Lattice-based Signcryption with Equality Test in Standard Model

Huy Quoc Le\textsuperscript{a,*}, Dung Hoang Duong\textsuperscript{a}, Partha Sarathi Roy\textsuperscript{a}, Willy Susilo\textsuperscript{a}, Kazuhide Fukushima\textsuperscript{b}, Shinsaku Kiyomoto\textsuperscript{b}

\textsuperscript{a} Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong NSW 2522, Australia

\textsuperscript{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 \cite{Zheng1999}, simultaneously 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] just to name a few. 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 [8], industrial Internet of Things [9].

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 [10]. 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 [11] in 2003 presented a RSA-based signcryption scheme in the random oracle model. In the same year, Boyen [12] 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. [8] presented a concrete SCET construction, which the authors call public key signcryption scheme with designated equality test on ciphertexts. However, the primitive in [8] is based on bilinear groups. More recently, Xiong et al. [9] 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 [13], [14], [15], [16] and [17]. Li et al. [13] built a lattice-based signcryption scheme that are only secure in the random oracle model (ROM). Gérard and Merckx [14] 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. [15] 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 [15] offers the IND-CCA2 security and and existential unforgeability under an adaptive chosen-message attack (EUF-aCMA). Lu et al. [16] 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 [17] 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. [16] can be easily broken even by a CPA attack (see Section 7 below) while the work [17] has some serious flaws described as follows.

Description and flaws of the signcryption proposed by [17]. The work [17] exploits the gadget-based trapdoor mechanism proposed by [18], namely, the algorithms GenTrap, Invert, SampleD (see Definition 5 and Lemma 7 below for further details). Then, for each receiver we generate the public key \( \mathbf{pk}_r = A_r \), and the private key \( \mathbf{sk}_r = T_r \) which is a \( \mathbf{G} \)-trapdoor for \( A_r \) with tag \( H = 0 \), i.e., \( A_r \left[ \begin{array}{c} T_r \\ 1 \end{array} \right] = 0 \mod q \), where \( \mathbf{G} \) is the gadget matrix specified in [18, Section 4]. The same thing \( \mathbf{pk}_s = A_s \), \( \mathbf{sk}_r = T_s \) is done for each sender except that \( H = I \), i.e., \( A_s \left[ \begin{array}{c} T_s \\ 1 \end{array} \right] = \mathbf{G} \mod q \).

For signcrypting a plaintext \( \mu \), one utilizes the Dual-Regev framework based on the LWE problem, i.e., one computes \( (c_0)^t := s^{t}A_r^{t} + (x_0)^t \) and \( (c_1)^t := s^{t}U + (x_1)^t \). Here \( U \) is a uniform matrix while the matrix \( A_r^{t} \) depends on \( A_r^{t} \) and some vector tag \( \mathbf{t} \) which relates to the public key of the receiver \( \mathbf{pk}_r \), to the public key of the sender and to a random small vector \( r_e \). Then one signs on the tuple \( (\mu|\mathbf{pk}_r|\mathbf{ct}) \), where \( \mathbf{ct} = (\mathbf{c}_0, \mathbf{c}_1, r_e) \), to get the signature \( (e, r_s) \). To do that, one calculates a vector tag \( \mathbf{h} \) using \( \mathbf{pk}_s, \mathbf{pk}_r, (\mu|\mathbf{pk}_r|\mathbf{ct}) \) and a random small vector \( r_s \), and then signs on \( \mathbf{h} \) using the same way as the signature scheme in [18, Section 6.2]. Finally, the signcryption finishes by computing \( c_0 := \mathbf{c}_0 + r_s, c_1 := \mathbf{c}_1 + \mu \cdot \lfloor q/2 \rfloor \) and outputting the ciphertext \( \mathbf{ct} = (c_0, c_1, r_e, e) \).

In the unsigncryption algorithm, one can use SampleD to obtain a matrix \( \mathbf{E} \) such that \( A_r^{t} \mathbf{E} = U \mod q \) which helps us to recover \( \mu \). However, in this algorithm, one also needs to verify whether the ciphertext is valid or not which needs to recover \( \mathbf{ct} = (\mathbf{c}_0, \mathbf{c}_1, r_e) \) from \( \mathbf{ct} \) first. The way of [17] to do that is to run Invert on input \( c_0 \) to get \( r_s \) and then compute \( \mathbf{c}_0 = c_0 - r_s, \mathbf{c}_1 = c_1 - \mu \cdot \lfloor q/2 \rfloor \). Unfortunately, we can see that this is not correct since Invert will actually output the sum \( x_0 + r_s \) instead of \( r_s \).

One more flaw is that the dimensions of some matrices and some vectors in the signcryption of [17] 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 [17, Theorem
2], after showing that $A_S \cdot z = 0 \mod q$, the authors do not prove why $z \neq 0$.

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 [16] and [17] 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
[17] and the recent method of Duong et al. [5] for equality test. We have
shown above that [17] does not work correctly. Fortunately, we success-
fully fix this error simply by in the signcryption algorithm setting $c_0 := \overline{c}_0$ instead of $c_0 := c_0 + r_s$ and outputting $ct = (c_0, c_1, r_e, r_s, e)$ instead of $ct = (c_0, c_1, r_e, e)$. Also, to fix the dimension-related flaw in [17], 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 plain-
text $\mu$. This releases some things named $c'_0, \overline{c}'_1, r'_e$, corresponding to $c_0, \overline{c}_1, r_e$, for $\mu$. Note that, the signing phase (which produces the signature $(e, r_e)$) now runs on input $(\mu, pk, ct)$ with $\overline{ct} = (c_0, \overline{c}_1, r_e, c'_0, \overline{c}'_1, r'_e)$ instead of $\overline{ct} = (c_0, \overline{c}_1, r_e)$. Therefore, the final ciphertext for the proposed SCET is
$ct = (c_0, c_1, r_e, r_s, c'_0, \overline{c}'_1, r'_e, e)$. 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 $\mu$.

Furthermore, for the security proof, we also utilize the so-called abort-
resistant hash functions defined by [19, Section 7] provided in Lemma 12.
Also note that, the presence of $B, r_e, r'_e, r_s$ helps us to simulate the responses
to the adversary’s queries.

Organisation. In Section 2, we give a background of lattices. The frame-
work of signcryption with equality test will be provided in Section 3. Section
4 is for our lattice-based signcryption construction. The security of the pro-
posed 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-
tion construction proposed by Lu et al. [16] in Section 7. In Section 8 we
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 [13]| LWE & SIS        | IND-CCA2         | ROM            | ×       | ×              |
| Lu [16]| Not secure even with IND-CPA (see Section 7) |                  |                |         |                |
| Sato [17]|                |                  |                |         |                |
| Gérard [14]| RLWE & SIS      | IND-CPA          | ROM            | ×       | ×              |
| Yang [15]| RLWE & ideal-SIS| IND-CCA2         | EUF-aCMA       |         |                |
| This work| LWE & SIS        | IND-iCCA1        | OW-iCCA1       | ✓       | ✓              |
|         |                  | SUF-iCMA         | SDM            | ✓       | ✓              |

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 \(D_{\alpha q}\).

make some conclusions on our work.

2. Preliminaries

Throughout this work, the norm \(\|S\|\) of a set of vectors \(S = \{s_1, \cdots, s_n\}\) is computed as \(\max_{i \in [n]} \|s_i\|\).

**Lattices.** Given a matrix \(B = [b_1, \cdots, b_m] \in \mathbb{R}^{n \times m}\) of \(m\) linearly independent vectors, the set \(L(B) := \{\sum_{i=1}^{m} b_i z_i : z_i \in \mathbb{Z}\}\) is called a lattice of basis \(B\). In this work, we focus on the so-called \(q\)-ary lattices: \(\Lambda_q^+(A) = \{e \in \mathbb{Z}^m : A e = 0 \pmod{q}\}\), and \(\Lambda_q^u(A) = \{e \in \mathbb{Z}^m \text{ s.t. } A e = u \pmod{q}\}\), where \(A \overset{\$}{\in} \mathbb{Z}^{n \times m}\) is a random matrix. Note that, if \(t \in \Lambda_q^u(A)\) then \(\Lambda_q^u(A) = t + \Lambda_q^+(A)\).

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

**Gaussians.** Let \( m \geq 1 \), a vector \( c \in \mathbb{R}^m \) and a positive parameter \( s \), for \( x \in \mathbb{R}^m \) define \( \rho_{s,c}(x) = \exp(-\pi \|x - c\|^2/s^2) \). The continuous Gaussian distribution on \( \mathbb{R}^m \) with mean \( c \) and with width parameter \( s \) is proportional to \( \rho_{s,c}(x) \). Let \( T = \mathbb{R}/\mathbb{Z} \) be the additive group of real numbers modulo 1. Given \( \alpha > 0 \) and \( m = 1 \), we denote by \( \Psi_\alpha \) the continuous Gaussian distribution on \( T \) 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 \( \Psi_{\alpha q} \) is defined by sampling \( X \leftarrow \Psi_\alpha \) then outputting \( [x \cdot X] \mod q \).

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 \( c \in \mathbb{R}^m \) and a positive parameter \( s \), define \( \rho_{s,c}(\Lambda) := \sum_{x \in \Lambda} \rho_{s,c}(x) \). The discrete Gaussian distribution over \( \Lambda \) centered at \( c \in \mathbb{Z}^m \) with width parameter \( s \) is defined by: \( \forall x \in \Lambda \), \( D_{\Lambda,s,c}(x) := \rho_{s,c}(x)/\rho_{s,c}(\Lambda) \). If \( c = 0 \), we just simply write \( \rho_s, D_{\Lambda,s} \). If \( \Lambda = \mathbb{Z} \), we can write \( D_{\Lambda,s} \) as \( D_s \).

Note that in Definition 2 \( \Lambda \) 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 \( c \in \mathbb{R}^n \) and any \( y \in \Lambda + c \), \( \Pr[x \leftarrow D_{\Lambda + c,s} : x = y] \leq 2^{-n} \cdot \frac{1+\epsilon}{\epsilon} \).

