^{*(1)} \leftarrow D_{\mathbb{Z}^{m+nk},\sigma_2}$, set $\mathbf{u} := \mathbf{A}_{s^*,\mathbf{h}^*} \cdot \mathbf{e}^{*(1)} \in \mathbb{Z}_q^n$ and send $\mathbf{u}$ to $\mathcal{F}$ as a uniformly random one.

Queries. For almost PKQ, SCQ and TGQ queries,$\mathcal{G}$ responds in the same way as in the **Queries** phase of **Case 1**, except that with the target query SCQ(r^*, s^*, μ^*), $\mathcal{G}$ responds as follows:

1. Produce $\overline{ct}^* = (\mathbf{c}_0^*, \overline{\mathbf{c}}_1^*, \mathbf{r}_e^*, \mathbf{c}_0'^*, \overline{\mathbf{c}}_1'^*, \mathbf{r}_e'^*)$ as usual.
2. Generate a signature on $\mu^*|pk_{r^*}|\overline{ct}^*$:
   (a) $\mathbf{r}_s^* \leftarrow \mathsf{SampleD}(\mathbf{T}_{\mathbf{B}}, \mathbf{B}, \mathbf{h}^* - f_{\overline{\mathbf{A}}_{s^*}}(H_3(\mu^*|pk_{r^*}|\overline{ct}^*)), \alpha q)$.
   (b) Set $(\mathbf{e}^{*(1)}, \mathbf{r}_s^*)$ to be the signature.
3. $\mathbf{c}_1^* = \overline{\mathbf{c}}_1^* + \mu \cdot \lfloor q/2 \rfloor, \quad \mathbf{c}_1'^* = \overline{\mathbf{c}}_1'^* + H(\mu) \cdot \lfloor q/2 \rfloor$.

4. Return $ct^{*(1)} = (\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{r}_e^*, \mathbf{r}_s^*, \mathbf{c}_0'^*, \mathbf{c}_1'^*, \mathbf{r}_e'^*, \mathbf{e}^{*(1)})$ to $\mathcal{F}$.

Forge. The forger $\mathcal{F}$ outputs the target receiver's key-pair (pk_{r^*}, sk_{r^*}) together with a new valid ciphertext $ct^{*(2)} = (\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{r}_e^*, \mathbf{r}_s^*, \mathbf{c}_0'^*, \mathbf{c}_1'^*, \mathbf{r}_e'^*, \mathbf{e}^{*(2)})$ on the message μ^*. Note that, $\mathbf{e}^{*(2)} \neq \mathbf{e}^{*(1)}$ while $\mathbf{c}_0^*, \mathbf{c}_1^*, \mathbf{r}_e^*, \mathbf{r}_s^*, \mathbf{c}_0'^*, \mathbf{c}_1'^*, \mathbf{r}_e'^*$ are unchanged due to the validity of $ct^{*(2)}$ corresponding to $\mathbf{A}_{s^*,\mathbf{h}^*}$.

Analysis. At the moment, $\mathcal{G}$ proceeds the following steps:

- Recall $\mathbf{A}_{s^*,\mathbf{h}^*} = [\mathbf{A}_{s^*}|\mathbf{C}_0 + \sum_{i=1}^n h_i^* \cdot \mathbf{C}_i] = [\overline{\mathbf{A}}_{s^*}|\mathbf{W}| - \overline{\mathbf{A}}_{s^*}\mathbf{T}_{s^*}]$.

