Identity Based Proxy Re-encryption Scheme under LWE

Wei Yin[1], **Qiaoyan Wen**[1], **Wenmin Li**[1], **Hua Zhang**[1,2], **and Zheng Ping Jin**[1]
[1,2]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications
Beijing, 100876 - China
[e-mail:zhanghua_288@bupt.edu.cn]
*Corresponding author: Hua Zhang

Received September 28, 2016; revised January 10, 2017; accepted September 3, 2017;
Published December 31, 2017

Abstract

The proxy re-encryption allows an intermediate proxy to convert a ciphertext for Alice into a ciphertext for Bob without seeing the original message and leaking out relevant information. Unlike many prior identity based proxy re-encryption schemes which are based on the number theoretic assumptions such as large integer factorization and discrete logarithm problem. In this paper, we first propose a novel identity based proxy re-encryption scheme which is based on the hardness of standard Learning With Error(LWE) problem and is CPA secure in the standard model. This scheme can be reduced to the worst-case lattice hard problem that is able to resist attacks from quantum algorithm. The key step in our construction is that the challenger how to answer the private query under a known trapdoor matrix. Our scheme enjoys properties of the non-interactivity, unidirectionality, anonymous and so on. In this paper, we utilize primitives include G-trapdoor for lattice and sample algorithms to realize simple and efficient re-encryption.

Keywords: LWE, IBE, lattice, proxy re-encryption, standard model

1. Introduction

In 1998, Blaze, Bleumer and Strauss presented a new primitive called proxy re-encryption [4], of which the unique feature is the intermediation of delegation via proxy re-encryption key. In detail, the new primitive allows an intermediate agent to convert Alice's(delegator) ciphertext to Bob's(recipient) ciphertext by using proxy re-encryption key $rk_{Alice\to Bob}$ so that the latter can decrypt it. Naturally, the agent is required that cannot obtain information of Alice or Bob about the plaintext and the secret keys. For example, Alice can entrust her proxy to re-encrypt her ciphertext to B when she is out traveling. A natural idea is that Alice can decrypt the ciphertext with her private key, then uses Bob's public key to encrypt it. However, this method requires Alice to be always online. In order to solve this problem, Alice and Bob together set up proxy re-encryption key $rk_{Alice\to Bob}$, this situation is called interactive, as opposed to it, there is non-interactive that $rk_{Alice\to Bob}$ can be generated by Alice alone. The $rk_{Alice\to Bob}$ is stored in a semi-trusted server, and allows proxy to delegate Alice's ciphertext, and the server cannot decrypt it.

Green and Ateniese first proposed the identity based proxy re-encryption schemes(IB-PRE). It allows proxy to convert a ciphertext for Alice under Alice's identity to one encrypted ciphertext under Bob's identity[10][6][8]. The proxy employs a re-encryption key to complete the transformation, in the process, it could not obtain any information about the message and the private keys of Alice and Bob from the proxy re-encryption key.

Lattice based cryptography has been developed rapidly in recent years[3][2][1][13][14][15], due to the following advantages: (1)Number theoretic hard problem, large integer factoring problem and the discrete logarithm problem can be solved by quantum algorithms, so cryptographic protocols based on those problems in quantum computing system environment are no longer safe. However, so far, there have been no effective quantum algorithms to solve lattice hard problems. (2)Traditional cryptosystem is based on the hard problems in the average case. However, lattice-based cryptography is based on the hard problems in the worst case, which is stronger security.

2. Related Work

Blaze et.al proposed the first proxy re-encryption scheme in 1998[4]. Their scheme is based on the ElGamal encryption construction and it is CPA secure under the Decisional Diffie-Hellman assumption. Their re-encryption key from user A to B is $rk_{A\to B}=b/a$, in which a and b are the private keys of A and B. Through these useful information, the proxy can easily generate the re-encryption key b/a to convert ciphertext of A to B, then it also allows to convert the ciphertext of B from the opposite direction to A. So PRE schemes like that are called bidirectional, but more desirable schemes in practical application are unidirectional for any bidirectional scheme could always be acquired by calling a unidirectional one in both directions, where the proxy only works in one direction by the re-encryption key. The possibility of using lattice as a tool for PRE scheme was shown in [16] by Xagawa, but the scheme lacks a complete formalization security analysis.

Aono et.al first proposed a CPA-secure PRE scheme based on the hardness of the standard Learning-With-Errors(LWE) problem in the standard model[9]. A unidirectional single-hop PRE based on the hardness of lattice based problem was proposed by Kirshanova[7], which is

the first lattice based construction that achieves collusion resilience and non-interactivity. Green and Ateniese presented first proxy re-encryption scheme in the identity based setting[10], which is based on Decisional Diffie-Hellman assumption. Singh et.al proposed a lattice based identity based PRE scheme in the random oracle model for the single bit plaintext as well as for the multi bit plaintext [6], which is anonymous, bidirectional and multi-hop, but it does not meet the security requirement in the standard model, however, there is a potential threat in their construction, if the proxy and one of the parties collude, they can recover the secret key of another party easily, another issue is that a proxy obtained $rk_{i,k} = sk_{id_i} - sk_{id_k}$ through calculating $rk_{i,j} = sk_{id_i} - sk_{id_j}$ and $rk_{j,k} = sk_{id_j} - sk_{id_k}$.

Our Contribution

According to our knowledge, there does not exist any lattice based identity based PRE scheme in the standard model. In this paper, we first put forward a lattice based identity based proxy re-encryption scheme in the standard model. Our scheme satisfies the following properties[7][6]:

- Unidirectional: Unidirectional scheme allows proxy to convert a ciphertext of A to B in only one direction, but not vice versa.
- Non-interactivity: Re-encryption key $rk_{A\to B}$ can be generated by A alone using public key of B , the process does not require the participation of B and the proxy.
- Proxy transparency: Either A or B are not aware of the presence of the proxy. That is, recipients can not distinguish whether received ciphertext is the encryption by public key of recipient directly or re-encrypted by the proxy.
- Collusion resilience: The coalition of the proxy and recipient can not compute to obtain delegator's private key.
- Non-transitivity: It is hard for the proxy to re-delegate the decryption right, namely, to obtain $rk_{A\to C}$ from $rk_{A\to B}$ and $rk_{B\to C}$ by any method.
- Anonymous: Ciphertext does not reveal any information about the identity of the recipient.
- Multi-hop: A multi-hop scheme allows the proxy to perform multiple re-encryptions for a ciphertext, i.e. re-encrypt a ciphertext from A to B, then re-encrypt the result from B to C.

In our scheme, for the sake of CPA security we use the sample algorithms, include $SampleRight$ and $SampleLeft$ technologies in [3], to guarantee that master secret key could produce every identities' private key in the real system and simulator to answer private key query in the proof. In order to achieve the function of re-encryption, we use the $\mathbf{G}$ -trapdoor for lattice technology in [2] which solved LWE problem efficiently for its special structure, but it is not a simple combination of the two technologies that could achieve our desired results. Specifically, the user id_i's private key is separated into two components-(e_i , R_i), the generation of e_i by using sample function $SampleLeft$ is for correct decryption, the other one R_i is built to achieve the effect of re-encryption. When we designed our scheme, there was an obstacle: in the process of security proof, the adversary interacts with the challenger who simulate scheme so that the adversary cannot distinguish it with real scheme, the adversary obtains the user's private key by asking the challenger private key query for he cannot get the private key on his own. The original purpose of using $\mathbf{G}$ -trapdoor is to generate the re-encryption key, but the adversary can obtain the user's private key through the algorithm $e \leftarrow SampleRight(A|-A\mathbf{R}+H\mathbf{G})$ since $\mathbf{G}$ is public parameters(We employ the gadget matrix $\mathbf{G} = \mathbf{I} \otimes \mathbf{g}^{\intercal}$ with $\mathbf{g}^{\intercal} = (1,2,...2^{k-1})$ in the interest of sampling vectors according to the discrete

Gaussian distribution, so the basis of $\Lambda_q^{\perp}(\mathbf{G})$ is easy to get.)if we apply this $\mathbf{G}$-trapdoor to IBE of [3] directly.