**Lemma 2.** [21] Lemma 4.4 Let \( q > 2 \) and let \( A \) be a matrix in \( \mathbb{Z}^{n \times m}_q \) with \( m > n \). Let \( T_A \) be a basis for \( \Lambda_q^\perp(A) \). Then, for \( s \geq \|T_A\| : \omega(\sqrt{\log n}) \),

\[ \Pr[x \leftarrow D_{\Lambda_q^\perp(A),s} : \|x\| > s\sqrt{m}] \leq \negl(n). \]

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

**Definition 3 (LWE, [22]).** The decisional-LWE problem (dLWE\(_{n,q,\chi}\)) asks to distinguish a pair \( (a, b) \leftarrow L_{s,\chi} \) from a pair \( (a, b) \xleftarrow{\$} \mathbb{Z}_q^n \times \mathbb{Z}_q \).

In the case that \( \chi = \tilde{\Psi}_{\alpha q} \), we instead use the notations dLWE\(_{n,q,\alpha}\) and sLWE\(_{n,q,\alpha}\), and generally mention them as the LWE\(_{n,q,\alpha}\). Regarding the hardness of LWE, we have the following result:
Theorem 3 ([23, 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 \( \text{GapSVP}_{O(n/\alpha)} \) to \( \text{LWE}_{n,q,\alpha} \). 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 \( A \) $\rightarrow \mathbb{Z}_{n \times m}^{q} \), and a positive real number \( \beta \), the short integer problem \( \text{SIS}_{q,n,m,\beta} \) is to find a non-zero vector \( z \in \mathbb{Z}_{m} \setminus \{0\} \) satisfying that \( Az = 0 \) (mod \( q \)) and \( \|z\| \leq \beta \).

It is shown in [21] and then in [24] that for large enough \( q \), solving \( \text{SIS} \) is as hard as solving \( \text{SIVP} \) problem. Formally,

Lemma 4 ([24, Proposition 5.7]). For any poly-bounded \( m \), and \( \beta = \text{poly}(\sqrt{n}) \), and for any prime \( q \geq \beta \cdot \omega(\sqrt{n \log n}) \) the average-case problem \( \text{SIS}_{n,m,q,\beta} \) is as hard as \( \text{SIVP}_{\gamma} \) in the worst-case to within certain \( \gamma = \tilde{O}(\beta \sqrt{n}) \) factor.

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

Lemma 5 ([21, Lemma 5.2]). For any \( q \), \( A \in \mathbb{Z}_{n \times m}^{q} \), and \( \beta \geq \sqrt{mq^{n/m}} \), the \( \text{SIS}_{n,m,q,\beta} \) admits a solution.

Gadget-based Trapdoor. We will recall the notion of \( G \)-trapdoor and its related algorithms.

Definition 5 (G-trapdoors, [18, Definition 5.2]). Let \( n, q, m, k \) be positive integers and \( A \in \mathbb{Z}_{q \times m}^{n} \), \( G \in \mathbb{Z}_{q \times nk}^{n} \) be matrices with \( m \geq nk \). Let \( H \in \mathbb{Z}_{n \times n}^{q} \) be some invertible matrix. The \( G \)-trapdoor for \( A \) with tag \( H \) is a matrix \( R \in \mathbb{Z}_{(m-nk) \times nk}^{\} \) such that \( A [R_{1:nk}] = HG \) (mod \( q \)).

The largest singular value \( s_{1}(R) \) is used to measure the quality of a \( G \)-trapdoor \( R \) by its. Note that, \( s_{1}(R) \) is essentially small as claimed in the following lemma.

Lemma 6 ([18, Lemma 2.9]). Let \( D_{q \times m}^{\sigma} \) be a discrete Gaussian distribution with parameter \( \sigma \) and \( R \leftarrow D_{\sigma}^{n \times m} \). Then with overwhelming probability \( s_{1}(R) \leq \sigma \cdot \frac{1}{\sqrt{2\pi} \cdot (\sqrt{n} + \sqrt{m})} \).

Let \( k = \lceil \log_{2} q \rceil \), and \( g^{l} = (1, 2, 4, \ldots, 2^{k-1}) \in \mathbb{Z}_{q}^{k} \). We will be working with \( G = I_{n} \otimes g^{l} \in \mathbb{Z}_{q \times nk}^{n} \), where \( \otimes \) denotes the tensor product. Further details can be found in [18]. We will exploit the following algorithms for the proposed SCET construction.

Lemma 7. Let \( q \geq 2, m \geq 1, k = \lceil \log_{2} q \rceil \), and \( m = \overline{m} + nk = O(n \log q) \).
1. (A, R) $\leftarrow \text{GenTrap}(n, m, q, \sigma)$ \[18\] \textbf{Algorithm 1}: On input integer $n, m, q, \sigma$, GenTrap chooses a uniform matrix $A \in \mathbb{Z}_q^{n \times m}$ and a matrix $H \in \mathbb{Z}_q^{n \times k}$, then outputs a random matrix $A = [\overline{A} | H - \overline{A}R]$ and a $G$-trapdoor $R \sim D_{\sigma}^{m \times nk}$ with tag $H$. The condition for Gaussian parameter $\sigma$ is that for any $\epsilon \in (0, 1)$, $\sigma \geq \eta_{\epsilon}(\mathbb{Z})$, i.e., $\sigma \geq \sqrt{\ln(2(1+1/\epsilon))}/\pi$. Moreover, there exists $\epsilon = \epsilon(n)$ negligible for which $\sigma \geq \omega(\sqrt{\log n})$. Note also that, $s_1(R) \leq \sigma \cdot \sqrt{\frac{1}{2\pi}} \cdot (\sqrt{m} + \sqrt{nk})$ by Lemma 8.

2. $e \leftarrow \text{SampleD}(A, R, H, u, \sigma)$ \[18\] \textbf{Algorithm 3}: On input a matrix $A \in \mathbb{Z}_q^{n \times (m+nk)}$ and its $G$-trapdoor $R \in \mathbb{Z}_q^{m \times nk}$, an invertible matrix $H \in \mathbb{Z}_q^{n \times n}$, a uniform vector $u \in \mathbb{Z}_q^n$ and a Gaussian parameter $\sigma$, SampleD outputs a vector $e \in Z^{m+nk} \sim D_{\Lambda_q^2(A, \sigma)}$. The condition for $\sigma$ is that $\sigma \geq \sqrt{7}(s_1(R)^2 + 1) \cdot \omega(\sqrt{\log n})$ (see \[18\] Section 5.4).

3. $(s, e) \leftarrow \text{Invert}(R, A, b^f = s^fA + e^f)$ \[18\] \textbf{Algorithm 2}: On input a uniform matrix $A$ and its $G$-trapdoor $R$, and a vector $b$ such that $b^f = s^fA + e^f$, Invert returns $(s$ and $e)$. Note that if $e \leftarrow D_{\alpha \cdot \rho_{k-1}}$ and $1/\alpha \geq 2\sqrt{5s_1(R)^2 + 1} \cdot \omega(\sqrt{\log n})$ then Invert succeeds with overwhelming probability over the choice of $e$ (see \[18\] Theorem 5.4).

We adapt Lemma 6 in \[24\] for scalar tags (i.e., $H = x \times I_n$ 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.

\textbf{Lemma 8 (Adapted from \[25\] Lemma 6).} For $i = 0, \ldots, n$, let $T^{(i)}$ be $G$-trapdoor for $[A | A^{(i)}] \in \mathbb{Z}_q^{n \times (m-k)} \times \mathbb{Z}_q^{n \times k}$ with tag $H^{(i)} = x_iI_n$ for some $x_i \in \mathbb{Z}_q \setminus \{0\}$. Then any linear combination $T = \sum_{i=1}^{n} h_iT^{(i)}$ with $h_i \in \mathbb{Z}_q$ is a $G$-trapdoor for $[A | \sum_{i=1}^{n} h_iA^{(i)}]$ with tag $H = \sum_{i=1}^{n} h_iH^{(i)} = (\sum_{i=1}^{n} h_ix_i)I_n \neq 0$.

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 \[26\]. 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 \[27\] 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:

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

- **KGr**(pp) (resp., **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 R (resp., a public key pk_s and a private key sk_s for a sender S).

- **SC**(pk_r, sk_s, µ) 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.

- **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 ⊥.

- **Tag**(sk_r) is a DPT algorithm that on input a private key sk_r of a receiver to output a tag tg_r.

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

1. \Pr[µ = **USC**(sk_r, pk_s, **SC**(pk_r, sk_s, µ))] = 1 − 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 **USC**(sk_{r_1}, pk_{s_1}, ct_1) = **USC**(sk_{r_2}, pk_{s_2}, ct_2) ≠ ⊥, then
   \Pr[**Test**(tg_1, ct_1, tg_2, ct_2) = 1] = 1 − 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 \( \text{USC}(sk_{r_1}, pk_{s_1}, ct_1) \neq \text{USC}(sk_{r_2}, pk_{s_2}, ct_2) \), then

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

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 adversaries, 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 adversary** 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 \( A_1 \) playing the \( \text{INDCCA}_1^{\text{SCET}} \) game is negligible: \( \text{Adv}_{A_1}^{\text{IND-iCCA1}}(\lambda) := |\Pr[\text{INDCCA}_1^{\text{SCET}} \Rightarrow 1] - 1/2| \leq \text{negl}(\lambda) \).

The \( \text{INDCCA}_1^{\text{SCET}} \) game is defined as follows:

**Setup.** The challenger \( C \) first runs \( \text{Setup}(1^\lambda) \) to have the set of public parameters \( pp \). Then \( C \) runs \( \text{KGr}(pp) \) to get \( (pk_r, sk_r) \) for \( r \in [N] \), and...
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 \(A_1\). Let \(r^* \in [N]\) be the index of the target receiver.

**Phase 1.** \(A_1\) adaptively makes a polynomially bounded number of the following queries:

- Private key query PKQ\((r)\): If \(r = r^*\), the challenger \(C\) rejects the query. Otherwise, \(C\) returns the private key \(sk_r\) of the receiver \(R_r\) to \(A_1\).
- Signcryption query SCQ\((r, s, \mu)\): The challenger \(C\) sends the output \(ct\) of \(SC(pk_r, sk_s, \mu)\) back to \(A_1\).
- Unsigncryption query USQ\((r, s, ct)\): \(C\) sends the output of \(USC(sk_r, pk_s, ct)\) back to \(A_1\).
- Tag query TGQ\((r)\): If \(r = r^*\), the challenger \(C\) rejects the query. Otherwise, \(C\) in turn sends the output \(tg_r\) of \(Tag(sk_r)\) back to \(A_1\).