- From $\mathbf{A}_{s^*,\mathbf{h}^*} \cdot \mathbf{e}^{*(1)} = \mathbf{u} \bmod q$ and $\mathbf{A}_{s^*,\mathbf{h}^*} \cdot \mathbf{e}^{*(2)} = \mathbf{u} \bmod q$, we have

$$[\overline{\mathbf{A}}_{s^*}|\mathbf{W}] \cdot \left(\begin{bmatrix} \mathbf{I}_{\overline{m}} & \mathbf{0} & -\mathbf{T}_{s^*} \\ \mathbf{0} & \mathbf{I}_{nk} & \mathbf{0} \end{bmatrix} \cdot (\mathbf{e}^{*(1)} - \mathbf{e}^{*(2)}) \right) = \mathbf{0} \pmod q.$$

- Let $\widehat{\mathbf{x}} := \begin{bmatrix} \mathbf{I}_{\overline{m}} & \mathbf{0} & -\mathbf{T}_{s^*} \\ \mathbf{0} & \mathbf{I}_{nk} & \mathbf{0} \end{bmatrix} \cdot (\mathbf{e}^{*(1)} - \mathbf{e}^{*(2)})$, then $[\overline{\mathbf{A}}|\mathbf{W}] \cdot \widehat{\mathbf{x}} = \mathbf{0} \pmod q$,. Hence, $\mathcal{G}$ gets a solution $\mathbf{x}$ to the SIS problem (2), i.e., $\mathbf{F} \cdot \mathbf{x} = \mathbf{0} \pmod q$, with $\mathbf{x} = \binom{\widehat{\mathbf{x}}}{\mathbf{0}}$, and

$$\begin{aligned} \|\mathbf{x}\| = \|\widehat{\mathbf{x}}\| &\leq s_1(\mathbf{T}_{s^*}) \cdot \|\mathbf{e}^{*(1)} - \mathbf{e}^{*(2)}\| \\ &\leq 2\sigma_1 \cdot \sigma_2 \cdot \sqrt{n+1} \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk}) \cdot \sqrt{m+nk}, \end{aligned}$$

  by Lemma 2.

It remains to prove that $\widehat{\mathbf{x}} \neq 0$ with overwhelming probability. Let $\mathbf{w} := \mathbf{e}^{*(1)} - \mathbf{e}^{*(2)} \neq \mathbf{0}$ and parse $\mathbf{w} = \left(\begin{smallmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \end{smallmatrix}\right)$. Then $\widehat{\mathbf{x}} = (\mathbf{w}_1 - \mathbf{T}_{s^*}\mathbf{w}_3, \mathbf{w}_2)$. Obviously, if $\mathbf{w}_2 \neq \mathbf{0}$ or $\mathbf{w}_3 := (w_1, \cdots, w_{nk}) = \mathbf{0}$ then $\widehat{\mathbf{x}} \neq 0$. Otherwise, i.e., $\mathbf{w}_2 = \mathbf{0}$ and $\mathbf{w}_3 \neq \mathbf{0}$, we will show that $\overline{\mathbf{A}}_{s^*}(\mathbf{w}_1 - \mathbf{T}_{s^*}\mathbf{w}_3) = \mathbf{0}$ happens only with negligible probability. Indeed, without loss of generality, assume that $w_{nk} \neq 0$ then $\overline{\mathbf{A}}_{s^*}(\mathbf{w}_1 - \mathbf{T}_{s^*}\mathbf{w}_3) = \mathbf{0}$ only if $\mathbf{t}_{nk} \sim D_{\Lambda_q^\perp(\overline{\mathbf{A}}_{s^*})+\mathbf{c},\sigma_1}$ and $\mathbf{t}_{nk} \cdot w_{nk} = \mathbf{y}$ for some $\mathbf{y} \in \Lambda_q^\perp(\overline{\mathbf{A}}_{s^*}) + \mathbf{c}$ for any $\mathbf{c}$ in $\mathrm{span}(\Lambda_q^\perp(\overline{\mathbf{A}}_{s^*}))$, where $\mathbf{t}_{nk}$ is the nk-th column of $\mathbf{T}_{s^*,0}$. Then by Lemma 1, such a $\mathbf{t}_{nk}$ exists with negligible probability.

