Lattice-based Fuzzy Conditional Proxy Re-encryption

BaoHong Li, JieFei Xu, YanZhi Liu

School of Electronics and Information Engineering, Xi'an Jiaotong University, China
bhli@mail.xjtu.edu.cn, 857331911@qq.com, smart_lyz@stu.xjtu.edu.cn

Abstract

As an extension to conditional proxy re-encryption, fuzzy conditional proxy re-encryption allows for a proxy to re-encrypt a ciphertext only if it satisfies some *t*-out-of-*d* threshold condition. Therefore, it is more desirable in some applications where fine-grained control of the decryption delegation is required. In this paper, we construct the first lattice-based fuzzy conditional proxy re-encryption scheme as a post-quantum alternative to this primitive. In our construction, original ciphertexts and re-encryption ciphertexts have the same form, thus only one decryption algorithm is needed for both kinds of ciphertexts. We formalize its security model and prove that it is selective secure under the LWE assumption.

Keywords: Proxy re-encryption, Lattice-based cryptography, Fine-grained control, Decryption delegation

1 Introduction

1.1 Motivation

Proxy re-encryption (PRE) [1] allows a proxy, without performing decryption operation, to transform a ciphertext under Alice's public key into a ciphertext of the same message under Bob's public key. PRE turns out to be a useful primitive for delegation of decryption rights; however, it does not facilitate flexible delegation since a proxy can transform *all* of Alice's ciphertexts, without any discrimination.

To address this issue, Weng et al. [2] introduced the notion of *conditional proxy re-encryption* (CPRE), such that only ciphertexts satisfying certain condition can be transformed. Since the original CPRE in [2] can only deal with simple keyword-matching conditions, it is undesirable in some applications, such as cloud computing [3], where fine-grained decryption delegation is required. Therefore, some more expressive CPRE schemes, such as fuzzy CPRE [4] and attribute-based CPRE [5] are proposed to enable more fine-grained conditions.

To resist against quantum attacks, a substantial number of lattice-based PRE schemes (e.g., [6-8]) and one lattice-based CPRE scheme [9] have been proposed in the literature. However, probably due to the fact that lattices have far less algebraic structure, construction of more expressive CPRE from lattices remains an unsolved problem.

1.2 Contribution

In this paper, we take the first step in this direction by constructing the first lattice-based fuzzy CPRE. Our fuzzy CPRE uses *t*-out-of-*d* threshold conditions to control the decryption delegation. More concretely, each ciphertext in our fuzzy CPRE is labeled with a keyword set W. To re-encrypt a ciphertext, a re-encryption key is generated from another keyword set S with d keywords in it. The ciphertext can be re-encrypted only if W and S have at least t common keywords. Furthermore, the threshold conditions in our fuzzy CPRE are flexible, in the sense that a delegator (Alice) can choose different t and d for each delegation.

Like the fuzzy CPRE of [4], our fuzzy CPRE is single-hop unidirectional, in the sense that a ciphertext can only be delegated once and a re-encryption key can only work in one direction. However, our fuzzy CPRE is more compact, since original ciphertexts and re-encryption ciphertexts have the same form, and thus only one algorithm is needed to decrypt both kinds of ciphertexts.

We also strength the security model in [4] and prove that our fuzzy CPRE is selectively CPA-secure under the LWE assumption.

1.3 Technical Approach

We begin by recalling the basic idea in the fuzzy IBE of [10], and then explain our approach to make this idea work in our fuzzy CPRE.

In the fuzzy IBE of [10], each keyword w has a uniform matrix $\mathbf{E}_w$ as its public key and the corresponding lattice short basis is included in the master secret key. The master secret key is used to generate users' secret keys. To do so, the Shamir secret sharing scheme is used to construct l shares $(\mathbf{u}_1,\ldots, \mathbf{u}_l)$ of a public vector $\mathbf{u}$, and then for each $w = 1, \ldots, l$, a short vector $\mathbf{e}_w$ is sampled such that $\mathbf{E}_w\mathbf{e}_w = \mathbf{u}_w$. Using these short vectors as her secret key, a user can decrypt a ciphertext if it has at least t same public key matrices as in her secret key.

*Corresponding Author: BaoHong Li; E-mail: bhli@mail.xjtu.edu.cn
DOI: 10.3966/160792642019092005006

In our fuzzy CPRE, a secret key is a lattice short basis so that a set of short vectors are sampled as a re-encryption key. The dual trapdoor technique from [11] is used to reduce the secret key size. More concretely, for each user i, a uniform matrix $\mathbf{A}_i$ is generated and the corresponding lattice short basis is her secret key. The public key for each keyword w is the matrix $[\mathbf{A}_i \mid \mathbf{E}_w + \mathbf{B}]$, where $\mathbf{B}$ is a public random matrix. Accordingly, a ciphertext includes one $\mathbf{A}_i$ component and multiple $[\mathbf{E}_w + \mathbf{B}]$ components, so that the combination of the $\mathbf{A}_i$ component and any subset of $[\mathbf{E}_w + \mathbf{B}]$ components is used to perfrom the re-encryption operation.

The benefits of this approach are three-fold. First, using the basis delegation technique [12], the user i can obtain the trapdoor for the matrix $[\mathbf{A}_i \mid \mathbf{E}_w + \mathbf{B}]$, thus she can generate re-encryption keys for any keyword sets. Second, secret keys are constant size, while the master secret key in [10] is linear in the size of keyword universe. Third, since a ciphertext is split into $\mathbf{A}_i$ component and $[\mathbf{E}_w + \mathbf{B}]$ components, the combination of different users and different keyword sets need not to be considered, thus the ciphertext size does not increase.