**Challenge.** \(A_1\) submits two messages \(\mu^*_0, \mu^*_1\) together with the target sender’s keys \((pk_{s^*}, sk_{s^*})\). The challenger \(C\) then chooses uniformly at random a bit \(b \in \{0, 1\}\) and returns the challenge ciphertext \(ct^* \leftarrow SC(pk_{r^*}, sk_{s^*}, \mu^*_b)\) to \(A_1\).

**Phase 2.** \(A_1\) queries the oracles again as in **Phase 1** with a restriction that \(A_1\) is not allowed to make the query PKQ\((r^*)\) and all unsigncryption queries USQ\((r, s, ct)\).

**Output.** \(A_1\) 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 \(A_2\) playing the OWCCA1\(_{SCET}^A_{A_2}\) game is negligible: \(Adv_{A_2}^{OW-iCCA1}(\lambda) := Pr[OWCCA1_{SCET}^A_{A_2} \Rightarrow 1] \leq \text{negl}(\lambda)\).

The OWCCA1\(_{SCET}^A_{A_2}\) game is defined as follows:

**Setup.** The challenger \(C\) first runs \(\text{Setup}(1^\lambda)\) to have the set of public parameters \(pp\) and then runs \(\text{KGr}(pp)\) to get \((pk_r, sk_r)\) for \(r \in [N]\), and \(\text{KGs}(pp)\) 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 \(A_2\). Let \(r^* \in [N]\) be the index of the target receiver.

**Phase 1.** \(A_2\) adaptively makes polynomially bounded number of the following queries:

- Private key query PKQ\((r)\): If \(r = r^*\), \(C\) rejects the query. Otherwise, \(C\) returns the private key \(sk_r\) of the receiver \(R_r\).
- Signcryption query SCQ\((r, s, \mu)\): \(C\) returns the output \(ct\) of \(SC(pk_r, sk_s, \mu)\).
- Unsigncryption query USQ\((r, s, ct)\): \(C\) sends the output of \(USC(sk_r, pk_s, ct)\) back to \(A_2\).
• Tag query TGQ(r): \( C \) returns the output \( t_g(r) \) of \( \text{Tag}(sk_r) \) (even when \( r = r^* \)).

**Challenge.** \( A_2 \) submits the target sender’s keys \( (pk_{s^*}, sk_{s^*}) \), \( C \) chooses a random message \( \mu^* \in \mathcal{M} \) and returns the challenge ciphertext \( ct^* \leftarrow \text{SC}(pk_{r^*}, sk_{s^*}, \mu^*) \) to \( A_2 \).

**Phase 2.** \( A_2 \) queries the oracles again as in **Phase 1** with a restriction that \( A \) is not allowed to make the query \( \text{PKQ}(r^*) \) and all unsigncryption queries \( \text{USQ}(r, s, ct) \).

**Output.** \( A_2 \) outputs \( \mu'' \). 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 \( A_3 \) playing the UFCA\(^{A_3}\) \( \text{SCET} \) game is negligible: 
\[
\text{Adv}_{A_3}^{\text{UF-iCMA}}(\lambda) : = \Pr[\text{UFCA}_{A_3}^{\text{SCET}} \Rightarrow 1] \leq \text{negl}(\lambda).
\]

The UFCA\(^{A_3}\) \( \text{SCET} \) game is defined as follows:

**Setup.** The challenger \( C \) first runs \( \text{Setup}(1^\lambda) \) to have the set of public parameters \( pp \) and then runs \( \text{KGr}(pp) \) to get \( (pk_r, sk_r) \) for \( r \in [N] \), and \( \text{KGs}(pp) \) 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 \( A_3 \). Let \( s^* \in [M] \) be the index of the target sender.

**Queries.** \( A_3 \) adaptively makes polynomially bounded number of the following queries:

- Private key query PKQ(s): If \( s = s^* \), \( C \) rejects the query. Otherwise, \( C \) returns the private key \( sk_s \) of the sender \( S_s \) to \( A_3 \).
- Signcryption query SCQ(r, s, \mu): \( C \) returns the output \( ct \) of \( \text{SC}(pk_r, sk_s, \mu) \) back to \( A_3 \).
- Unsigncryption query USQ(r, s, ct): \( C \) returns the output of \( \text{USC}(sk_r, pk_s, ct) \).
- Tag query TGQ(r): \( C \) returns the output \( t_g(r) \) of \( \text{Tag}(sk_r) \).

**Forge.** \( A_3 \) outputs an index \( r^* \) of some receiver and a ciphertext \( ct^* \) on a message \( \mu^* \), where \( ct^* \) must not be the output of any query \( \text{SCQ}(r, s, \mu) \) in the query phase. He wins the game if \( \text{USC}(sk_{r^*}, pk_{s^*}, ct^*) \neq \bot \). Note that, if \( \mu^* \) is not the same as the messages queried previously, the SCET scheme is called EUF-iCMA (i.e., existential unforgeability). If \( \mu^* \) is one of the messages queried previously but \( (\mu^*, ct^*) \neq (\mu, ct) \) for all \( (\mu, ct) \) 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 $W$ defined as $f_W(x) := Wx \mod q$ (cf. [21]).

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

1. Set parameters $n,q,m,\ell,N,M,\alpha,\sigma_1,\sigma_2,k = \lceil \log q \rceil$, $m = m + nk$ as in Section 6.
2. Samples randomly and independently matrices $C_0,\ldots,C_n, C'_0,\ldots,$ $C'_n \in \mathbb{Z}_q^{n \times nk}$, $B, B' \in \mathbb{Z}_q^{n \times m}$, $U, U' \in \mathbb{Z}_q^{n \times \ell}$.
3. Samples randomly vector $u \in \mathbb{Z}_q^n$.
4. One-way hash function $H : \{0,1\}^\ell \rightarrow \{0,1\}^\ell$, collision-resistant hash functions $H_1 : \mathbb{Z}_q^{n \times m} \rightarrow \mathbb{Z}_q^m$, $H_2$ is a full-rank differences (FRD) encoding $^1$ and a universal hash function $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^m$.
5. A plaintext (message) space $M = \{0,1\}^\ell$.
6. Return $pp = \{n,q,k,m,\ell,N,M,\mathcal{M},(C_i,C'_i)_{i=0}^n,H,H_1,H_2,H_3\}$ 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 $\overline{A}_r,\overline{A}'_r \leftarrow \mathbb{Z}_q^{n \times m}$, $T_r, T'_r \leftarrow D_{2^m \times nk,\sigma_1}$ and then set $A_r = [\overline{A}_r] - [\overline{A}'_r \cdot T_r] \in \mathbb{Z}_q^{n \times m}$, $A'_r = [\overline{A}_r'] - [\overline{A}'_r \cdot T'_r] \in \mathbb{Z}_q^{n \times m}$.
2. Similarly, for each sender $s \in [M]$, generate $\overline{A}_s,\overline{A}'_s \leftarrow \mathbb{Z}_q^{n \times m}$, $T_s, T'_s \leftarrow D_{2^m \times nk,\sigma_1}$ and then set $A_s = [\overline{A}_s]G - [\overline{A}_s \cdot T_s] \in \mathbb{Z}_q^{n \times m}$, $A'_s = [\overline{A}'_s]G - [\overline{A}'_s \cdot T'_s] \in \mathbb{Z}_q^{n \times m}$.
3. Return $pk_r = (A_r,A'_r)$, and $sk_r = (T_r,T'_r)$ as public key and private key for a receiver $R$ of index $r$, $pk_s = (A_s,A'_s)$, and $sk_s = (T_s,T'_s)$ as public key and private key for a sender $S$ of index $s$.

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

1. $r_e, r'_e \leftarrow D_{2^m \times nk}$, $t = f_{\overline{A}_r}(H_1(A_s)) + f_B(r_e) \in \mathbb{Z}_q^n$, $t' = f_{\overline{A}'_r}(H_1(A'_s)) + f_{B'}(r'_e) \in \mathbb{Z}_q^n$.

$^1$See [19] Section 5] for details on FRD.
2. $A_{r,t} = A_r + [0 | H_2(t) | G] \in \mathbb{Z}^{n \times m}_q = [\overline{A}_r | H_2(t) G - \overline{A}_r \cdot T_r] \in \mathbb{Z}^{n \times m}_q$, $A'_{r,t} = A'_r + [0 | H_2(t') | G] \in \mathbb{Z}^{n \times m}_q = [\overline{A}_r' | H_2(t') G - \overline{A}_r' \cdot T'_r] \in \mathbb{Z}^{n \times m}_q$.

3. $(s,s') \leftarrow \mathbb{Z}^{n}_q$, $x_1, x'_1 \leftarrow D_{2^{\ell \cdot \sigma} \cdot q}$.

4. $c_0 = s'A_{r,t} + x'_0 \in \mathbb{Z}^n_q$, $\overline{c}_1 = s'U + x'_1 \in \mathbb{Z}^\ell_q$, $c'_0 = (s')^tA'_{r,t} + (x'_0)^t \in \mathbb{Z}^n_q$, $\overline{c}_1' = (s')^tU' + (x'_1)^t \in \mathbb{Z}^\ell_q$.

5. Set $\overline{c}_t = (c_0, \overline{c}_1, r_s, c'_0, \overline{c}_1', r'_s)$.

6. Sign on $\mu | pk_r | \overline{c}_t$ to get the signature $(e, r_s)$ as follows:
   
   (a) $r_s \leftarrow D_{\mathbb{Z}^{m \cdot \sigma} \cdot q}$.
   
   (b) $h = (h_1, \ldots, h_n) = f_{\overline{A}_r} (H_3(\mu | pk_r | \overline{c}_t)) + f_B (r_s) \in \mathbb{Z}^n_q$.

   (c) $A_{s,h} = [A_s | C_0 + \sum_{i=1}^n h_i \cdot C_i] \in \mathbb{Z}^{n \times (m+n\cdot k)}_q$.

   (d) $e \in \mathbb{Z}^{n+m\cdot k} \leftarrow \text{SampleD}(T_s, A_{s,h}, u, \sigma_2)$.

7. $c_1 = \overline{c}_1 + \mu \cdot |q/2| \in \mathbb{Z}^\ell_q$, $c'_1 = \overline{c}_1' + H(\mu) \cdot |q/2| \in \mathbb{Z}^\ell_q$.