In conclusion, we choose the common SIS problem $\mathsf{SIS}_{n,2\overline{m}+2nk+1,q,\beta}$ for both two cases (i.e., **Case 1** and **Case 2**) with $\beta := \max\{\beta_1, \beta_2\}$ which should be $\beta := 2\sigma_1 \cdot \sigma_2 \cdot \sqrt{n+1} \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk}) \cdot \sqrt{m+nk}$. □

6. Parameter Selection

- We take n as the security parameter.

- For the gadget-based trapdoor mechanism to work: Choose $q \geq 2, \overline{m} \geq 1$, $k = \lceil \log_2 q \rceil$, and $m = O(n \log q)$.

- Also choose $q, n, \overline{m}, \ell, Q$ such that H is one-way, H_1 is collision-resistant, H_3 is universal, $\mathcal{H}_{Wat}$ (in Lemma 12) is a family of abort-resistant hash functions, and the functions $f_{\overline{\mathbf{A}}_r}(\cdot) + f_{\mathbf{B}}(\cdot)$ are collision-resistance for any $\overline{\mathbf{A}}_r \xleftarrow{\$} \mathbb{Z}_q^{n\times\overline{m}}$ and any $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.

- By Theorem 3, in order for the decisional-LWE $\mathsf{dLWE}_{n,2(\overline{m}+\ell),q,\alpha q}$ to be hard: $q > 2\sqrt{n}/\alpha$ for $\alpha = 1/\mathsf{poly}(n) \in (0,1)$.

- For the Gaussian parameter σ_1 used in KeyGen (which follows GenTrap): $\sigma_1 \geq \omega(\sqrt{\log n})$. Note also that, all private keys for senders and users $\mathbf{T}_s, \mathbf{T}_r$ satisfy that $s_1(\mathbf{T}_s), s_1(\mathbf{T}_r) \leq \sigma_1 \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk})$ by Lemma 6.

- For the parameter α used in Invert: We should choose α such that $1/\alpha \geq 2\sqrt{5(s_1(\mathbf{R})^2+1)} \cdot \omega(\sqrt{\log n})$ by Lemma 7.

- For the Gaussian parameter σ_2 used in SampleD: $\sigma_2 \geq \sqrt{7(s_1(\mathbf{T}_s)^2+1)} \cdot \omega(\sqrt{\log n})$ (see [20, Section 5.4]).

- For the $\mathsf{SIS}_{n,2\overline{m}+2nk+1,q,\beta}$ problem has a solution and to be hard (in the SUF-iCMA security proof): $\beta \geq \sqrt{2\overline{m}+2nk+1} \cdot q^{n/(2\overline{m}+2nk+1)}$, $q \geq \beta \cdot \omega(\sqrt{n\log n})$, $\beta := 2\sigma_1 \cdot \sigma_2 \cdot \sqrt{n+1} \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk}) \cdot \sqrt{m+nk}$.

7. Insecurity of the Signcryption by Lu et al. [18]

7.1. Description

The signcryption by Lu et al. [18] (called Lu-SC) exploits the basis-based trapdoor mechanism by [30] which consists of algorithms TrapGen, ExtBasis, SamplePre. The Lu-SC includes the following algorithms:

- Setup(1^n): On input the security parameter n, performs the following:

  1. Generate common parameters: $q = \mathsf{poly}(n)$, $m = \lceil 6n\log q \rceil$, $\tilde{L} = O(\sqrt{n\log q})$, $\sigma \geq \tilde{L}\omega(\sqrt{\log m})$, error rate $\alpha = 1/\mathsf{poly}(n)$ such that $\alpha q > 2\sqrt{n}$.
  2. Sample randomly and independently matrices $\mathbf{C}_0, \cdots, \mathbf{C}_\tau \in \mathbb{Z}_q^{n\times m}$.
  3. A collision-resistant hash function $H_1 : \{0,1\}^* \to \{0,1\}^\tau$, a universal hash function $H_2 : \{0,1\}^* \to \mathbb{Z}_q^n$.
  4. A message space $\mathcal{M} = \mathbb{Z}_q^n$.