To solve this dilemma and avoid affecting normal private key query with challenger in proof, we introduce a random uniform matrix parameters T which constitutes a binding form of $T\mathbf{G}$. With this approach, the information of $\mathbf{G}$ is hidded. In this case the adversary will not be able to obtain user's private keys except $e \leftarrow SampleLeft(A|-A\mathbf{R}+HT\mathbf{G})$ by a basis T_A for $\Lambda_q^{\perp}(A)$. And the challenger still answers private key query via algorithm $e \leftarrow SampleRight(A|-A\mathbf{R}+HT\mathbf{G})$ with a basis T_G for $\Lambda_q^{\perp}(\mathbf{G})$.

Table 1. Comparison with the previous schemes

Authors	Security	Unidirectional	Standard Model	Assumption	Identity Based
Singh2013	CPA	×	×	LWE	√
Chu2007	CPA&CCA	√	√	BDH	√
MG2007	CPA&CCA	√	×	DBDH	√
PKC2014	CCA	√	√	LWE	×
Ours	CPA	√	√	LWE	√

Paper Outline

The rest of our paper is organized as follows. In section 2 we give some basic definitions, hard problems, some conclusions in lattice, and IB-PRE security model. In section 3 we show strong trapdoors in [2] and sample algorithms in [3]. In section 4 we present a CPA-secure IB-PRE scheme in the standard model, and prove the security of our scheme in section 5. In section 6 we conclude this paper.

3. Preliminaries

3.1 Identity-Based Unidirectional Proxy Re-encryption Scheme (IB-uPRE)

An Identity-Based unidirectional proxy re-encryption scheme is a tuple of algorithms-(**Setup, Extract, Encrypt, ReKeyGen, ReEnc, Decrypt**)[6]:

- **Setup(λ):** On input a security λ, output the public parameters *PP* and master secret key MK.
- **Extract(*PP*, *MK*, *id*):** On input public parameters *PP*, master secret key *MK*, and an identity id, output the private key SK_{id} corresponding to the identity id.
- **Encrypt(*PP*, id, *M*):** On input public parameters *PP*, an identity id, and a message *M*, this algorithm outputs ciphertext C_{id}.
- **ReKeyGen(*PP*, sk_{id_i}, id_i, id_j):** On input a secret key sk_{id_i}, the algorithm output a re-encryption key $rk_{i,j}$.
- **ReEnc(*PP*, C_{id_i}, $rk_{i,j}$):** On input a ciphertext C_{id_i} of identity id_i and re-encryption key $rk_{i,j}$, the algorithm outputs a re-encrypted ciphertext C_{id_j} for an identity id_j.
- **Decrypt(*PP*, sk_{id}, C_{id}):** On input public parameters *PP*, a private key sk_{id} of identity id and a ciphertext C_{id}, this algorithm outputs message *M*.

Correctness Identity-Based Unidirectional Proxy Re-encryption Scheme is correct if:

- For all PP, sk_{id} outputted by **Extract** and for all M in plaintext space, it holds that **Decrypt(** sk_{id} ,**Encrypt(** id , M**))=***M*.
- For re-encryption key $rk_{i,j}$ outputted by **ReKeyGen** and for any C_{id_i} outputted by **Encrypt (***PP*, id_i ,*M***),** and for all M in plaintext space, it holds that **Decrypt(** sk_{id} ,**ReEnc(** $rk_{i,j}$, C_{id_i} **)=***M*.

Definition 1. *A proxy re-encryption system is called multi-hop if a proxy can re-encrypted the encrypted ciphertext repeatedly. By comparison in a signle-hop system, a ciphertext can be re-encrypted only once.*

Whether our system is single-hop or multi-hop, the requirements of correctness for decryption are the same. That is to say, we can obtain the plaintext message by using decryption algorithm from the resulting ciphertext. No matter what the ciphertext is, it just needs produced or re-encrypted.

Security Game[8,10] We define IB-uPRE selective−ID security using a series of games that are played between the challenger and the adversary. This security includes semantic security and recipient anonymity. The game plays as follows.

Before introducing the game model, we first divide all users into two categories: honest user and corrupted user. HU represents honest user that the adversary only knows their public key, and CU represents corrupted user that the adversary not only knows their public key, but also knows their private key. We let $\mathcal{M}$ denote the message space and let $\mathcal{C}$ denote the ciphertext space.

Init The adversary publishes the target identity id^*, which he wants to attack.

Setup The challenger runs **Setup(** 1^n **)** and gives the public parameters *PP* to adversary and keeps the master key *MK* to itself. HU and CU are defined as above.

Phase 1 The adversary can make the following queries:

- The adversary can ask a private key query on identity id except identity id^*, the challenger responds by running **Extract** algorithm to generate a private key sk_{id} for identity id and sents it to the adversary. The adversary can repeat issue query for different identities polynomial times.
- The adversary can ask a re-encryption key query $rk_{i,j}$ from identity id_i to identity id_j, the challenger responds by running **ReKeyGen** algorithm to generate a re-encryption key $rk_{i,j}$ from identity id_i to identity id_j and sents it to the adversary, all queries where $i = j$ or $i \in HU$, $j \in CU$ are ignored. The adversary can repeat polynomial times for different couple of identities.
- The adversary can ask re-encryption query C_{id_j} from (id_i, id_j, C_{id_i}) , the challenger responds by running **ReKeyGen** algorithm to generate a re-encryption key $rk_{i,j}$ from identity id_i to identity id_j and then the challenger generates ciphertext C_j by running **ReEnc** algorithm, and returns C_j to the adversary. All queries where $i = j$ or $i \in HU$, $j \in CU$ are ignored. The adversary can repeat polynomial times for different couples of identities.

Challenge Once adversary considers that Phase 1 could be over then it outputs a plaintext $M \in \mathcal{M}$ which he wishes to challenge on, and submits identity id^* and *M* to challenger, id^* should be in HU. The challenger picks a random bit $r \in \{0,1\}$ and a random ciphertext *C*. If $r = 0$, it sets the challenge ciphertext to $C_{id^*} = Encrypt(PP, id^*, m)$. If $r = 1$, it sets the challenge

ciphertext to $C_{id^*} = C$. Afterwards it sends C_{id^*} as a challenge ciphertext to the adversary.

Phase 2 The adversary could ask extra queries that for private key query, re-encryption key query and re-encryption query on the identity $id \neq id^*$, the challenger responds are the same as in Phase 1.

Guess Finally, the adversary outputs a guess $r' \in \{0,1\}$ and wins if $r = r'$.

We refer to the adversary $\mathcal{A}$ in above game as an IND-sID-CPA adversary. We define the advantage of the adversary $\mathcal{A}$ in attacking an IB-uPRE scheme ε as

$$Adv_{\varepsilon,\mathcal{A}} = |\Pr[r = r'] - \frac{1}{2}|$$

Definition 2. *We say that an IB-uPRE scheme is IND-sID-CPA if for all probabilistic polynomial time algorithm $\mathcal{A}$ and negligible function ε, we always have that $Adv_{\varepsilon,\mathcal{A}}$ is a negligible function, that is, $Adv_{\varepsilon,\mathcal{A}} \leq \varepsilon$.*

3.2 Lattice Definition

Definition 3 (*Integer Lattice*[13,15]). *Let $B = [b_1 | ... | b_m] \in \mathbb{R}^{m \times m}$ be a $m \times m$ matrix whose columns are linearly independent vectors $b_1, ..., b_m \in \mathbb{R}^m$. The m-dimensional full-rank lattice Λ generated by B is the set,*

$$\Lambda = \mathcal{L}(B) = \{y \in \mathbb{R}^m \ s.t. \ \exists s \in \mathbb{Z}^m, y = Bs = \sum_{i=1}^{m} s_i b_i\}$$

Here, we are interested in integer lattices, i.e., when $\mathcal{L}$ is contained in $\mathbb{Z}^m$. We let $\det(\Lambda)$ denote the determinant of Λ.