8. Output the ciphertext $ct = (c_0, c_1, r_s, r_s, c'_0, c'_1, r'_s, e)$.

**USC**$(sk_r, pk_s, ct)$: On input a sender’s public key $pk_s := (A_s, A'_s)$, a receiver’s private key $sk_r := (T_r, T'_r)$, a ciphertext $ct = (c_0, c_1, r_s, r_s, c'_0, c'_1, r'_s, e)$, do the following:

1. Compute $t = f_{\overline{A}_r} (H_1(A_s)) + f_B (r_s) \in \mathbb{Z}^n_q$ and $A_{r,t} = [\overline{A}_r | H_2(t) G - \overline{A}_r \cdot T_r] \in \mathbb{Z}^{n \times m}_q$.

2. $(s, x_0) \leftarrow \text{Invert}(T_r, A_{r,t}, c_0)$.

3. Compute $E \in \mathbb{Z}^{m \times \ell} \leftarrow \text{SampleD}(T_r, A_{r,t}, U, \sigma_2)$.

4. Compute $v' = c'_1 - (c_0 - x_0)^tE = x'_1 + \mu \cdot |q/2|$.

5. Recover $\mu$ from $v$ mod $q$.

6. $\overline{c}_t = c_1 - \mu \cdot |q/2| \mod q$, $\overline{c}_1 = c'_1 - H(\mu) \cdot |q/2| \mod q$, and let $\overline{c}_t := (c_0, c_1, r_s, c'_0, c'_1, r'_s)$.

7. Compute $h = (h_1, \ldots, h_n) = f_{\overline{A}_r} (H_3(\mu | pk_r | \overline{c}_t)) + f_B (r_s) \in \mathbb{Z}^n_q$.

8. $A_{s,h} = [A_s | C_0 + \sum_{i=1}^n h_i \cdot C_i] \in \mathbb{Z}^{n \times (m+n\cdot k)}_q$.

9. If $A_{s,h} \cdot e = u \mod q$ and $||e|| \leq \sigma_2 \sqrt{m+n\cdot k}$ then output $\mu$; otherwise, output ⊥.

**Tag**$(sk_r)$: On input a receiver’s private key $sk_r := (T_r, T'_r)$, return the tag $t_{gr} := T'_r$.

**Test**$(tg_{r,i}, ct_i), (tg_{r,j}, ct_j)$: On input a tag $tg_{r,i} := T'_{r,i}$, a ciphertext $ct_i = (c_{0,i}, c_{1,i}, r_{s,i}, r_{s,i}, c'_{0,i}, c'_{1,i}, r'_{s,i}, e_i)$ with respect to the receiver $R_i$, and a tag $tg_{r,j} := T'_{r,j}$, a ciphertext $ct_j = (c_{0,j}, c_{1,j}, r_{s,j}, r_{s,j}, c'_{0,j}, c'_{1,j}, r'_{s,j}, e_j)$ with respect to the receiver $R_j$, do the following:
• For $\mathcal{R}_i$, do:
  1. Compute $t_i' = f_{\mathcal{K}_{r,i}}(H_1(A'_{s,i})) + f_B(r'_{e,i}) \in \mathbb{Z}_q^n$ and $A'_{r,t_i} = \left[\mathcal{K}_{r,i}H_2(t_i')G - \mathcal{K}_{r,i}T'_{r,i}\right] \in \mathbb{Z}_q^{n \times m}$,
  2. $(s_i', x_{0,i}', s_i) \leftarrow \text{Invert}(T'_{r,i}, A'_{r,t_i}, c'_{0,i})$.
  3. Compute $E' \in \mathbb{Z}^{m \times t} \leftarrow \text{SampleD}(T'_{r}, A'_{r,t}, U', \sigma_2)$.
  4. Compute $(v'_i)^t = (c'_{1,i})^t - (c'_{0,i} - x_{0,i})E_i = (x_{1,i})^t + H(\mu_i) \cdot [q/2]$.
  5. Recover $H(\mu_i)$ from $v'_i \mod q$.

• 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 $pp \leftarrow \text{Setup}(1^\lambda), (pk_r, sk_r) \leftarrow \text{KGr}(pp), (pk_s, sk_s) \leftarrow \text{KGs}(pp), (pk_{r_1}, sk_{r_1}) \leftarrow \text{KGr}(pp), (pk_{s_1}, sk_{s_1}) \leftarrow \text{KGs}(pp), (pk_{r_2}, sk_{r_2}) \leftarrow \text{KGr}(pp), (pk_{s_2}, sk_{s_2}) \leftarrow \text{KGs}(pp), t_{g_1} \leftarrow \text{Tag}(sk_{r_1})$ and $t_{g_2} \leftarrow \text{Tag}(sk_{r_2})$, 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 $\Pr[\mu = \text{USC}(sk_r, pk_s, \text{SC}(pk_r, sk_s, \mu))] = 1 - \text{negl}(\lambda)$. Indeed, let $ct = (c_0, c_1, r_s, r_s', c_0', c_1', r_e')$ be a ciphertext outputted by $\text{SC}(pk_r, sk_s, \mu)$. Now what we need to verify is Step 5 in the USC algorithm. To successfully recover $\mu = (\mu_1, \cdots, \mu_\ell)$ from $v = x_1 + \mu \cdot [q/2]$, we compare each component of $v = (v_1, \cdots, v_\ell)$ to $q/2$. If $|v_i| < q/2$ then $\mu_i = 0$. Otherwise, $\mu_i = 1$. This is thanks to the smallness of $x_1 \leftarrow D_{\mathbb{Z}_q^{\ell}}$.

• Second, we need to show that if $\text{USC}(sk_{r_1}, pk_{s_1}, ct_1) = \text{USC}(sk_{r_2}, pk_{s_2}, ct_2) = \mu \neq \bot$, then

$$\Pr[\text{Test}(t_{g_1}, ct_1, t_{g_2}, ct_2) = 1] = 1 - \text{negl}(\lambda).$$

This can be done in the same way as above.

• Finally, we show that if $\text{USC}(sk_{r_1}, pk_{s_1}, ct_1) = \mu_1 \neq \text{USC}(sk_{r_2}, pk_{s_2}, ct_2) = \mu_2$, then

$$\Pr[\text{Test}(t_{g_1}, ct_1, t_{g_2}, ct_2) = 1] = \text{negl}(\lambda).$$

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 algorithm being used such as $\text{Invert}, \text{SampleD}$ 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 d\text{LWE}_{n,2(\pi+t),q,\sigma_{q}} problem and the collision-resistance of the functions $f_{\mathbf{X}}(\cdot) + f_{\mathbf{B}}(\cdot)$ for any $\overline{\mathbf{A}} \leftarrow \mathbb{Z}_{q}^{n \times m}$ and any $\mathbf{B} \leftarrow \mathbb{Z}_{q}^{n \times m}$.

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 demonstrate that Game INDi 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 $t^* \leftarrow f_{\mathbf{X}}(H_1(\mathbf{A}_{s^*})) + f_{\mathbf{B}}(r^*_c)$, and $t'^* \leftarrow f_{\mathbf{X}}(H_1(\mathbf{A}_{s^*})) + f_{\mathbf{B}}(r'^*_c)$.

Game IND1. This game is same as Game IND0, except that if $\mathcal{A}_1$ makes an unsigncryption query $(r^*, s, ct)$ such that $f_{\mathbf{X}}(H_1(\mathbf{A}_{s^*})) + f_{\mathbf{B}}(r^*_e) = t^*$ or $f_{\mathbf{X}}(H_1(\mathbf{A}_{s^*})) + f_{\mathbf{B}}(r'^*_e) = t'^*$, where $t^*$ and $t'^*$ are defined as in Game IND0 (we name this event by Event1), then the challenger outputs $\perp$.

Game IND1 and Game IND0 are indistinguishable since the probability that the event Event1 happens is negligible due to the collision resistance of $f_{\mathbf{X}}(\cdot) + f_{\mathbf{B}}(\cdot)$, and $f_{\mathbf{X}}(\cdot) + f_{\mathbf{B}}'(\cdot)$.

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 GenTrap$(n, m, q, \sigma_{1})$ to generate $(\mathbf{B}, \mathbf{T}_\mathcal{B}), (\mathbf{B}', \mathbf{T}'_\mathcal{B}) \in \mathbb{Z}_{q}^{n \times m} \times \mathbb{Z}_{q}^{m \times nk}$.

Game IND2 and Game IND1 are indistinguishable due to the property of GenTrap algorithm. Namely, although being generated 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 $t^*, t'^* \in \mathbb{Z}_{q}^{n}$ uniformly at random. The challenger uses $t^*, t'^*$ to build $\mathbf{A}_{r^*}, \mathbf{A}'_{r^*}$.
2. Choose $\mathbf{T}_{r^*}, \mathbf{T}'_{r^*} \leftarrow D_{\mathbb{Z}_{\mathbb{P}^{n \times m}, \sigma_{2}}}$ and then set $\mathbf{A}_{r^*} = [\overline{\mathbf{A}}_{r^*} - H_2(t^*)\mathbf{G} - \overline{\mathbf{A}}_{r^*} \cdot \mathbf{T}_{r^*}] \in \mathbb{Z}_{q}^{n \times m}$, $\mathbf{A}'_{r^*} = [\overline{\mathbf{A}}_{r^*} - H_2(t'^*)\mathbf{G} - \overline{\mathbf{A}}_{r^*} \cdot \mathbf{T}'_{r^*}] \in \mathbb{Z}_{q}^{n \times m}$.
3. The public key for $r^*$ is $pk_{r^*} = (\mathbf{A}_{r^*}, \mathbf{A}'_{r^*})$ and the private key key for $r^*$ is $sk_{r^*} = (\mathbf{T}_{r^*}, \mathbf{T}'_{r^*})$. 