- KeyGen(n): Do the following:

  1. Use TrapGen(1^n) to generate the matrix-pairs $(\mathbf{A}_R, \mathbf{T}_R)$, $(\mathbf{A}_S, \mathbf{T}_S)$, where each pair belongs to $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{m\times m}$.
  2. Sample randomly matrices $\mathbf{B}_R, \mathbf{B}_S$, each from $\mathbb{Z}_q^{n\times m}$.

3. Return
   - $\mathsf{pk}_R := (\mathbf{A}_R, \mathbf{B}_R)$, and $\mathsf{sk}_R := \mathbf{T}_R$ as public key and secret key for a receiver $\mathcal{R}$,
   - $\mathsf{pk}_S := (\mathbf{A}_S, \mathbf{B}_S)$, and $\mathsf{sk}_S := \mathbf{T}_S$ as public key and secret key for a sender $\mathcal{S}$.

- $\mathsf{SignCrypt}(\mu, \mathsf{sk}_S, \mathsf{pk}_R)$: On input a plaintext $mu \in \mathcal{M}$, $\mathsf{pk}_R := (\mathbf{A}_R, \mathbf{B}_R)$, $\mathsf{sk}_S := \mathbf{T}_S$, do the following:

  1. $\mathbf{h} = (h_i)_{i\in[\tau]} = H_1(\mu, \mathsf{pk}_R) \in \mathbb{Z}_q^{\tau}$.
  2. $\mathbf{F}_\mu := [\mathbf{A}_S||\mathbf{C}_0 + \sum_{i=1}^{\tau}(-1)^{h_i}\mathbf{C}_i] \in \mathbb{Z}_q^{n\times 2m}$.
  3. Compute $\mathbf{T}_\mu$ as the short basis for $\Lambda_q^{\perp}(\mathbf{F}_\mu)$ from $\mathbf{T}_S$ using $\mathsf{ExtBasis}$; $\mathbf{v} \leftarrow \mathsf{SamplePre}(\mathbf{F}_\mu, \mathbf{T}_\mu, \mathbf{0}, \sigma)$, $\mathbf{v} \in \mathbb{Z}_q^{2m}$, i.e., $\mathbf{F}_\mu{\cdot}\mathbf{v} = \mathbf{0} \pmod{q}$.
  4. $\mathbf{t} = H_2(\mu, \mathsf{pk}_S, \mathsf{pk}_R, \mathbf{v})$ and $\mathbf{c} := \mathbf{t} + \mu \bmod q$. [2].
  5. $\mathbf{e} \leftarrow \widetilde{\Psi}_\alpha^{2m}$, $\mathbf{b}_1 = [\mathbf{A}_R||\mathbf{C}_0]^T \cdot \mathbf{t} + \mathbf{v} \in \mathbb{Z}_q^{2m}$, $\mathbf{b}_2 = [\mathbf{B}_R||\mathbf{C}_1]^T \cdot \mathbf{t} + \mathbf{e} \in \mathbb{Z}_q^{2m}$.
  6. Output ciphertext $\mathsf{CT} = (\mathbf{c}, \mathbf{b}_1, \mathbf{b}_2)$.

- $\mathsf{UnSignCrypt}(\mathsf{CT}, \mathsf{pk}_S, \mathsf{sk}_R)$: On input $\mathsf{pk}_S := (\mathbf{A}_R, \mathbf{B}_R)$, $\mathsf{sk}_R := (\mathbf{T}_R)$, $\mathsf{CT} := (\mathbf{c}, \mathbf{b}_1, \mathbf{b}_2)$, do the following:

  1. Compute $\mathbf{t}$ and $\mathbf{v}$ from $\mathbf{b}_1$ using $\mathsf{ExtBasis}$ and Invert with the help of $\mathbf{T}_R$
  2. Compute $\mathbf{e} = \mathbf{b}_2 - [\mathbf{B}_R||\mathbf{C}_1]^T \cdot \mathbf{t}$ and check whether $0 < \|\mathbf{e}\| \leq \sigma\sqrt{2m}$ or not. If not, reject it and return $\perp$
  3. Compute $\mu = \mathbf{c} - \mathbf{t} \bmod q$ and $\mathbf{h} = (h_i)_{i\in[\tau]} = H_1(\mu, \mathsf{pk}_R)$
  4. Check whether $\mathbf{t} = H_2(\mu, \mathsf{pk}_S, \mathsf{pk}_R, \mathbf{v})$ or not. If not, reject it and output $\perp$
  5. If $\mathbf{v} \in \mathbb{Z}^{2m}$ and $0 < \|\mathbf{v}\| \leq \sigma\sqrt{2m}$ and $[\mathbf{A}_S||\mathbf{C}_0 + \sum_{i=1}^{\tau}(-1)^{h_i}\mathbf{C}_i] \cdot \mathbf{v} = \mathbf{0} \pmod{q}$ output μ; otherwise, output $\perp$.

7.2. An Attack against IND-CPA

Recall that, in the challenge phase of the IND-CPA security, the adversary submits two plaintexts μ_0^*, μ_1^* together with the target public key pk_{S^*}. The challenger then chooses uniformly at random a bit $b \in \{0, 1\}$ and returns the challenge ciphertext $\mathsf{CT}^* \leftarrow \mathsf{SignCrypt}(\mu_b^*, \mathsf{sk}_{S^*}, \mathsf{pk}_{R^*})$ back to the adversary. The adversary wins the game if he can guess correctly the bit b.

Given the challenge ciphertext $\mathsf{CT}^* = (\mathbf{c}^*, \mathbf{b}_1^*, \mathbf{b}_2^*)$ of a plaintext either μ_0^* or μ_1^*, the adversary is able to check the following conditions:

[2] Actually, in [18], the authors wrote $\mathbf{c} := \mathbf{t} \oplus \mu$. However, since μ and $\mathbf{t}$ belong to $\mathbb{Z}_q^n$, we think that it should be $\mathbf{c} := \mathbf{t} + \mu \bmod q$.

- $\mathbf{v} = \mathbf{b}_1 - [\mathbf{A}_{R*}|\mathbf{C}_0]^T \cdot (\mathbf{c} - \mu_b^*)$, and $\mathbf{e} = \mathbf{b}_2 - [\mathbf{B}_{R^*}|\mathbf{C}_1]^T \cdot (\mathbf{c} - \mu_b^*)$ are small,
- $\mathbf{c} - \mu_b^* = H_2(\mu_b^*, \mathsf{pk}_{S^*}, \mathsf{pk}_{R^*}, \mathbf{v})$,
- $[\mathbf{A}_{S^*}|\mathbf{C}_0+\sum_{i=1}^{\tau}(-1)^{h_i}\mathbf{C}_i]\mathbf{v} = \mathbf{0} \mod q$, where $(h_i)_{i\in[\tau]} = H_1(\mu_b^*, \mathsf{pk}_{R^*})$.

We see that the correct b satisfies all above conditions, whilst the incorrect b does not meet all the conditions with high probability. Then the adversary is able to win the IND-CPA security game with high probability.

Moreover, in the case that both $b = 0$ and $b = 1$ satisfy all the conditions, then the adversary is able to find a collision of H_2.

8. Conclusions

In this work, we constructed a lattice-based signcryption scheme associated with a capacity of equality testing in the standard model. To the best of our knowledge, this scheme is the first post-quantum signcryption with equality test in the literature. The proposed scheme satisfies confidentiality (IND-iCCA1 and OW-iCCA1) and the (strong) unforgeability under chosen message attack (SUF-iCMA) against insider attacks at the same time in which the former is based on the decisional-LWE assumption and the latter is guaranteed by the hardness of the SIS problem. Our main tool in the construction is the gadget-based trapdoor technique introduced in [20]. We also showed that some lattice-based signcryptions in the literature neither are secure nor work correctly.