Definition 4 (*q-ary lattice*). *For prime q, $A \in \mathbb{Z}_q^{n \times m}$ and $u \in \mathbb{Z}_q^n$, define:*

$$\Lambda(A)_q := \{e \in \mathbb{Z}^m \ s.t. \ \exists s \in \mathbb{Z}_q^n \ where \ A^\top s = e (\mathrm{mod}\, q)\}$$

$$\Lambda_q^\perp(A) := \{e \in \mathbb{Z}^m \ s.t. \ Ae = 0 (\mathrm{mod}\, q)\}$$

$$\Lambda_q^u(A) := \{e \in \mathbb{Z}^m \ s.t. \ Ae = u (\mathrm{mod}\, q)\}$$

We can observe that if $t \in \Lambda_q^u(A)$ then $\Lambda_q^u(A) = \Lambda_q^\perp(A) + t$ and hence $\Lambda_q^u(A)$ is a shift of $\Lambda_q^\perp(A)$

3.3 The Gram-Schmidt Norm

Definition 5 (*Gram-Schmidt norm*[13]). *Let S be a set of vectors $S = \{s_1, ..., s_k\}$ in $\mathbb{R}^m$. We use the following standard notations:*

- *$\|S\|$ denotes the L_2 length of the longest in S, i.e., $\max_{1 \leq i \leq k} \|s_i\|$.*
- *$\tilde{S} = \{\tilde{s_1}, ..., \tilde{s_k}\} \subset \mathbb{R}^m$ denotes the Gram-Schmidt orthogonalization of the vectors $s_1, ..., s_k$ taken in that order.*

We refer to $\|\tilde{S}\|$ as the Gram-Schmidt norm of S.

Lemma 1 ([17], Lemma 7.1). *There is a deterministic poly-time algorithm ToBasis(S,B) that, given a full rank set S of lattice vectors in $\Lambda = \mathcal{L}(B)$, outputs a basis T of Λ such that $\|\tilde{t_i}\| \leq \|\tilde{s_i}\|$ for all i.*

In 1996, Ajtai[18] showed how to sample an essentially uniform matrix $A \in \mathbb{Z}_q^{n \times m}$ with an associated basis S_A of $\Lambda_q^\perp(A)$ with low Gram-Schmidt norm. Here we use an improved algorithm from [1]. The following Theorems 3.2 are derived from [1] taking $\sigma := \frac{1}{3}$.

Theorem 1. *Let* $q \geq 3$ *be odd and* $m := \lceil 6n\log q \rceil$. *There is a probabilistic polynomial-time algorithm* $TrapGen(q,n)$ *that outputs a pair* $(A \in \mathbb{Z}_q^{n\times m}, S \in \mathbb{Z}^{n\times m})$ *such that* A *is statistically close to a uniform matrix in* $\mathbb{Z}_q^{n\times m}$ *and* S *is a basis for* $\Lambda_q^{\perp}(A)$ *satisfying*

$$\left\|\tilde{S}\right\| \leq O(\sqrt{n\log q}) \text{ and } \|S\| \leq O(n\log q)$$

with all but negligible probability in n.

3.4 The LWE Problems

Construction of this paper reduces to the **Learning with Errors** problem, which may be seen as average case problem related to the family of lattices described above.

Definition 6 (Learning with Errors[19]). *For a prime* q, *a positive integer* n, *and a distribution* $\mathcal{X}$ *over* $\mathbb{Z}_q$, *the* $LWE_{q,\mathcal{X}}$ *problem is to distinguish, given oracle access to any desired* $m = poly(n)$ *samples, between the distribution* $A_{s,\mathcal{X}}$ *(for uniformly random and secret* $\boldsymbol{s} \in \mathbb{Z}_q^n$*)and the uniform distribution over* $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

We give an outline of Gaussian distributions over lattice. For any $s > 0$ and dimension $m \geq 1$, the Gaussian function $\rho_s : \mathbb{R}^m \to (0,1]$ is defined as $\rho_s(\boldsymbol{x}) = \exp(-\pi \| \boldsymbol{x} \|^2 / s^2)$. For any coset $\Lambda_y^{\perp}(A)$, and probability zero elsewhere. We summarize several facts from the literature about discrete Gaussian over lattices, again specialized to our family of interest.

Lemma 2 ([21], Lemma 4.4). *For any n-dimensional lattice* Λ, *vector* $\boldsymbol{c} \in \mathbb{R}^n$, *and reals* $0 < \varepsilon < 1$, $s \geq \eta_\varepsilon(\Lambda)$, *we have*

$$\Pr_{x \sim \mathcal{D}_{\Lambda,s,c}} \{ \| \boldsymbol{x} - \boldsymbol{c} \| > s\sqrt{n} \} \leq \frac{1+\varepsilon}{1-\varepsilon} \cdot 2^{-n}$$

Lemma 3 ([13]). *There are two PPT algorithms* $SampeGaussia(A, T_A, \sigma, \mathbf{c})$ *and a PPT algorithm* $SampePre(A, T_A, \sigma, u)$, *the former returns* $x \in \Lambda_q^{\perp}(A)$ *drawn from a distribution statistically close to* $\mathcal{D}_{\Lambda,s,\mathbf{c}}$, *and the latter returns* $x \in \Lambda_q^u(A)$ *sampled from a distribution statistically close to* $\mathcal{D}_{\Lambda_q^u(A),\sigma}$, *whenever* $\Lambda_q^u(A)$ *is not empty, where* T_A *be a basis for* $\Lambda_q^{\perp}(A)$ *and* $\sigma \geq \|\widetilde{T_A}\| \omega(\sqrt{logm})$, *for* $\mathbf{c} \in \mathbb{R}^m$ *and* $u \in \mathbb{Z}_q^n$.

3.5 Encoding Identities as Matrices

Our construction needs a function $H : \mathbb{Z}_q^n \leftarrow \mathbb{Z}_q^{n\times n}$ which could map identities(in $\mathbb{Z}_q^n$) to matrices(in $\mathbb{Z}_q^{n\times n}$), and the proof of our scheme's security requires the function to satisfy a strong injectivity, i.e., for two distinct identities id_1, id_2, $det(H(id_1) - H(id_2)) \neq 0$.

Definition 7. *For a prime* q *and a positive integer* n. *We say that a function* $H : \mathbb{Z}_q^n \leftarrow \mathbb{Z}_q^{n\times n}$ *is an encoding with full rank differences(FRD) if:*

1. *For all distinct* $u, v \in \mathbb{Z}_q^n$, *the matrix* $H(u) - H(v) \in \mathbb{Z}_q^{n\times n}$ *is full rank.*
2. *H is computable in polynomial time.*

We use an injective FRD encoding function that is described in [20]. A short instruction is as follows: For the finite field $\mathbb{Z}_q$, a polynomial $g \in \mathbb{F}[X]$ of degree less than n, $\text{coeffs}(g) \in \mathbb{F}^n$ be defined as n-vector which the element is coefficients of g. Let f be some polynomial of degree n in $\mathbb{F}[X]$ that is irreducible. For input $u = (u_0, u_1, \ldots, u_{n-1})$, define the polynomial $g(x) = \sum_{i=0}^{n-1} u_i x^i$.

Define $H(u)$ as

$$H(u) := \begin{pmatrix} coeffs(g) \\ coeffs(x \cdot g \bmod f) \\ coeffs(x^2 \cdot g \bmod f) \\ \vdots \\ coeffs(x^{n-1} \cdot g \bmod f) \end{pmatrix} \in \mathbb{F}^{n \times n} \quad (1)$$

Theorem 2 ([20]). *Let* $\mathbb{F}$ *be a field and* f *a polynomial in* $\mathbb{F}[X]$. *If* f *is irreducible in* $\mathbb{F}(X)$, *then the function H defined in (1) is an encoding with full rank differences.*

4. G-trapdoor and Sample Algorithms

In this section, we briefly describe the main results in [2] and [3], which be used in our construction: the definition of a so called G-trapdoor and the sampling algorithms include *SampleLeft* and *SampleRight*.