16
Game IND3 and Game IND2 are indistinguishable since the distribution of $A_r$, $A'_r$ is the same as that of $A_r$, $A'_r$ 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 $b \leftarrow \{0, 1\}$.
2. $A_{r^*}, t^* = A_r + [0]|H_2(t^*)G| = [\overline{A}_r] - [\overline{A}_r] \cdot T_{r^*} \in \mathbb{Z}_q^{m \times n}$,
   $A'_r, t^* = A'_r + [0]|H_2(t^*)G| = [\overline{A}_r] - [\overline{A}_r] \cdot T'_{r^*} \in \mathbb{Z}_q^{m \times n}$.
3. Compute $r_e^* \leftarrow \text{SampleD}(T_B, B, (t^* - f_{\overline{A}_r}(H_1(A_r^*))), \alpha q), r_e^* \leftarrow \text{SampleD}(T_B, B', (t^{*'} - f_{\overline{A}'_r}(H_1(A'_r^*))), \alpha q)$.
4. Sample $s, s^* \leftarrow \mathcal{D}_{\mathbb{Z}_q^m}, x_0, x_0' \leftarrow D_{\mathbb{Z}^{m \times \alpha q}^\|}, x_1, x_1' \leftarrow D_{\mathbb{Z}^{m \times \alpha q}^\|}$.
5. Compute $c_0 = s^*\overline{A}_r + x_0' \in \mathbb{Z}_q^{m}, c_1 = s^*U + x_1' \in \mathbb{Z}_q^{m}, c_0' = (s')^*\overline{A}_r + (x_0')^* \in \mathbb{Z}_q^{m}, c_1' = (s')^*U + (x_1')^* \in \mathbb{Z}_q^{m}$.
6. Set $(c_0')^t := (\tilde{c}_0, \tilde{c}_0') \cdot (\mathcal{C}_{r^*}), (c_0')^t := ((\tilde{c}_0, \tilde{c}_0)' \cdot \mathcal{T}_{r^*})$.
7. Sign on $|\mu^*_0|pq^{r^*} |\tilde{c}^*|$ with $\tilde{c}^* = (c_0, c_1, e^*, c_0^*, c_1^*, r_e^*, e^*)$ to get the signature $(c_0^*, c_1^*, r_e^*, c_0^*, c_1^*, r_e^*, e^*)$ to $A_1$.
8. $c_1^* = c_1^* + |\mu^*_0|p \cdot |q/2|, c_1^* = c_1^* + H(\mu^*_0) \cdot |q/2|
9. Return $ct^* = (c_0^*, c_1^*, r_e^*, c_0^*, c_1^*, r_e^*, e^*)$ to $A_1$.

Game IND4 and Game IND3 are indistinguishable as the challenger is just following the real signcryption algorithm $SC$ with $A_r = [\overline{A}_r] - H_2(t^*)G - [\overline{A}_r] \cdot T_{r^*} \in \mathbb{Z}_q^{m \times n}, A'_r = [\overline{A}_r] - H_2(t^*)G - [\overline{A}'_r] \cdot T'_{r^*} \in \mathbb{Z}_q^{m \times n}$ and the distribution of $r_e^*, c_0^*$ is still $D_{\mathbb{Z}^{m \times \alpha q}^\|}$ by the property of SampleD.

Game IND5. This game is the same as Game 4, except that $c_0^*, c_1^*, r_e^*, c_0^*, c_1^*, r_e^*$ 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 $A_1$ can distinguish Game IND5 and Game IND4. Then we will construct an algorithm $B_1$ that can solve an LWE instance.

LWE Instance. $B_1$ is given a pair $(F, c') \in \mathbb{Z}_q^{n \times (2m^2 + 2\ell)} \times \mathbb{Z}_q^{2(N + \ell)}$ that can be parsed as $(A_1|A'_1|U'|U', \tilde{c}_0') \cdot (\tilde{c}_0')^t \cdot (\tilde{c}_1')^t \in \mathbb{Z}_q^{n \times (2m^2 + 2\ell)} \times \mathbb{Z}_q^{2(2m^2 + 2\ell)} \times \mathbb{Z}_q^{2(m^2 + m^2 + \ell + 2\ell)}$, and $B_1$ has to decide whether
Phase 1.

Setup. \(B_1\) simulates public parameters \(pp\), public keys for \(M\) senders and \(N\) receivers as follows:

- Pick \(n, q, k, \bar{m}, m, \ell, n, N, M, \alpha, \sigma_1, \sigma_2\) and use hash functions \(H, H_1, H_2, H_3\). The message space is \(\mathcal{M}\).
- Randomly guess \(r^* \in \{1, \ldots, N\}\) to be the target receiver targeted by \(A_1\), and then set \(\overline{A}_{r^*} := A\), \(\overline{A}'_{r^*} := A'\).
- Choose \(t^*, t'^* \in \mathbb{Z}_q^n\) uniformly at random and choose \(T_{r^*}, T'_{r^*} \leftarrow Z_{2\overline{m} \times nk, \sigma_1}\) and then set \(A_{r^*} := [\overline{A}_{r^*} - H_2(t^*)G - \overline{A}'_{r^*} \cdot T_{r^*}] \in \mathbb{Z}_{q \times m}^{n, m}, A'_{r^*} := [\overline{A}'_{r^*} - H_2(t'^*)G - \overline{A}_{r^*} \cdot T'_{r^*}] \in \mathbb{Z}_{q \times m}^{n, m}.
- Also, use \(\text{GenTrap}(n, \bar{m}, q, \sigma_1)\) to generate \((B, T_B), (B', T'_B) \in \mathbb{Z}_{q \times n}^{n, m} \times \mathbb{Z}_{\bar{m} \times nk}^{n, m}.
- Sample \(u \in \mathbb{Z}_q^n\) and matrices \(C_0, \ldots, C_n, C'_0, \ldots, C'_n \in \mathbb{Z}_{q \times m}^{n, m}.
- For each receiver \(r \in [N] \setminus \{r^*\}\) and each sender \(s \in [M]\), use the algorithm KG to generate \((A_r, T_r), (A'_r, T'_r), (A_s, T_s), (A'_s, T'_s) \in \mathbb{Z}_{q \times n}^{n, m} \times \mathbb{Z}_{\bar{m} \times nk}^{n, m}.
- Set \(pp = \{n, q, k, \bar{m}, m, \ell, n, \alpha, \sigma_1, \sigma_2, N, M, \mathcal{M}, (C_i, C'_i)_{i=0}^n, H, H_1, H_2, H_3\}\) as public parameters and \(pk_r = (A_r, A'_r), pk_s = (A_s, A'_s)\) 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 \(A_1\).

Phase 1. \(A_1\) adaptively makes a polynomially bounded number of the following queries:

- Private key query \(PKQ(r)\): If \(r = r^*\), \(B_1\) rejects the query. Otherwise, \(B_1\) returns the private key \(sk_r = (T_r, T'_r)\) to \(A_1\).
- Signcryption query \(SCQ(r, s, \mu)\): \(B_1\) sends the output \(ct\) of \(SC(pk_r, sk_s, \mu)\) back to \(A_1\).
- Unsigncryption query \(USQ(r, s, ct)\): If \(A_1\) makes an unsigncryption query \((r, s, ct)\) such that \(t^* = f_{\overline{A}_r}(H_1(A_s)) + f_{B}(r_e)\) and \(t'^* = f_{\overline{A}'_r}(H_1(A'_s)) + f_{B'}(r'_e)\) then \(B_1\) outputs \(\bot\). Otherwise, \(B_1\) sends the output \(\mu/\bot\) of \(USC(sk_r, pk_s, ct)\) back to \(A_1\).
- Tag query \(TGQ(r)\): If \(r = r^*\), \(B_1\) rejects the query. Otherwise, \(B_1\) sends the output \(tg_r = T'_r\) back to \(A_1\).
Challenge. $A_1$ submits two messages $\mu_0^*, \mu_1^*$ together with the target sender’s keys $(pk_s, sk_r)$. The adversary $B_1$ does the following:

1. Choose randomly $b \xleftarrow{\$} \{0, 1\}$.
2. Compute $r_e^* \leftarrow \text{SampleD}(T_B, B, (t^* - f_{\overline{A}_r^*}(H_1(A_{s^*}))), \alpha q)$, $r_e^{**} \leftarrow \text{SampleD}(T_B', B', (t^{**} - f_{\overline{A}_r^*}(H_1(A_{s^*}))), \alpha q)$.
3. $A_{r^*, t^*} = A_{r^*} + [0|H_2(t^*)]G = [\overline{A}_{r^*} - \overline{A}_{r^*} \cdot T_{r^*}] \in \mathbb{Z}_q^{n \times m}$, $A_{r^{**}, t^{**}} = A_{r^{**}} + [0|H_2(t^{**})]G = [\overline{A}_{r^{**}} - \overline{A}_{r^{**}} \cdot T_{r^{**}}] \in \mathbb{Z}_q^{n \times m}$.
4. Set $(c_0^*)^t := (\hat{c}_0^* | c_0^* T_{r^*})$, $(c_0^{**})^t := ((\hat{c}_0^{**})^t | (\hat{c}_0^{**})^t T_{r^{**}})$.
5. Sign on $\mu_0^* | pk_{r^*} / T^*$ with $T^* = (c_0^*, c_1^*, r_e^*, c_0^{**}, c_1^{**}, r_e^{**}, e^*)$ to get the signature $(e^*, r_e^*)$ as usual.
6. $c_1^* = \overline{c}_1^* + \mu_0^* \cdot [q/2]$, $c_1^{**} = \overline{c}_1^{**} + H(\mu_0^*) \cdot [q/2]$
7. Return $ct^* = (c_0^*, c_1^*, r_e^*, r_e^{**}, c_0^{**}, c_1^{**}, r_e^{**}, e^*)$ to $A_1$.

Phase 2. $A_1$ queries the oracles again as in Phase 1 with a restriction that $A_1$ is not allowed to make the queries $PKQ(r^*)$ and USQ$(r^*, s^*, ct^*)$. Output. $B_1$ outputs whatever $A_1$ outputs.