Finding a better way to simplify the equality test mechanism would be an interesting future task. Additionally, a signcryption with multi-ciphertext equality test and/or flexible multi-ciphertext equality test as have been done for public key encryption in [31] could also be a good topic in the future.

References

[1] Y. Zheng, Digital signcryption or how to achieve cost(signature & encryption) $\leq$ cost(signature) + cost(encryption), in: Advances in Cryptology–CRYPTO 1997, Springer-Verlag, 1997, pp. 165–179. doi:10.1007/BFb0052234.

[2] G. Yang, C. H. Tan, Q. Huang, D. S. Wong, Probabilistic public key encryption with equality test, in: J. Pieprzyk (Ed.), Topics in Cryptology - CT-RSA 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 119–131. doi:10.1007/978-3-642-11925-5_9.

[3] H. T. Lee, S. Ling, J. H. Seo, H. Wang, T.-Y. Youn, Public key encryption with equality test in the standard model, Information Sciences 516 (2020) 89 – 108. doi:10.1016/j.ins.2019.12.023.

[4] X.-J. Lin, L. Sun, H. Qu, Generic construction of public key encryption, identity-based encryption and signcryption with equality test, Information Sciences 453 (2018) 111 – 126. doi:10.1016/j.ins.2018.04.035.

[5] D. H. Duong, K. Fukushima, S. Kiyomoto, P. S. Roy, W. Susilo, A Lattice-Based Public Key Encryption with Equality Test in Standard Model, in: J. Jang-Jaccard, F. Guo (Eds.), Information Security and Privacy, Springer International Publishing, Cham, 2019, pp. 138–155. doi:10.1007/978-3-030-21548-4_8.

[6] D. H. Duong, K. Fukushima, S. Kiyomoto, P. S. Roy, A. Sipasseuth, W. Susilo, Lattice-based public key encryption with equality test supporting flexible authorization in standard model (2020). arXiv:2005.05308.

[7] Q. Tang, Towards Public Key Encryption Scheme Supporting Equality Test with Fine-Grained Authorization, in: U. Parampalli, P. Hawkes (Eds.), Information Security and Privacy, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 389–406.

[8] D. H. Duong, H. Q. Le, P. S. Roy, W. Susilo, Lattice-Based IBE with Equality Test in Standard Model, in: R. Steinfeld, T. H. Yuen (Eds.), Provable Security, Springer International Publishing, Cham, 2019, pp. 19–40.

[9] G. L. D. Nguyen, W. Susilo, D. H. Duong, H. Le, F. Guo, Lattice-Based IBE with Equality Test Supporting Flexible Authorization in the Standard Model, in: K. Bhargavan, E. Oswald, M. Prabhakaran (Eds.), Progress in Cryptology – INDOCRYPT 2020, Springer International Publishing, Cham, 2020, pp. 624–643.

[10] Y. Wang, H. H. Pang, R. H. Deng, Y. Ding, Q. Wu, B. Qin, Securing messaging services through efficient signcryption with designated equality test, Information Sciences 490 (2019) 145–165.

[11] H. Xiong, Y. Zhao, Y. Hou, X. Huang, C. Jin, L. Wang, S. Kumari, Heterogeneous Signcryption with Equality Test for IIoT environment, IEEE Internet of Things Journal (2020) 1–1.

[12] P. W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM Journal on Computing 26 (5) (1997) 1484–1509. doi:10.1137/s0097539795293172. URL http://dx.doi.org/10.1137/S0097539795293172

[13] J. Malone-Lee, W. Mao, Two Birds One Stone: Signcryption Using RSA, in: M. Joye (Ed.), Topics in Cryptology — CT-RSA 2003, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 211–226. doi:10.1007/3-540-36563-X_14.

[14] X. Boyen, Multipurpose Identity-Based Signcryption, in: D. Boneh (Ed.), Advances in Cryptology - CRYPTO 2003, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 383–399. doi:10.1007/978-3-540-45146-4_23.