4.1 G-trapdoor Generation

In brief, a G-trapdoor is a key matrix that can transform a public matrix $\mathbf{A}$ to a special matrix G. This trapdoor(represented by a matrix $\mathbf{R}$) has special properties, and includes algorithms for sampling *SIS* preimages and inverting *LWE* problems, which be admitted very efficient and high quality, and these problems are considered to be hard if for a uniform $\mathbf{A}$. There is an example of G-trapdoor in [2] make $\mathbf{G} := \mathbf{I}_n \otimes \mathbf{g}^t \in \mathbb{Z}_q^{n \times nk}$

$$G := \begin{pmatrix} \cdots g^t \cdots & & & \\ & \cdots g^t \cdots & & \\ & & \ddots & \\ & & & \cdots g^t \cdots \end{pmatrix} \in \mathbb{Z}_q^{n \times nk}$$

where

$$g^t := [1\ 2\ 4 \cdots 2^{k-1}] \in \mathbb{Z}_q^{1 \times k}, k = \lceil log_2 q \rceil$$

There are efficient algorithms for inverting $g_G(s,e) := s^t G + e^t \bmod q$ and preimage Gaussian sampling for $f_G(x) = Gx \bmod q$.

Definition 8. ([2], Definition5.2). *Let* $A \in \mathbb{Z}_q^{n \times m}$ *and* $G \in \mathbb{Z}_q^{n \times w}$ *be matrices with* $m \geqslant w \geqslant n$. *A* $\mathbf{G}$*-trapdoor for* A *is a matrix* $\mathbf{R} \in \mathbb{Z}^{(m-w) \times w}$ *such that* $A\begin{pmatrix} \mathbf{R} \\ I \end{pmatrix} = H\mathbf{G}$ *for some invertible matrix* $H \in \mathbb{Z}_q^{n \times n}$. *We refer to* H *as the tag or label of the trapdoor. The quality of the trapdoor is measured by its largest singular value* $s_1(R)$.

We construct $A = [A_0 \mid -A_0\mathbf{R} + \mathbf{G}]$, where A_0 is a uniform matrix, and $\mathbf{R}$ is a transformation in order to make A with this structured matrix. By the Leftover Hash Lemma, let appropriate parameters, $(A, A\mathbf{R})$ is negl(n)-far from uniform. This $\mathbf{R}$ can be transformed as follows:

$$[A_0 \mid -A_0\mathbf{R} + \mathbf{G}]\begin{pmatrix} \mathbf{R} \\ I \end{pmatrix} = \mathbf{G}$$

In order to construct a CPA-secure encryption scheme, there is an invertible matrix H can be used like this

$$[A_0 \mid -A_0\mathbf{R} + H\mathbf{G}]\begin{pmatrix}\mathbf{R}\\ I\end{pmatrix} = H\mathbf{G}$$

In this situation, to complete the transformation and answer adversary's queries must know both $\mathbf{R}$ and H. Once the matrix H is zero matrix, then the challenger cannot answer adversary's queries, because of $[A_0 \mid -A_0\mathbf{R}]\begin{pmatrix}\mathbf{R}\\ I\end{pmatrix} = 0$, and solving *LWE* or *SIS* problems cannot be reduced to solve the same problem about G. Therefore, in this case, the challenger can make a challenge ciphertext.

LWE Inversion. We give an effective algorithm which denoted by $Invert^{\mathcal{O}}(R, A, b)$ for inverting the function $g_{\mathbf{A}}(s, \mathbf{e}) = \mathbf{s}^t A + \mathbf{e}^t$ for any $s \in \mathbb{Z}_q^n$ and suitably small vector $\mathbf{e} \in \mathbb{Z}^m$, $A \in \mathbb{Z}_q^{n\times m}$ is a parity-check matrix, $\mathbf{G}$-trapdoor $\mathbf{R}$ for A with invertible tag H. There is an oracle $\mathcal{O}$ for inverting the function $g_{\mathbf{G}}(\hat{\mathbf{s}}, \hat{\mathbf{e}})$ where $\hat{\mathbf{e}} \in \mathbb{Z}^w$ is small. Then we compute $\hat{\mathbf{b}}^t = \mathbf{b}^t\begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix}$, get $(\hat{\mathbf{s}}, \hat{\mathbf{e}}) \leftarrow \mathcal{O}(\hat{\mathbf{b}})$, return $\mathbf{s} = \mathbf{H}^{-t}\hat{\mathbf{s}}$, $\mathbf{e} = \mathbf{b} - \mathbf{A}^t\mathbf{s}$ and output vectors $(\mathbf{s}, \mathbf{e})$.

Gaussian Sampling. We show how to sample a discrete Gaussian over $\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A})$, and this efficient algorithm is denoted as $Sample^{\mathcal{O}}(\mathbf{R}, \overline{\mathbf{A}}, \mathbf{H}, \mathbf{u}, s)$. There is an oracle $\mathcal{O}$ for sampling over a desired coset $\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G})$ with fixed parameter $r\sqrt{\Sigma_{\mathbf{G}}} \geq \eta_{\varepsilon}(\Lambda^{\perp}(\mathbf{G}))$, for some $\Sigma_{\mathbf{G}} \geq 2$ and $\epsilon \leqslant \frac{1}{2}$. We define matrix $\mathbf{A} = [\overline{\mathbf{A}} \mid \mathbf{HG} - \overline{\mathbf{A}}\mathbf{R}]$ with invertible tag $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, a parity-check matrix $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n\times \bar{m}}$, $\mathbf{G}$-trapdoor matrix $\mathbf{R} \in \mathbb{Z}^{\bar{m}\times w}$, positive definite $\sum \geqslant \begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix}(2 + \Sigma_{\mathbf{G}})\begin{bmatrix}\mathbf{R}^t & \mathbf{I}\end{bmatrix}$ and syndrome $\mathbf{u} \in \mathbb{Z}_q^n$. In order to obtain a vector x drawn from a distribution within $O(\epsilon)$ statistical distance of $D_{\Lambda_{\mathbf{u}}^{\perp}(A), r\sqrt{\Sigma}}$, first choose a fresh perturbation $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, r\sqrt{\Sigma_p}}$, let $\mathbf{P} = \begin{bmatrix}\mathbf{p}_1\\ \mathbf{p}_2\end{bmatrix}$ for $\mathbf{p}_1 \in \mathbb{Z}^{\bar{m}}$, $\mathbf{p}_2 \in \mathbb{Z}^w$, compute $\overline{\mathbf{w}} = \overline{\mathbf{A}}(\mathbf{p}_1 - \mathbf{R}\mathbf{p}_2)$ and $\mathbf{w} = \mathbf{G}\mathbf{p}_2$, let $v \leftarrow H^{-1}(\mathbf{u} - \overline{\mathbf{w}}) - \mathbf{w} = H^{-1}(\mathbf{u} - \mathbf{AP})$, choose $z \leftarrow D_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}), r\sqrt{\Sigma_G}}$ by calling $\mathcal{O}(v)$, and return $x \leftarrow \mathbf{P} + \begin{pmatrix}\mathbf{R}\\ \mathbf{I}\end{pmatrix}z$.

4.2 SampleLeft and SampleRight

In short, lattices in this system are built with two parts called “Left” and “Right” lattices. In the real system, a trapdoor for left lattices is used as master secret to generate every user's private key. While in the simulation system, a trapdoor for right lattice is used to generate private keys for all identities except for one, which is selected by the adversary[3].