Analysis. The probability that an unsigncryption query $(r, s, ct)$ makes $t = t^*$ and $t' = t^{**}$ is negligible as $t^*$ and $t^{**}$ are chosen randomly in Setup phase. Then, we have $H_2(t - t^*)$ and $H_2(t' - t^{**})$ are invertible then we can apply Invert as in the real unsigncryption algorithm USC. Obviously, if $(F, c^t)$ is the LWE instance then the view of $A_1$ as in Game IND4; while if $(F, c^t)$ is uniform in $\mathbb{Z}_q^{n \times (2m + 2\ell)} \times \mathbb{Z}_q^{2(m + \ell)}$ then the view of $A_1$ as in Game IND5. Therefore, if $A_1$ can distinguish Game IND4 and Game IND5 then $B_1$ 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 dlWEM$_{n, 2(\pi + \ell), q, \alpha q}$ problem is hard and the functions $f_{\overline{A}_r^*}(\cdot) + f_B(\cdot)$ are collision-resistant for any $\overline{A}_r \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and any $B \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$. In particular, the advantage of the OW-iCCA1 adversary 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 $f_{\overline{A}_r^*}(\cdot) + f_B(\cdot)$ and $\epsilon_{LWE}$ is the advantage of solving the dlWEM$_{n, 2(\pi + \ell), q, \alpha q}$ 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 \(A_2\) wins Game \(\text{OW}_i\), 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 \(\text{OW}_0\). 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 \(A_2\) is \(s^*\). Also, assume that \(t^* := f_{\overline{X}_{s^*}}(H_1(A,s^*)) + f_B(r_e)\), and \(t'^* := f_{\overline{X}_{s^*}}(H_1(A',s^*)) + f_B'(r'^*_e)\). Note that, the adversary \(A_2\) can get the trapdoor \(T'_{r^*}\) using the trapdoor query for the target receiver \(r^*\).

Game \(\text{OW}_1\). This game is same as Game \(\text{OW}_0\), except that in the Challenge phase, on the challenge plaintext \(\mu^* \leftarrow - M\), the challenger first chooses \(\mu'^* \leftarrow - M\) then signcrypts \(\mu^*\) in \(c_1^* = c'_1 + H(\mu^*) \cdot \lfloor q/2 \rfloor\), \(c_1^* = c'_1 + H(\mu'^*) \cdot \lfloor q/2 \rfloor\).

Since the view of the adversary \(A_2\) is the same in both Game \(\text{OW}_1\) and Game \(\text{OW}_0\), except the case \(A_2\) can break the one-wayness of \(H\), we have
\[
|\Pr[W_1] - \Pr[W_0]| \leq \epsilon_{H,\text{OW}}.
\]

Game \(\text{OW}_2\). This game is same as Game \(\text{OW}_1\), except that if \(A_2\) makes an unsigcryption query \((r^*, s, ct)\) such that \(f_{\overline{X}_{s^*}}(H_1(A,s^*)) + f_B(r_e) = t^*\) or \(f_{\overline{X}_{s^*}}(H_1(A',s^*)) + f_B'(r'_e) = t'^*\), where \(t^*\) and \(t'^*\) are defined as in Game \(\text{OW}_0\) (we name this event by Event1), then the challenger outputs \(\perp\).

Since the view of the adversary \(A_2\) is the same, except once the event Event1 happens, in both Game \(\text{OW}_2\) and Game \(\text{OW}_1\), we have
\[
|\Pr[W_2] - \Pr[W_1]| \leq \epsilon_{f,\text{CR}}.
\]

Game \(\text{OW}_3\). This game is same as Game \(\text{OW}_2\), except that instead of \(B, B'\) being uniform in \( \mathbb{Z}_q^{n \times m}\), the challenger uses \(\text{GenTrap}(n, m, q, \sigma_1)\) to generate \((B, T_B), (B', T'_B) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{m \times nk}\).

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

Game \(\text{OW}_4\). This game is same as Game \(\text{OW}_3\), except that in the Setup phase, for the target receiver \(r^*\), the challenger generates as follows:
1. Choose \( t^*, t'^* \in \mathbb{Z}_q^n \) uniformly at random. The challenger uses \( t^*, t'^* \) to build \( A_{r*}, A'_{r*} \).

2. Choose \( T_{r*}, T'_{r*} \leftarrow D_{\{0,1\}^{\mathbb{Z}_q}} \) and then set \( A_{r*} = [\mathbb{X}_{r*}] - H_2(t^*) \mathbb{G} - \mathbb{A}_{r*} \cdot T_{r*} \), \( A'_{r*} = [\mathbb{X}_{r*}] - H_2(t'^*) \mathbb{G} - \mathbb{A}_{r*} \cdot T'_{r*} \) \in \mathbb{Z}_q^{n \times m}.

3. The public key for \( r^* \) is \( pk_{r*} = (A_{r*}, A'_{r*}) \) and the private key for \( r^* \) is \( sk_{r*} = (T_{r*}, T'_{r*}) \).

In this game, once the adversary \( A_2 \) makes a trapdoor query for \( r^* \), the challenger still easily returns \( T'_{r*} \) to \( A_2 \). In **Game OW4** the view of \( A_2 \) is unchanged in comparison with in **Game OW3** since the distribution of \( A_{r*}, A'_{r*} \) is the same as that of \( A_r, A'_r \) 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 \( \mu^* \), the challenger does the following:

1. Compute \( r^*_c \leftarrow \text{SampleD}(T_B, B, (t^* - f_{\mathbb{X}_{r*}}(H_1(A_{r*}))), \alpha q), r'_c \leftarrow \text{SampleD}(T_B, B', (t'^* - f_{\mathbb{X}_{r*}}(H_1(A'_{r*}))), \alpha q) \).

2. Sample \( s, s' \leftarrow D_{\mathbb{Z}_q^n}, \hat{x}_0, \hat{x}_0' \leftarrow D_{\mathbb{Z}_q}, x_1, x_1' \leftarrow D_{\mathbb{Z}_q^\ell} \).

3. Compute \( \hat{c}_0 = s' \mathbb{X}_{r*} + \hat{x}_0 \in \mathbb{Z}_q^n, \epsilon_1 = s' \mathbb{U} + x_1 \in \mathbb{Z}_q^\ell, \hat{c}_0' = (s')' \mathbb{X}_{r*} + (\hat{x}_0')' \in \mathbb{Z}_q^n, \epsilon_1' = (s')' \mathbb{U} + (x_1')' \in \mathbb{Z}_q^\ell \).

4. Set \( (c_0')^t := (c_0^t|c_0'^t)T_{r*}, (c_0''^t) := (c_0'')^t \).

5. Sign on \( \mu_q^t | pk_{r^*} | \epsilon^t \) with \( \sigma^t = (c_0^t, \epsilon_1, r_c^t, c_0'^t, c_0''^t, r_c''^t, e^t) \) to get the signature \( (e^t, r_c^t) \) as usual.

6. \( \mu' \leftarrow \mathcal{M}, c_1^* = c_1^t + \mu^* \cdot [q/2], c_1'^* = c_1'^t + H(\mu') \cdot [q/2] \).

7. Return \( ct^* = (c_0^t, c_1^t, r_c^t, c_0'^t, c_1'^t, r_c''^t, e^t) \) to \( A_2 \).

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 \( ct^* = (c_0^t, c_1^t, r_c^t, c_0'^t, c_1'^t, r_c''^t, e^t) \) is chosen uniformly at random. The advantage of \( A_2 \) 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] \( \square \)
Before stating the SUF-iCMA security, we recap the so-called abort-resistant hash functions, presented in [19 Section 7.4.1]. We will exploit the hash functions in designing answers to the adversary’s queries.

**Definition 9 ([19 Definition 26]).** Let \( \mathcal{H} := \{ H : X \rightarrow Y \} \) be a family of hash functions \( H \) from \( X \) to \( Y \) where \( 0 \in Y \). For a set of \( Q + 1 \) inputs \( \mathbf{h} := (h^*, h^{(1)}, \ldots , h^{(Q)}) \), the non-abort probability of \( \mathbf{h} \) is defined as

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

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

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

**Lemma 12 ([19 Lemma 27]).** Let \( q \) be a prime and \( 0 < Q < q \). Consider the family \( \mathcal{H}_{\text{Wat}} := \{ H_x : \mathbb{Z}_q^n \setminus \{0\} \rightarrow \mathbb{Z}_q : x = (x_1, \ldots , x_n) \in \mathbb{Z}_q^n \setminus \{0\} \} \) defined as \( H_x(h) = 1 + \sum_{i=1}^n x_i h_i \in \mathbb{Z}_q \) where \( h = (h_1, \ldots , h_n) \in \mathbb{Z}_q^n \). Then \( \mathcal{H}_{\text{Wat}} \) is \((Q, \frac{1}{Q}(1-\frac{Q}{q}), \frac{1}{2})\) 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 \( \text{SIS}_{n,2m+2nk+1,q,\beta} \) 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 \((r^{(1)}, s^{(1)}, \mu^{(1)}), \ldots , (r^{(Q)}, s^{(Q)}, \mu^{(Q)})\).

We consider Case 1 first.

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

\[
F \cdot x = 0 \pmod{q},
\]

where \( F \stackrel{\$}{\leftarrow} \mathbb{Z}_q^{n \times (2m+2nk+1)}, \|x\| \leq \beta_1, \quad (1)
\]

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

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 \( s^* \leftarrow \{1, \ldots, M\} \) to be the target sender that \( \mathcal{F} \) would like to forge, and then set \( \overline{A}_s^* := \overline{A}, \overline{A}_s := \overline{A}_s^\prime, A_s^* := [\overline{A}_s^*|W] \in \mathbb{Z}_q^{n \times m}, A_s^\prime := [\overline{A}_s^\prime|W] \in \mathbb{Z}_q^{n \times m} \). Recall that \( m = m + nk \).
- Also, use \( \text{GenTrap}(n, m, q, \sigma_1) \) to generate \((B, T_B), (B^\prime, T_B^\prime) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{m \times \ell} \), and choose randomly \( U, U^\prime \leftarrow \mathbb{Z}_q^{n \times \ell} \).

- For \( i \in \{0, \ldots, n\} \), sample \( T_{s^*, i}, T_{s^*, i}^\prime \leftarrow D_{\mathbb{Z}_{\mathcal{M}\times nk}, \sigma_1} \). Let \( x_0 := 1 \). Choose \( x := (x_1, \ldots, x_n) \leftarrow \mathbb{Z}_q^n \setminus \{0\} \) and for \( i \in \{0, \ldots, n\} \), let \( H_i = x_i I_n \) and set \( C_i := H_i G - \overline{A}_s T_{s^*, i} (\mod q) \in \mathbb{Z}_q^{n \times nk} \), \( C_i^\prime := H_i^\prime G - \overline{A}_s^\prime T_{s^*, i} (\mod q) \in \mathbb{Z}_q^{n \times nk} \). Obviously, by Lemma \( \ref{lemma:randomness} \), such an \( x \) will define an abort-resistant hash function belonging to \( \mathcal{H}_{\text{Wat}} \).