[15] F. Li, F. Muhaya, M. Khan, T. Takagi, Lattice-based Signcryption, Concurrency and Computation: Practice and Experience 25 (14) (2012) 2112–2122. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.2826, doi:10.1002/cpe.2826.

[16] F. Gérard, K. Merckx, SETLA: Signature and Encryption from Lattices, 2018, pp. 299–320. doi:10.1007/978-3-030-00434-7_15.

[17] X. Yang, H. Cao, W. Li, H. Xuan, Improved lattice-based signcryption in the standard model, IEEE Access 7 (2019) 155552–155562.

[18] X. Lu, Q. Wen, Z. Jin, L. Wang, C. Yang, A lattice-based signcryption scheme without random oracles, Frontiers of Computer Science 8 (4) (2014) 667–675. doi:10.1007/s11704-014-3163-1.
URL https://doi.org/10.1007/s11704-014-3163-1

[19] S. Sato, J. Shikata, Lattice-based signcryption without random oracles, in: T. Lange, R. Steinwandt (Eds.), Post-Quantum Cryptography, Springer International Publishing, Cham, 2018, pp. 331–351. doi:10.1007/978-3-319-79063-3_16.

[20] D. Micciancio, C. Peikert, Trapdoors for lattices: Simpler, tighter, faster, smaller, in: D. Pointcheval, T. Johansson (Eds.), Advances in Cryptology – EUROCRYPT 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 700–718. doi:10.1007/978-3-642-29011-4_41.

[21] S. Agrawal, D. Boneh, X. Boyen, Efficient Lattice (H)IBE in the Standard model, in: H. Gilbert (Ed.), Advances in Cryptology – EUROCRYPT 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 553–572. doi:10.1007/978-3-642-13190-5_28.

[22] C. Peikert, A. Rosen, Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices, in: S. Halevi, T. Rabin (Eds.), Theory of Cryptography, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 145–166.

[23] D. Micciancio, O. Regev, Worst-case to average-case reductions based on gaussian measures, SIAM J. Comput. 37 (1) (2007) 267–302. doi:10.1137/S0097539705447360.

[24] O. Regev, On Lattices, Learning with Errors, Random Linear Codes, and Cryptography, in: In STOC, ACM Press, 2005, pp. 84–93.

[25] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé, Classical Hardness of Learning with Errors, in: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing, STOC '13, ACM, New York, NY, USA, 2013, pp. 575–584. doi:10.1145/2488608.2488680.

[26] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for Hard Lattices and New Cryptographic Constructions, Cryptology ePrint Archive, Report 2007/432, https://eprint.iacr.org/2007/432 (2007).

[27] L. Ducas, D. Micciancio, Improved Short Lattice Signatures in the Standard Model, in: J. A. Garay, R. Gennaro (Eds.), Advances in Cryptology – CRYPTO 2014, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 335–352.

[28] J. H. An, Y. Dodis, T. Rabin, On the Security of Joint Signature and Encryption, in: L. R. Knudsen (Ed.), Advances in Cryptology — EUROCRYPT 2002, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 83–107.

[29] J. Baek, R. Steinfeld, Y. Zheng, Formal Proofs for the Security of Signcryption, Journal of Cryptology 20 (2) (2007) 203–235. doi:10.1007/s00145-007-0211-0.
URL https://doi.org/10.1007/s00145-007-0211-0

[30] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for Hard Lattices and New Cryptographic Constructions, in: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, STOC '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 197–206. doi:10.1145/1374376.1374407.
URL https://doi.org/10.1145/1374376.1374407

[31] W. Susilo, F. Guo, Z. Zhao, G. Wu, PKE-MET: Public-Key Encryption with Multi-Ciphertext Equality Test in Cloud Computing, IEEE Transactions on Cloud Computing 14 (8) (2020) 1–1. doi:10.1109/tcc.2020.2990201.