Let $A, B \in \mathbb{Z}_q^{n\times m}$ and $R \in \{-1, 1\}^{m\times m}$. Our framework is constructed by form $F = (A \mid AR + B)$, and we use different methods(Precisely, it is from a different direction) to sample short vectors from $\Lambda_q^u(F)$ for some $u \in \mathbb{Z}_q^n$. These two specific algorithms are in the following :

- Algorithm *SampleLeft*

SampleLeft takes a basis for $\Lambda_q^{\perp}(A)$ (i.e., the left side of F) and outputs a short vector $e \in \Lambda_q^u(F)$. Specifically, in algorithm $SampleLeft(A, M_1, T_A, u, \sigma)$, $A \in \mathbb{Z}_q^{n\times m}$ is a rank n matrix, a matrix $M_1 \in \mathbb{Z}_q^{n\times m}$, a short basis T_A of $\Lambda_q^{\perp}(A)$ and a vector $u \in \mathbb{Z}_q^n$ as input, gaussian parameter

$\sigma > \widetilde{T_A} \cdot w(\sqrt{log(m+m_1)})$. Then it samples a random vector $e_2 \in \mathbb{Z}^{m_1}$ distributed statistically close to $D_{\mathbb{Z}^{m_1},\sigma}$, and runs $e_1 \leftarrow SamplePre(A,T_A,y,\sigma)$ where $y = u - (M_1 \cdot e_2) \in \mathbb{Z}_q^n$, outputs $e \leftarrow (e_1,e_2)$.

- Algorithm *SampleRight*

SampleRight takes a basis for $\Lambda_q^{\perp}(B)$ (i.e., the right side of F) and outputs a short vector $e \in \Lambda_q^u(F)$. Specifically, Algorithm $SampleRight(A,B,R,T_B,u,\sigma)$ where $A \in \mathbb{Z}_q^{n\times k}$, $B \in \mathbb{Z}_q^{n\times m}$ where B is rank n, and a matrix $R \in \mathbb{Z}^{k\times m}$, where $s_R := \|R\|$, with a basis T_B of $\Lambda_q^{\perp}(B)$ and a vector $u \in \mathbb{Z}_q^n$ as input. Then it constructs a set T_{F_2} of $(m+k)$ linearly independent vectors in $\Lambda_q^{\perp}(F_2)$, uses Lemma 3 to convert T_{F_2} into a basis $T_{F_2'}$ of $\Lambda_q^{\perp}(F_2)$ with same Gram-Schmidt norm, invokes $SamplePre(F_2,T_{F_2'},u,\sigma)$ to generate a vector $e \in \Lambda_q^u(F_2)$ as output.

5. The Basic Construction

In our construction, we need to publish identity identifier for each user, that is $[A_0 \mid -A_0R_i]$, in which the function of R_i is used to implement re-encryption, for $i = 1,\ldots,n$, n is the number of users.

Setup(λ): On input a security parameter λ, set the parameters q,n,m,σ,α as specified before. then do as follows:

- Use algorithm $TrapGen(q,n)$ to select a uniformly random $n\times m-matrix$ $A_0 \in \mathbb{Z}_q^{n\times m}$ with a basis T_{A_0} such that $\|\widetilde{T_{A_0}}\| \le O(\sqrt{n\log q})$.
- Select $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ with special structure, and select a uniformly random $n\times m-matrix$ $T \in Z_q^{n\times n}$.
- Select $\mathbf{R_1},\mathbf{R_2},\ldots,\mathbf{R_n} \in \mathcal{D}$ in which n is the number of users($\mathbf{R_i}$ is sampled from the Gaussian $\mathcal{D} = \mathcal{D}_{\mathbb{Z},\omega\sqrt{logm}}^{m\times m}$), and id_i will be assigned by $\mathbf{R_i}$ for $i = 1,\ldots,n$.
- Select a uniformly random n-vector $u \xleftarrow{R} \mathbb{Z}_q^n$.
- An encoding function $H: \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$.

$$PP = (A_0,u,H,T\mathbf{G},A_0\mathbf{R_i},i = 1,\ldots,n); MK = (\mathbf{G},T_{A_0},\mathbf{R_1},\mathbf{R_2},\ldots,\mathbf{R_n})$$

Extract(PP,MK,id_i): On input public parameters PP, master key MK, and an identity $id_i \in \mathbb{Z}_q^n$, do:

- Sample $e_i \leftarrow SampleLeft(A_0,-A_0\mathbf{R_i}+H(id_i)T\mathbf{G},T_{A_0},u,\sigma)$
- Assigned to id_i $\mathbf{R_i}$.
- Output $SK_{id_i} := (e_i,\mathbf{R_i})$

Let $F_{id_i} := (A_0 \mid -A_0\mathbf{R_i}+H(id_i)T\mathbf{G})$, then $F_{id_i}\cdot e_i = u$, and e_i is distributed as $D_{\Lambda_q^u(F_{id_i}),\sigma}$.

Encrypt(PP,id_i,b): On input public parameters PP, an identity id_i, and a message $b \in \{0,1\}$, do:

- Set $F_{id_i} \leftarrow (A_0 \mid -A_0\mathbf{R_i}+H(id_i)T\mathbf{G}) \in \mathbb{Z}_q^{n\times 2m}$.
- Choose a uniformly random $s \leftarrow \mathbb{Z}_q^n$.
- Choose a uniformly random $m\times m$ matrix $\mathbf{R} \xleftarrow{R} \{-1,1\}^{m\times m}$.
- Choose noise vector $x \xleftarrow{\Psi_\alpha} \mathbb{Z}_q^n$ and $y \xleftarrow{\Psi_\alpha^m} \mathbb{Z}_q^m$, and set $z \leftarrow \mathbf{R}^\top y \in \mathbb{Z}_q^m$.

- Set $c_0 \leftarrow s^{\top}u + x + b\left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$ and $c_1 \leftarrow s^{\top}F_{id_i} + \left[y^{\top} \mid z^{\top} \right] \in \mathbb{Z}_q^{1\times 2m}$.
- Output the ciphertext $CT_{id_i} := (c_0, c_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{1\times 2m}$.

ReKeyGen$(PP, id_i, id_j, SK_{id_i})$: On input $F_{id_i} \leftarrow (A_0 \mid -A_0\mathbf{R_i} + H(id_i)T\mathbf{G})$ and $F_{id_j} \leftarrow (A_0 \mid -A_0\mathbf{R_j} + H(id_j)T\mathbf{G})$. do:

- Use the second part of a secret key - the Gaussian matrix $\mathbf{R_i}$ and the invertible $H(id_i) \in \mathbb{Z}_q^{n\times n}$, execute $Sample^{\mathcal{O}}$ to sample from the cosets of the $\left[-A_0\mathbf{R_j} + H(id_j)T\mathbf{G} \right]$. Specifically, we sample column-wise so that for each column of the $\left[-A_0\mathbf{R_j} + H(id_j)T\mathbf{G} \right]$, we obtain a $2m-$ dimensional column of the re-encryption key. After sampling m times we can receive an $2m \times m$ matrix and parse it as two matrices $\mathbf{X}_3 \in \mathbb{Z}_q^{m\times m}$ and $\mathbf{X}_4 \in \mathbb{Z}_q^{m\times m}$

$$\left[A_0 \mid -A_0\mathbf{R_i} + H(id_i)T\mathbf{G} \right] \begin{bmatrix} \mathbf{X_3} \\ \mathbf{X_4} \end{bmatrix} = \left[-A_0\mathbf{R_j} + H(id_j)T\mathbf{G} \right]$$

- To set up the equation, continue sampling for the cosets obtained from the columns of the matrix $[A_0]$

$$\left[A_0 \mid -A_0\mathbf{R_i} + H(id_i)T\mathbf{G} \right] \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix} = \left[A_0 \right]$$

- The re-encryption key is a matrix with Gaussian entries:

$$rk = \begin{pmatrix} \mathbf{X_1} & \mathbf{X_3} \\ \mathbf{X_2} & \mathbf{X_4} \end{pmatrix} \in \mathbb{Z}_q^{2m\times 2m}$$

ReEnc(PP, CT_{id_i}, rk) : Compute the component c_1 in the ciphertext as follows :

- Compute

$$\begin{aligned} c_1' &= c_1 \cdot \begin{pmatrix} \mathbf{X_1} & \mathbf{X}_3 \\ \mathbf{X}_2 & \mathbf{X}_4 \end{pmatrix} = (s^{\top}F_{id_i} + \left[y^{\top} \mid z^{\top} \right]) \cdot \begin{pmatrix} \mathbf{X_1} & \mathbf{X}_3 \\ \mathbf{X}_2 & \mathbf{X}_4 \end{pmatrix} \\ &= (s^{\top}F_{id_i} + \left[y^{\top}\mathbf{X_1} + z^{\top}\mathbf{X}_2 \mid y^{\top}\mathbf{X}_3 + z^{\top}\mathbf{X}_4 \right]) \\ &= (s^{\top}F_{id_i} + \left[y'^{\top} \mid z'^{\top} \right]) \end{aligned}$$