- For each \( j \in [Q] \): repeat choosing \( h^{(j)} = (h^{(j)}_1, \ldots, h^{(j)}_n) \in \mathbb{Z}_q^n \) uniformly at random until being such that \( (1 + \sum_{i=1}^n x_i \cdot h^{(j)}_i) \neq 0 \).
- For all \( s \in [M] \setminus \{s^*\} \) and all \( r \in [N] \), use \( \text{GenTrap}(n, m, q, \sigma_1) \) to generate \((A_r, T_r), (A_r^\prime, T_r^\prime), (A_s, T_s), (A_s^\prime, T_s^\prime) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{m \times \ell} \). Set \( pp = \{n, q, k, m, \ell, n, N, M, \alpha, \sigma_1, \sigma_2, \mathcal{M}, (C_i, C_i^\prime)_{i=0}^n, H, H_1, H_2, H_3\} \) as public parameters and \( pk_r = (A_r, A_r^\prime), pk_s = (A_s, A_s^\prime) \) as public keys corresponding to each receiver \( r \in [N] \), and each sender \( s \in [M] \).

Send \( pp, pk_s, pk_r \) 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 \( sk_s = (T_s, T_s^\prime) \) of a sender \( S_s \) to \( \mathcal{F} \).
- For the \( j \)-th signcryption query SCQ\((s^{(j)}, r^{(j)}, \mu^{(j)})\) querie: if \( s^{(j)} \neq s^* \), \( \mathcal{G} \) sends the output \( ct \) of \( SC(pk_r^{(j)}, sk_s^{(j)}, \mu^{(j)}) \) back to \( \mathcal{F} \). Otherwise, \( \mathcal{G} \) creates a ciphertext \( ct \) on input \((pk_r^{(j)}, sk_s^{(j)}, \mu)\) as follows:

1. \( r_e, r_e^\prime \leftarrow D_{\mathbb{Z}_q^{m \times nk}}, t = f_{\overline{A}_r}(H_1(A_s^*)) + f_B(r_e) \in \mathbb{Z}_q^n, t^\prime = f_{\overline{A}_r^\prime}(H_1(A_s^\prime)) + f_B^\prime(r_e^\prime) \in \mathbb{Z}_q^n \).
2. \( A_r, t = A_r + [0|H_2(t)|G] \in \mathbb{Z}_q^{n \times m} = [\overline{A}_r|H_2(t)G - \overline{A}_r \cdot T_r] \in \mathbb{Z}_q^{n \times m} \), \( A_s^\prime, t^\prime = A_s^\prime + [0|H_2(t^\prime)|G] \in \mathbb{Z}_q^{n \times m} = [\overline{A}_s^\prime|H_2(t^\prime)G - \overline{A}_s^\prime \cdot T_r^\prime] \in \mathbb{Z}_q^{n \times m} \).
3. $\mathbf{s}, \mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x}_0, \mathbf{x}'_0 \xleftarrow{\$} D_{\mathbb{Z}^m, \alpha q}$, $\mathbf{x}_1, \mathbf{x}'_1 \xleftarrow{\$} D_{\mathbb{Z}^t, \alpha q}$.

4. $c_0 = \mathbf{s}^t A_{r, t} + x'_0 \in \mathbb{Z}_q^t$, $\overline{c}_1 = \mathbf{s}^t U + x'_1 \in \mathbb{Z}_q^t$.

$c_0' = (s')^t A_{r, t} + (x'_0)^t \in \mathbb{Z}_q^t$, $\overline{c}_1' = (s')^t U' + (x'_1)^t \in \mathbb{Z}_q^t$.

5. $\overline{c} = (c_0, \overline{c}_1, c_0', \overline{c}_1')$.

6. Generate a signature on $\mu^{(j)}|pk_{r(j)}|\overline{c}|\overline{c}'$:

(a) Compute $r_s \xleftarrow{\$} \text{SampleD}(T_B, \mathbf{B}, (h^{(j)} - f_{\overline{\mathbf{x}}_s}(H_3(\mu^{(j)}|pk_r|\overline{c}))))$, $\alpha q$.

(b) $A_{s^*, h} = (A_{s^*} C_0 + \sum_{i=1}^n h_i^{(j)} \cdot C_i) \in \mathbb{Z}_q^{n \times (m+nk)}$ with $H := H_0 + \sum_{i=1}^n h_i \cdot H_i = (1 + \sum_{i=1}^n x_i \cdot h_i)^{i} \not\equiv 0$ and $T_{s^*} := T_{s^*} + \sum_{i=1}^n h_i \cdot T_{s^*, i}$. Note that, by Lemma [11], $T_{s^*}$ is a $G$-trapdoor for $A_{s^*, h}$.

(c) $e \in \mathbb{Z}^{m+nk} \xleftarrow{\$} \text{SampleD}(T_{s^*}, A_{s^*, h}, \mathbf{u}, \sigma_2)$.

(d) $(e, r_s)$ is the signature.

7. $c_1 = \overline{c}_1 + \mu^{(j)} \cdot \lfloor q \rfloor / 2$, $c'_1 = \overline{c}'_1 + H(\mu^{(j)}) \cdot \lfloor q \rfloor / 2$.

8. Output $ct = (c_0, c_1, r_s, r_e, c_0', c_1', r_e, e)$.

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

- For tag query TGQ($r$): $\mathcal{G}$ simply sends the output $tg_r := T'_r$ 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 $ct^* = (c_0^*, c_1^*, r_s^*, c_0'^*, c_1'^*, r_e^*, e^*)$ on message $\mu^*$, i.e., $ct^*$ must be not the output of any previous query SCQ($r, s, \mu$), and USC$(sk_r, pk_s, ct^*) \not\equiv \bot$.

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

- Compute $h^* = (h_1^*, \ldots, h_n^*) \xleftarrow{\$} \text{SampleD}(H_3(\mu^*|pk_r|\overline{c}'))$, $f_B(r_s) \in \mathbb{Z}_q^n$ and check if $H := H_0 + \sum_{i=1}^n h_i^* \cdot H_i = (1 + \sum_{i=1}^n x_i \cdot h_i^*)I_n = 0$. If not, $\mathcal{G}$ aborts the game. Otherwise, it computes $A_{s^*, h^*} = (A_{s^*} C_0 + \sum_{i=1}^n h_i^* \cdot C_i) \in \mathbb{Z}_q^{n \times (m+nk)}$. Note that, the probability that $\mathcal{G}$ aborts the game is negligible by appropriately choosing parameters via Lemma [12].

- From $A_{s^*, h^*} \cdot e^* = u \text{ mod } q$ and $\|e^*\| \leq \sigma_2 \sqrt{m+nk}$, we have $[A_{s^*} |W] - [\overline{A}_{s^*} |T_{s^*}] \cdot e^* = u \text{ (mod } q)$, equivalently,

$$[A_{s^*} |W] \cdot \left[ \begin{array}{cc} I_m & 0 \\ 0 & -T_{s^*} \\ I_{nk} & 0 \end{array} \right] \cdot e^* = u \text{ (mod } q).$$

- Let $\hat{x} := \left[ \begin{array}{cc} I_m & 0 \\ 0 & -T_{s^*} \\ I_{nk} & 0 \end{array} \right] \cdot e^* \not\equiv 0$, then $[\overline{A} |W|u] \cdot (\overline{\hat{x}_1}) = 0 \text{ (mod } q)$. Hence, $\mathcal{G}$ gets a solution $x$ to the SIS problem [1], i.e., $F \cdot x =$
0 (mod q), with \( x = (\overline{x}_i) \), and by Lemma 2:

\[
\|x\| = \|\overline{x}\| + 1 \leq s_1(T_{s^*}) \cdot \|f^s\| + 1 \\
\leq \sigma_1 \cdot \sigma_2 \cdot \sqrt{n} + 1 \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{m} + \sqrt{nk}) \cdot \sqrt{m + nk} + 1.
\]

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

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

where \( \beta_2 := 2\sigma_1 \cdot \sigma_2 \cdot \sqrt{n} + 1 \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{m} + \sqrt{nk}) \cdot \sqrt{m + nk} \). Then, \( \mathcal{G} \) parses \( F \) as \( F := [\overline{A}W|\overline{A}'W'] \), where \( \overline{A}, \overline{A}' \in \mathbb{Z}_q^{n \times m} \), and \( W, W' \in \mathbb{Z}_q^{n \times nk} \). 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 \( s^* \overset{\$}{\leftarrow} \{1, \ldots, M\} \) and \( r^* \overset{\$}{\leftarrow} \{1, \ldots, N\} \) to be the target sender and the target receiver respectively, that \( \mathcal{F} \) would like to forge, and then set \( \overline{A}_{s^*} := \overline{A}, \overline{A}'_{s^*} := \overline{A}', A_{s^*} := [\overline{A}_{s^*}|W] \in \mathbb{Z}_q^{n \times m}, A'_{s^*} := [\overline{A}'_{s^*}|W'] \in \mathbb{Z}_q^{n \times m} \). Recall that \( m = m + nk \).
- Repeat choosing \( h^{(j)} = (h_1^{(j)}, \ldots, h_n^{(j)}) \in \mathbb{Z}^n_q \) uniformly at random until \( (1 + \sum_{i=1}^n x_i \cdot h_i^{(j)}) = 0 \) 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 \( h^{(j_0)} \) to be the one corresponding to the target query \( (r^*, s^*, \mu^*) \) and let \( h^* = (h_1^*, \ldots, h_n^*) \leftarrow h^{(j_0)} \).
- Set \( A_{s^*}, h^* = [A_{s^*}C_0 + \sum_{i=1}^n h_i^* \cdot C_i] = [\overline{A}_{s^*}W - \overline{A}_{s^*}T_{s^*}] \), where \( T_{s^*} := T_{s^*,0} + \sum_{i=1}^n h_i^* \cdot T_{s^*,i} \).
- Choose \( e^{(1)} = D_{\mathbb{Z}_q^{m+nk}, \sigma_2} \), set \( u := A_{s^*}, h^* \cdot e^{(1)} \in \mathbb{Z}_q^n \) and send \( u \) 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 \( SCQ(r^*, s^*, \mu^*) \), \( \mathcal{G} \) responds as follows:

1. Produce \( \overline{c}^* = (c_0^*, \overline{c}_1^*, r^*_c, c_0^*, \overline{c}_1^*, r^*_c) \) as usual.
2. Generate a signature on \( \mu^*|pk_r^*|\overline{c}^* \):
   (a) \( r^*_s \leftarrow \text{SampleD}(T_{B}, B, h^* - f_{\overline{A}_{s^*}}(H_3(\mu^*|pk_r^*|\overline{c}^*)), \alpha q) \).
   (b) Set \( (e^{(1)}, r^*_s) \) to be the signature.
3. \( c^*_1 = \overline{c}_1^* + \mu \cdot \lfloor q/2 \rfloor, \quad c^*_1 = \overline{c}_1^* + H(\mu) \cdot \lfloor q/2 \rfloor \).
4. Return \(ct^{(1)} = (c_{0*}, c_1^*, r_{e*}, r_{s*}, c_0^*, c_1^*, r_{e*}, e^{*(1)})\) to \(F\).