- Output ciphertext for id_2 :

$$CT_{id_j} := (c_0, c_1') = (s^{\top}u + x + b\left\lfloor \frac{q}{2} \right\rfloor, s^{\top}F_{id_j} + \left[y'^{\top} \mid z'^{\top} \right])$$

Decrypt(PP, CT_{id_j}, SK_{id}) : On input public parameters PP, a private key $SK_{id} := e_{id}$, and a ciphertext $CT = (c_0, c_1')$, do:

- Compute $w = c_0 - c_1'e_{id} \in \mathbb{Z}_q$.
- Compare w and $\left\lfloor \frac{q}{2} \right\rfloor$ treating them as integers in $\mathbb{Z}$. If they are close, i.e., if $\left| w - \left\lfloor \frac{q}{2} \right\rfloor \right| < \left\lfloor \frac{q}{4} \right\rfloor \in \mathbb{Z}$, output 1, otherwise output 0.

Intuition on non-interactivity The generation process of re-encryption key $rk_{id_i \to id_j}$ depends upon public key of user F_{id_i} without resorting to id_j secret SK_{id_j}, which means the process does not require the participation of id_j, which is non-interactivity. Error term in re-encryption ciphertext stems from raw ciphertext which is random and unrelated to Bob, which is the recipient anonymous property.

Unidirectionality Essentially, this property is mainly to ensure that user id_i and the proxy cannot decrypt user $id_j's$ ciphertexts through collusion. Intuitively, id_j is not involved in the whole process of re-encryption, so privacy information's disclosure is unlikely to occur. Actually, neither $rk_{id_i \to id_j}$ is impossible to reverse, nor get $rk_{id_j \to id_i}$ through the calculation method expect making use of $d_j's$ secret information. Therefore, the information would not contribute to the directionality.

5.1 Parameters and Correctness

When the cryptosystem is operated as specified, we have during decryption(here we focus on the re-encryption ciphertext):

$$\begin{aligned} w &= c_0 - c_1' e_{id} \\ &= b\left\lfloor \frac{q}{2} \right\rfloor + \underbrace{x - \left[y'^{\top} \mid z'^{\top} \right] e_{id}}_{error\ term} \end{aligned}$$

Lemma 4. The norm of error term is bounded by $[q\sigma\sqrt{n}m^2\alpha\omega(logn^{\frac{3}{2}})]$ w.h.p.

Proof. Letting $e_{id} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$ with $e_1, e_2 \in \mathbb{Z}^m$, the error term is

$$\begin{aligned} x - \left[y'^{\top} \mid z'^{\top} \right] e_{id} &= x - (y'^{\top} e_1 + z'^{\top} e_2) \\ &= x - [(y^{\top}\mathbf{X_1} + z^{\top}\mathbf{X}_2)e_1 + (y^{\top}\mathbf{X}_3 + z^{\top}\mathbf{X}_4)e_2] \\ &= x - y^{\top}[(\mathbf{X_1} + \mathbf{R}\mathbf{X}_2)e_1 + (\mathbf{X}_3 + \mathbf{R}\mathbf{X}_4)e_2] \end{aligned}$$

By Lemma 8 we have $\|e_{id}\| \le \sigma\sqrt{2m}$ w.h.p.

By Lemma 15 we have $\left\| y^{\top}[(\mathbf{X_1} + \mathbf{R}\mathbf{X}_2)e_1 + (\mathbf{X_3} + \mathbf{R}\mathbf{X}_4)e_2] \right\| \le \sigma m\sqrt{n}\omega(logn^{\frac{3}{2}})$. Then, by Lemma 12 the error term is bounded by

$$\begin{aligned} |\, x - y^{\top}[(\mathbf{X_1} + \mathbf{R}\mathbf{X}_2)e_1 + (\mathbf{X_3} + \mathbf{R}\mathbf{X}_4)e_2] | &\le |\, x\,| + |\, y^{\top}[(\mathbf{X_1} + \mathbf{R}\mathbf{X}_2)e_1 + (\mathbf{X}_3 + \mathbf{R}\mathbf{X}_4)e_2]\,| \\ &\le q\sigma\sqrt{n}m^2\alpha\omega(logn^{\frac{3}{2}}) \end{aligned}$$

which proves the lemma.

In order to make the system work correctly, we need to ensure that:

- the error term is less than $\frac{q}{5}$,
- *TrapGen* can operate, i.e., $m > 6n\log q$,
- σ is sufficiently large for *SampleLeft* and *SampleLeft*, i.e., $\sigma > m\omega(\sqrt{logm})$
- Regev's reduction can apply, i.e., $q > 2\sqrt{n}/\alpha$

To satisfy these requirements we set the parameters (q, m, σ, α) as follows, taking n to be the security parameter:

$$m = 6n^{1+\delta},\ q = m^4 \cdot \omega(logn^2),\ \sigma = m \cdot \omega(\sqrt{logn}),\ \alpha = [m^3\sqrt{n}\omega(logn^2)]^{-1}$$

and round up m to the nearest larger integer and q to the nearest larger prime. Here we assume that δ meets $n^{\delta} > \lceil logn \rceil = \mathcal{O}(logn)$.

Since the matrices $A_0,\ R_i (i = 1, \ldots, n)$ are random in $\mathbb{Z}^{n \times m}$ and $m > n\log q$, both matrices will have rank n with overwhelming probability. Hence, calling *SampleLeft* in algorithm *Extract* succeeds w.h.p.

6. Security Reduction

Under a selective identity attack, our construction is indistinguishable from random one, which means the challenge ciphertext is indistinguishable from a random element in the ciphertext space. This property implies both semantic security and recipient anonymity.

Theorem 3. *Under the parameters* (m,q,σ,α,n), *The PRE scheme is IND-sID-CPA secure provided that the* $(\mathbb{Z}_q,n,\Psi_\alpha)$*-LWE assumption holds.*

Proof. Our proof process is a group of games where the first game is identical to the IND-sID-CPA secure game from Definition 2. In the last game, the adversary's advantage is zero. We can prove that the order of the two games are indistinguishable for the adversary, so the adversary's advantage in the final game is zero, the advantage of the original IND-sID-CPA game is also zero.

Game 0 This is the original IND-sID-CPA game from Definition 2 between the adversary $\mathcal{A}$ against our scheme and an IND-sID-CPA challenger.

Game 1 In Game 1, we slightly change the way the challenger generates $-A_0\mathbf{R}_i$ in the public parameters. Let id^* be the identity that $\mathcal{A}$ wants to attack. The challenger in Game 1 chooses $-H(id^*)T\mathbf{G}$ at the setup phase and makes $-A_0\mathbf{R}_i$ as

$$-A_0\mathbf{R}_i \leftarrow -A_0\mathbf{R}_i - H(id^*)T\mathbf{G}$$

The remainder of the game is unchanged. We show that Game 0, Game 1 is statistically indistinguishable. Because from the adversary's views, the $-A_0\mathbf{R}_i$ is statistically close to uniform, and the result of alternative $-A_0\mathbf{R}_i - H(id^*)T\mathbf{G}$ by minus a portion $-H(id^*)T\mathbf{G}$ is also close to uniform. Hence, Game 0 and Game 1 are indistinguishable for the attacker.

Game 2 In Game 2, we change the generation mode of A_0. The challenger generates A_0 as a random matrix in $\mathbb{Z}_q^{n\times m}$, selects a new $\mathbf{G}\in\mathbb{Z}_q^{n\times m}$ with a special structure and a trapdoor $T_\mathbf{G}$ for $\Lambda_q^\perp(\mathbf{G})$ for the challenger. The construction of $-A_0\mathbf{R}_i$ remains as it is in Game 1, namely $-A_0\mathbf{R}_i \leftarrow -A_0\mathbf{R}_i - H(id^*)T\mathbf{G}$.