**Forge.** The forger \(F\) outputs the target receiver’s key-pair \((pk_r, sk_s)\) together with a new valid ciphertext \(ct^{(2)} = (c_{0*}, c_1^*, r_{e*}, r_{s*}, c_0^*, c_1^*, r_{e*}, e^{*(2)})\) on the message \(\mu^*\). Note that, \(e^{*(2)} \neq e^{*(1)}\) while \(c_{0*}, c_1^*, r_{e*}, r_{s*}, c_0^*, c_1^*, r_{e*}\) are unchanged due to the validity of \(ct^{(2)}\) corresponding to \(A_{s^*}h^*\).

**Analysis.** At the moment, \(G\) proceeds the following steps:

- Recall \(A_{s^*}h^* = [A_{s^*}|C_0 + \sum_{i=1}^n h_i^* \cdot C_i] = [\overline{A}_{s^*}|W] - [\overline{A}_{s^*}T_{s^*}]\).
- From \(A_{s^*}h^* \cdot e^{(1)} = u \mod q\) and \(A_{s^*}h^* \cdot e^{(2)} = u \mod q\), we have

\[
\overline{[A_{s^*}|W]} \cdot \left( \begin{bmatrix} I_m & 0 \\ 0 & I_{nk} \end{bmatrix} - T_{s^*} \right) \cdot (e^{(1)} - e^{(2)}) = 0 \pmod q.
\]

- Let \(\tilde{x} := \left[ \begin{bmatrix} I_m & 0 \\ 0 & I_{nk} \end{bmatrix} - T_{s^*} \right] \cdot (e^{(1)} - e^{(2)})\), then \([\overline{A}|W] \cdot \tilde{x} = 0 \pmod q\).

Hence, \(G\) gets a solution \(x\) to the SIS problem \([2]\), i.e., \(F \cdot x = 0 \pmod q\), with \(x = (\tilde{x})\), and

\[
\|x\| = \|\tilde{x}\| \leq s_1(T_{s^*}) \cdot \|e^{(1)} - e^{(2)}\|
\]

\[
\leq 2\sigma_1 \cdot \sigma_2 \cdot \sqrt{n+1} \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{m} + \sqrt{nk}) \cdot \sqrt{m + nk},
\]

by Lemma [2].

It remains to prove that \(\tilde{x} \neq 0\) with overwhelming probability. Let \(w := e^{(1)} - e^{(2)} \neq 0\) and parse \(w = (w_1, w_2, w_3)\). Then \(\tilde{x} = (w_1 - T_{s^*}w_3, w_2)\).

Obviously, if \(w_2 \neq 0\) or \(w_3 := (w_1, \cdots, w_{nk}) = \theta\) then \(\tilde{x} \neq 0\). Otherwise, i.e., \(w_2 = 0\) and \(w_3 \neq 0\), we will show that \(A_{s^*}(w_1 - T_{s^*}w_3) = 0\) happens only with negligible probability. Indeed, without loss of generality, assume that \(w_{nk} \neq 0\) then \(A_{s^*}(w_1 - T_{s^*}w_3) = 0\) only if \(t_{nk} \sim D_{\Lambda_q^1(A_{s^*})} c_{r_1}\) and \(t_{nk} \cdot w_{nk} = y\) for some \(y \in \Lambda_q^1(A_{s^*}) + c\) for any \(c\) in \(\text{span}(\Lambda_q^1(A_{s^*}))\), where \(t_{nk}\) is the \(nk\)-th column of \(T_{s^*}\). Then by Lemma [1] such a \(t_{nk}\) exists with negligible probability.

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

\[\square\]

6. **Parameter Selection**

- We take \(n\) as the security parameter.
- For the gadget-based trapdoor mechanism to work: Choose \(q \geq 2, m \geq 1, k = \lceil \log_2 q \rceil\), and \(m = O(n \log q)\).
• Also choose $q, n, m, \ell, Q$ 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 $f_{\mathbf{A}_r}(\cdot) + f_{\mathbf{B}}(\cdot)$ are collision-resistance for any $\mathbf{A}_r \leftarrow Z_q^{n \times m}$ and any $\mathbf{B} \leftarrow Z_q^{n \times m}$.

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

• For the Gaussian parameter $\sigma_1$ used in $\text{KeyGen}$ (which follows $\text{Gen-Trap}$): $\sigma_1 \geq \omega(\sqrt{\log n})$. Note also that, all private keys for senders and users $T_s, T_r$ satisfy that $s_1(T_s), s_1(T_r) \leq \sigma_1 \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{m} + \sqrt{nk})$ by Lemma 6.

• For the parameter $\alpha$ used in $\text{Invert}$: We should choose $\alpha$ such that $1/\alpha \geq \sqrt{5}(s_1(R)^2 + 1) \cdot \sigma_1 \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{m} + \sqrt{nk})$ by Lemma 7.

• For the Gaussian parameter $\sigma_2$ used in $\text{SampleD}$: $\sigma_2 \geq \sqrt{\frac{7}{2}}(s_1(T_s)^2 + 1) \cdot \omega(\sqrt{\log n})$ (see [18, Section 5.4]).

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

7. Insecurity of the Signcryption by Lu et al. [16]

7.1. Description

The signcryption by Lu et al. [16] (called Lu-SC) exploits the basis-based trapdoor mechanism by [28] which consists of algorithms $\text{TrapGen}$, $\text{ExtBasis}$, $\text{SamplePre}$. The Lu-SC includes the following algorithms:

• Setup($1^n$): On input the security parameter $n$, performs the following:
  1. Generate common parameters: $q = poly(n)$, $m = [6n \log q]$, $\bar{L} = O(\sqrt{n \log q})$, $\sigma \geq \bar{L}\omega(\sqrt{m})$, error rate $\alpha = 1/poly(n)$ such that $\alpha q > 2 \sqrt{n}$.
  2. Sample randomly and independently matrices $\mathbf{C}_0, \cdots, \mathbf{C}_r \in \mathbb{Z}_q^{n \times m}$.
  3. A collision-resistant hash function $H_1: \{0, 1\}^* \rightarrow \{0, 1\}^r$, a universal hash function $H_2: \{0, 1\}^* \rightarrow \mathbb{Z}_q^n$.
  4. A message space $\mathcal{M} = \mathbb{Z}_q^n$.

• KeyGen($n$): Do the following:
  1. Use $\text{TrapGen}(1^n)$ 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 $\mathbb{Z}_q^{n \times m}$.
we think that it should be

7.2. An Attack against IND-CPA

b adversary. The adversary wins the game if he can guess correctly the bit

The challenger then chooses uniformly at random a bit

Actually, in [16], the authors wrote

Recall that, in the challenge phase of the IND-CPA security, the adversary is able to check the following conditions:

\[ \sigma \leftarrow \text{SamplePre}(F_{\mu,T_{\mu}},0,\sigma), v \in \mathbb{Z}_q^{2m}, \text{i.e., } F_{\mu,v} = 0 \pmod{q}. \]

6. Output ciphertext \( CT = (c,b_1,b_2). \)

7.2. An Attack against IND-CPA

Recall that, in the challenge phase of the IND-CPA security, the adversary submits two plaintexts \( \mu_0^*, \mu_1^* \) together with the target public key \( pk_{SS^*}. \) The challenger then chooses uniformly at random a bit \( b \in \{0,1\} \) and returns the challenge ciphertext \( CT^* \leftarrow \text{SignCrypt}(\mu_0^*,sk_{SS^*},pk_{SS^*}) \) back to the adversary. The adversary wins the game if he can guess correctly the bit \( b. \)

Given the challenge ciphertext \( CT^* = (c^*,b_1^*,b_2^*) \) of a plaintext either \( \mu_0^* \) or \( \mu_1^* \), the adversary is able to check the following conditions:

\[ \exists (c,b_1,b_2) :|v| \leq \sqrt{2m} \text{ and } [A_S||C_0 + \sum_{i=1}^r (-1)^{h_i} C_i]. \]

\[ v = 0 \pmod{q} \text{ output } \mu; \text{ otherwise, output } \perp. \]

\[ 2_8 \]
\[ v = b_1 - [A_{R^*}C_0]^T(c - \mu_b^*), \text{ and } e = b_2 - [B_{R^*}C_1]^T(c - \mu_b^*), \]

are small,

\[ c - \mu_b^* = H_2(\mu_b^*, pk_{S^*}, pk_{R^*}, v), \]

\[ [A_{S^*}C_0 + \sum_{i=1}^r (-1)^{h_i} C_i]v = 0 \mod q, \text{ where } (h_i)_{i \in [r]} = H_1(\mu_b^*, pk_{R^*}). \]

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. We also showed that some lattice-based signcrypions in the literature neither are secure nor work correctly. Our main tool in the construction is the gadget-based trapdoor technique introduced in [18]. Finding a better way to simplify the equality test mechanism would be an interesting future task.

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] 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.

[9] 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.

[10] 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

[11] 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.

[12] 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.

[13] 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.

[14] F. Gérard, K. Merckx, SETLA: Signature and Encryption from Lattices, 2018, pp. 299–320. doi:10.1007/978-3-030-00434-7_15
[15] X. Yang, H. Cao, W. Li, H. Xuan, Improved lattice-based signcryption in the standard model, IEEE Access 7 (2019) 155552–155562.

[16] 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

[17] 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

[18] 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

[19] 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

[20] 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.

[21] 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

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

[23] 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

[24] 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).

[25] 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.
[26] 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.

[27] 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

[28] 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