The challenger responds to private key queries using the trapdoor $T_\mathbf{G}$. To respond to a private key query for $id \neq id^*$, the challenger needs a short vector $e\in\Lambda_q^u(F_{id})$ where

$$\begin{aligned} F_{id} &= (A_0 \mid -A_0\mathbf{R}_i - H(id^*)T\mathbf{G} + H(id)T\mathbf{G}) \\ &= (A_0 \mid -A_0\mathbf{R}_i + (H(id) - H(id^*))T\mathbf{G}) \end{aligned}$$

By construction, therefore $H(id)-H(id^*)$ is non-singular and $T_\mathbf{G}$ is a trapdoor for $\Lambda_q^\perp(H(id)-H(id^*))T\mathbf{G})$. The challenger can respond to the private key query like this:

$$e \leftarrow SampleRight(A_0,(H(id)-H(id^*))T\mathbf{G},-\mathbf{R}_i,T_\mathbf{G},u,\sigma)$$

and sending $SK_{id}=e$ to $\mathcal{A}$. When $\sigma > \|\widetilde{T_G}\| s_R\omega(\sqrt{logm})$, the generated e is distributed closed to $D_{\Lambda_q^u(F_{id}),\sigma}$, as in Game 1. Therefore σ used in this system, as defined before, is sufficiently large to satisfy the conditions of algorithm $SampleRight$.

The challenger responds to re-encryption key queries like this:

First, the challenger obtains trapdoor of $\Lambda^\perp(A_0 \mid -A_0\mathbf{R}_i + (H(id)-H(id^*))T\mathbf{G})$ by running

$$e' \leftarrow SampleRight(A_0,(H(id)-H(id^*))T\mathbf{G},-\mathbf{R}_i,T_\mathbf{G},0,\sigma)$$

Then, using the trapdoor inversion algorithm $SamplePre([A_0 \mid -A_0\mathbf{R}_i + (H(id)-H(id^*))T\mathbf{G}$ [13] which u is each column of the A_0. We sample column-wise for the sake of each column of the A_0 and obtain a $2m$-dimensional column of the re-encryption key. After sampling m times,

we chalk up a $2m \times m$ matrix and divide it into two matrices X_{11} and X_{21} with Gaussian entries of parameter s.

$$A_0 = [A_0 \mid -A_0\mathbf{R_i} + (H(id_i) - H(id^*))T\mathbf{G}]\begin{pmatrix} X_{11} \\ X_{21} \end{pmatrix}$$

continue sampling for the cosets of the $-A_0\mathbf{R_i} + (H(id) - H(id^*))T\mathbf{G}$,

$$-A_0\mathbf{R_j} + (H(id_j) - H(id^*))T\mathbf{G} = [A_0 \mid -A_0\mathbf{R_i} + (H(id_i) - H(id^*))T\mathbf{G}]\begin{pmatrix} X_{12} \\ X_{22} \end{pmatrix}$$

the simulated re-encryption key

$$rk_{id_i \to id_j} = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$$

has the same distribution as a re-encryption key in the original scheme.

To answer the re-encryption query from id_i to id_j , that is, we apply $rk_{i \to j} = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$ generated before to re-encrypt ciphertext from $F_{id_i} = (A_0 \mid -A_0\mathbf{R_i} + (H(id_i) - H(id^*))T\mathbf{G})$ to $F_{id_j} = (A_0 \mid -A_0\mathbf{R_i} + (H(id_i) - H(id^*))T\mathbf{G})$. The re-encryption transforms

$$CT_{id_i} = (c_0 = s^\top u + x + b\left\lfloor \frac{q}{2} \right\rfloor, c_1 = s^\top F_{id_i} + [y \mid z])$$

to

$$CT_{id_j} = (c_0 = s^\top u + x + b\left\lfloor \frac{q}{2} \right\rfloor, c_1 = s^\top F_{id_j} + [y \mid z])$$

, where $F_{id_i} = (A_0 \mid -A_0\mathbf{R_i} + (H(id_i) - H(id^*))T\mathbf{G})$ and $F_{id_j} = (A_0 \mid -A_0\mathbf{R_j} + (H(id_j) - H(id^*))T\mathbf{G})$, decryption using key e_{id} for identity id.

Game 3 In Game 3, we change the way the challenger generates challenge ciphertext $CT = (c_0, c_1)$, make it $CT = (c_0^*, c_1^*)$ where (c_0^*, c_1^*) is always chosen as a random independent element in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$. Since the challenge ciphertext is a fresh random in the ciphertext space, the advantage of $\mathcal{A}$ in Game 3 is zero.

Next, we will give a reduction from LWE problem to prove that Game 2 and Game 3 are computationally indistinguishable for a PPT adversary.

Reduction from LWE Presume $\mathcal{A}$ has non-negligible advantage in distinguishing Game 2 and Game 3. We use $\mathcal{A}$ to construct an LWE algorithm $\mathcal{B}$.

An LWE problem instance is provided with a sampling oracle $\mathcal{O}$ which can be either truly random $\mathcal{O}_\$$ or a noisy pseudo-random $\mathcal{O}_s$ for some secret $s \in \mathbb{Z}_q^n$. The simulator $\mathcal{B}$ uses the adversary $\mathcal{A}$ to distinguish between two oracles as follows:

Instance $\mathcal{B}$ requests for $\mathcal{O}$ and receives a fresh pair $(u_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for each $i = 0, \ldots, m$.

Targeting $\mathcal{A}$ announces to $\mathcal{B}$ the target identity id^* that it intends to attack.

Setup $\mathcal{B}$ may construct the system's public parameters PP as follows:

- Assemble the random matrix $A_0 \in \mathbb{Z}_q^{n \times m}$ from m of the previously given LWE sample by letting the i-th column of A_0 be the n-vector u_i for all $i = 0, \ldots, m$.
- Assign the 0-th LWE sample (so far unused) to become the public random n-vector $u_0 \in \mathbb{Z}_q^n$.
- Construct the remainder of the public parameters using id^* and $\mathbf{R_i^*}$, namely $A_0\mathbf{R_i}$ and $T\mathbf{G}$ as in Game 2.

- Send $PP=(A_0,u_0,H,T\mathbf{G},A_0\mathbf{R}_i,i=1,\ldots,n)$ to $\mathcal{A}$.

Queries $\mathcal{B}$ answers all kinds of queries from $\mathcal{A}$ as in Game 2.

Challenge $\mathcal{B}$ prepares, when prompted by $\mathcal{A}$ with a message bit $b^*\in\{0,1\}$, a challenger ciphertext for the target identity id^* as follows:

- Let $v_0,\ldots,v_m$ be entries from the LWE instance and set $v^*=[v_1,\ldots,v_m]^{\perp}\in\mathbb{Z}_q^m$.
- Blind the message bit by letting $c_0^*=v_0+b^*\lceil\frac{q}{2}\rfloor$.
- Set $c_1^*=[v^*\,|\,(-\mathbf{R}_i^*)^{\perp}v^*]^{\perp}\in\mathbb{Z}_q^{2m}$.
- Send $CT^*=(c_0^*,c_1^*)$ to the adversary.

We argue that when the LWE oracle is pseudorandom, that is, $\mathcal{O}=\mathcal{O}_s$, then CT^* is distributed exactly as in Game 2. That is, first, observing that $F_{id^*}=(A_0\,|\,-A_0\mathbf{R}_i^*)$. Second, by the definition of $\mathcal{O}_s$, we know that $v^*=A_0^{\top}s+y$ for some random noise vector $y\in\mathbb{Z}_q^m$ distributed as $\bar{\Psi}_\alpha$. Therefore, c_1^* defined above satisfies

$$c_1^*=\begin{pmatrix}A_0^{\top}s+y\\(-\mathbf{R}_i^*)^{\top}A_0^{\top}s+(-\mathbf{R}_i^*)^{\top}y\end{pmatrix}=\begin{pmatrix}A_0^{\top}s+y\\(-A_0\mathbf{R}_i^*)^{\top}s+(-\mathbf{R}^*)^{\top}y\end{pmatrix}=(F_{id^*})^{\top}s+\begin{pmatrix}y\\(-\mathbf{R}^*)^{\top}y\end{pmatrix}$$

and the quantity on the right is precisely equal to the c_1 part of a valid challenge ciphertext in Game 2 except the second part add a minus sign. And also note that $v_0=u_0^{\top}+x$ for some x distributed as $\bar{\Psi}_\alpha$, and therefore c_0^* in step 2 satisfies $c_0^*=u_0^{\top}s+x+b^*\lfloor\frac{q}{2}\rceil$, just as the c_0 part of a challenge ciphertext in Game 2.

When $\mathcal{O}=\mathcal{O}_s$, we have that v_0 is uniform in $\mathbb{Z}_q$ and v^* is uniform in $\mathbb{Z}_q^m$. Therefore c_1^* as defined above is uniform and independent in $\mathbb{Z}_q^{2m}$ by the standard leftover hash lemma where the hash function is defined by the matrix $(A_0^{\top}\,|\,v^*)$ and ensures that $-A_0\mathbf{R}_i^*$ and $(\mathbf{R}^*)^{\top}v^*$ are uniform independent quantities. Consequently, the challenge ciphertext is always uniform in $\mathbb{Z}_q\times\mathbb{Z}_q^{2m}$, as in Game 3.

Guess After being allowed to make additional queries, $\mathcal{A}$ guess if it is interacting with a Game 2 or Game 3 challenger. Our simulator outputs $\mathcal{A}'s$ guess as the answer to the LWE challenge, which it is trying to solve.

We already argued that when $\mathcal{O}=\mathcal{O}_s$ the adversary's view is the same as in Game 2. When $\mathcal{O}=\mathcal{O}_\$$, the adversary's view is the same as in Game 3. Hence, $\mathcal{B}'s$ advantage in solving LWE is the same as $\mathcal{A}'s$ advantage in distinguishing Game 2 and Game 3, as required. This completes the description of algorithm $\mathcal{B}$ and completes the proof.

7. Conclusion

In this paper, we present an identity based proxy re-encryption scheme which is CPA secure in standard model. The scheme satisfies the properties of the non-interactivity, unidirectionality, anonymous and etc. The security of our scheme is based on the LWE assumption in the lattice. However, we have proved our scheme to be semantically secure in selective-ID, construction of adaptive-ID secure identity based proxy re-encryption scheme that is CCA secure in standard model is an open problem.

References

[1] Agrawal, S. and X. Boyen, "Identity-based encryption from lattices in the standard model," *Manuscript*, July,2009.

[2] Micciancio, D. and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," *EUROCRYPT 2012*, pp. 700, 2012. Article (CrossRef Link)

[3] Agrawal, Shweta and Boneh, Dan and Boyen, Xavier, "Efficient lattice (H) IBE in the standard model," *EUROCRYPT 2010*, pp. 553-572, 2010. Article (CrossRef Link)

[4] Blaze Matt, Bleumer Gerrit and Strauss, Martin, "Divertible protocols and atomic proxy cryptography," *EUROCRYPT 1998*, pp.127-144, 1998. Article (CrossRef Link)

[5] Shamir, Adi, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979. Article (CrossRef Link)

[6] Singh Kunwar, Pandu Rangan C and Banerjee AK, "Lattice based identity based proxy re-encryption scheme," *Journal of Internet Services and Information Security (JISIS)*, vol. 3, no. 3/4, pp. 38-51, 2013.

[7] Kirshanova Elena, "Proxy Re-encryption from Lattices," in *Proc. of International Workshop on Public Key Cryptography*, pp. 77-94, 2014. Article (CrossRef Link)

[8] Chu, Cheng-Kang and Tzeng, Wen-Guey, "Identity-based proxy re-encryption without random oracles," in *Proc. of International Conference on Information Security*, pp. 189-202, 2007. Article (CrossRef Link)

[9] Aono Yoshinori, Boyen Xavier, Wang Lihua and others, "Key-Private Proxy Re-encryption under LWE," in *Proc. of International Conference on Cryptology in India*, pp. 1-18, 2013. Article (CrossRef Link)

[10] Green Matthew and Ateniese Giuseppe, "Identity-based proxy re-encryption," *Applied Cryptography and Network Security*, pp. 288-306, 2007. Article (CrossRef Link)

[11] Zhang Jiang, Zhang Zhenfeng and Chen Yu, "PRE: Stronger security notions and efficient construction with non-interactive opening," *Theoretical Computer Science*, vol. 542, pp. 1-16, 2014. Article (CrossRef Link)

[12] Canetti Ran and Hohenberger Susan, "Chosen-ciphertext secure proxy re-encryption," in *Proc. of the 14th ACM conference on Computer and communications security*, pp. 185-194, 2007. Article (CrossRef Link)

[13] Gentry Craig, Peikert Chris and Vaikuntanathan Vinod, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. of the 40th annual ACM symposium on Theory of computing*, pp. 197-206, 2008. Article (CrossRef Link)

[14] Cash David, Hofheinz Dennis and Kiltz Eike, "How to Delegate a Lattice Basis," *IACR Cryptology ePrint Archive 2009*, vol. 2009, pp. 351, 2009.

[15] Micciancio Daniele and Regev Oded, "Lattice-based cryptography," *Post-quantum cryptography*, pp. 147-191, 2009. Article (CrossRef Link)

[16] Xagawa, D. K, "Cryptography with lattices," 2010.

[17] Micciancio Daniele and Goldwasser Shafi, "Complexity of Lattice Problems: A Cryptographic Perspective," *Siam Journal on Computing*, vol. 671, 2002. Article (CrossRef Link)

[18] Ajtai, M, "Generating hard instances of lattice problems," in *Proc. of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 99-108, 1996. Article (CrossRef Link)

[19] Regev Oded, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 34, 2009. Article (CrossRef Link)

[20] Cramer R. and I. Damgård, "On the amortized complexity of zero-knowledge protocols," *Advances in Cryptology-CRYPTO 2009*, pp. 177-191, 2009. Article (CrossRef Link)

[21] Micciancio Daniele and Regev Oded, "Worst-case to Average-case Reductions based on Gaussian Measures," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 267-302, 2007. Article (CrossRef Link)

Wei Yin, received the B.S. degree in Mathematics and Applied Mathematics from Huaibei Normal University, Huaibei, Anhui, China, in 2012. His research interests include public key cryptography, lattice cryptography, and provable security. He is currently a PhD candidate in Beijing University of Posts and Telecommunications.

Qiaoyan Wen, received the B.S. and M.S. degrees in Mathematics from Shaanxi Normal University, Xi'an, Shaanxi, China, in 1981 and 1984, respectively, and the PhD degree in cryptography from Xidian University, Xi'an, Shaanxi, China, in 1997. Her present research interests include coding theory, cryptography, information security, Internet security, and applied mathematics. She is a professor in Beijing University of Posts and Telecommunications.

Wenmin Li, received the B.S. and M.S. degrees in Mathematics and Applied Mathematics from Shaanxi Normal University, Xi'an, Shaanxi, China, in 2004 and 2007, respectively, and the Ph.D. degree in Cryptology from Beijing University of Posts and Telecommunications, Beijing, China, in 2012. She is currently a post-doctoral in Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include cryptography and information security.

Hua Zhang, received the B.S. degree in telecommunications engineering from the Xidian University in 1998, the M.S. degree in cryptology from Xidian University in 2005, and the PhD degree in cryptology from Beijing University of Posts and Telecommunications in 2008. Now she is an associate professor in Beijing University of Posts and Telecommunications. Her research interests include cryptography, information security and network security.

Zhengping Jin received the BS degree in Math and Applied Math, MS degree in Applied Math from Anhui Normal University in 2004 and in 2007 respectively, and the Ph.D degree in Cryptography from Beijing University of Posts and Telecommunications in 2010. Now he is a lecturer of Beijing University of Posts and Telecommunications. His research interests include cryptography, information security, internet security and applied mathematics.