To show the efficiency of our fuzzy CPRE, we choose the fuzzy IBE of [10] as a benchmark and make a comparison in Table 1, where n is the main security parameter, l is the size of keyword universe and $|W| \leq l$ is the size of keywords set attached to a ciphertext. The sizes for the public parameters, a secret key (or the master secret key in [10]) and a ciphertext are denoted by #PP, #SK and #CT.

Table 1. Comparison with the fuzzy IBE scheme of [10]

Schemes	#PP	#SK	#CT	m	q
fuzzy IBE of [11]	$O(l)$	$O(l)$	$O(l)$	$5n\log q$	$m^3 \cdot \log m \cdot 2^{5l}$
Our fuzzy CPRE	$O(l)$	$O(1)$	$O(\|W\|)$	$5n\log q$	$m^5 \cdot \log^2 m \cdot 2^{5l}$

Our fuzzy CPRE has shorter secret key size and ciphertext size, but slightly larger parameter q to allow the growth of noise during the re-encryption operation. Since the fuzzy CPRE is a more sophisticated primitive than the fuzzy IBE, this increase of parameter q seems unavoidable.

1.4 Related Work in Lattice World

After Xagawa [6] constructed the first PRE under lattice assumptions, a substantial number of lattice-based PRE schemes have been proposed in the lierature. A main line of these research is trying to strengthen the security of PRE. For example, Kirshanova [7] presented a CCA-1 secure PRE in the selective model under the LWE assumption. Fan and Liu [8] constructed a *tag-based* CCA secure PRE. Aono et al. [13] presented a *key-private* PRE to gain anonymity. Singh et al. [14] constructed a *master secret key secure* PRE to prevent coalition.

Some efforts have also been made to to add new features to PRE, or find new approaches to construct PRE. For example, Chandran et al. [15] construct a secure obfuscator for the PRE primitive under LWE assumption, and Ma et al. [9] constructed the first CPRE under lattice assumptions.

2 Preliminaries

Notation. We use $D_{\Lambda,\sigma}$ to denote the discrete Gaussian distribution over the lattice Λ with parameter σ. We also use $[m]$ as the abbreviation of the integer set $\{1,\ldots, m\}$. We denote vectors and matrices by bold lowercase and uppercase letters, and denote by $GS(\mathbf{A})$ the Gram-Schmidt orthogonalization of a matrix $\mathbf{A}$. Finally, we use $\|\cdot\|$ to denote the Euclidean norm of a vector or a matrix, and $|S|$ to denote the size of a set S.

2.1 Integer Lattices and Trapdoors

We consider integer lattices defined by Ajtai [16], and the trapdoor generation algorithm proposed by Alwen and Peikert [17].

Definition 2.1 (***q*-ary Lattices**). Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for some integers n, m, q and any $\mathbf{u} \in \mathbb{Z}_q^n$, define two m-dimensional full-rank lattices as:

$$\Lambda_q^{\perp}(\mathbf{A}) = \{ \mathbf{y} \in \mathbb{Z}^m : \mathbf{A}\mathbf{y} = 0 \bmod q \}$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{ \mathbf{y} \in \mathbb{Z}^m : \mathbf{A}\mathbf{y} = \mathbf{u} \bmod q \}$$

Lemma 2.2 [17]. There is a PPT algorithm **TrapGen** (n, q, m) that, on input some integers $n = n(\lambda)$, $q \geq 2$ and $m \geq 5n\log q$, outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a basis $\mathbf{T}_\mathbf{A}$ for $\Lambda_q^{\perp}(\mathbf{A})$ such that:

- $\mathbf{A}$ is statistically close to uniform.
- With overwhelming probability, $\|GS(\mathbf{T}_\mathbf{A})\| \leq O(\sqrt{n\log q})$.

2.2 Discrete Gaussians and Sampling Algorithms

Lemma 2.3. Let $\mathbf{T}_\mathbf{A}$ be any basis of $\Lambda_q^{\perp}(\mathbf{A})$ for some $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ whose columns generate $\mathbb{Z}_q^n$. Then for any vector $\mathbf{u} \in \mathbb{Z}_q^n$ and $\sigma \geq \|GS(\mathbf{T}_\mathbf{A})\| \cdot \omega(\sqrt{\log m})$.

- $\Pr[\mathbf{e} \leftarrow D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma} : \|\mathbf{e}\| > \sqrt{m}\sigma] \leq \mathrm{negl}(n)$ [18].
- For $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,\sigma}$, the marginal distribution of $\mathbf{u} = \mathbf{A}\mathbf{e} \in \mathbb{Z}_q^n$ is uniform (up to negl(n) statistical distance), and the conditional distribution of $\mathbf{e}$ given $\mathbf{u}$ is $D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$ [19].
- There is a PPT algorithm **SamplePre** $(\mathbf{A}, \mathbf{T}_\mathbf{A}, \mathbf{u}, \sigma)$ that outputs $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$ distributed statistically close to $D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$ [19].

Lemma 2.4 [11]**.** Let $q > 2$ and $m > 2n\log q$. There exists a PPT algorithm **SampleLeft** $(\mathbf{A}, \mathbf{B}, \mathbf{T}_\mathbf{A}, \mathbf{u}, \sigma)$

that, on input a full rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, a basis $\mathbf{T_A}$ of $\Lambda_q^{\perp}(\mathbf{A})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$ and a parameter $\sigma \geq \| GS(\mathbf{T_A}) \| \omega(\sqrt{\log 2m})$, outputs a sample $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{F}_1)$ distributed statistically close to $D_{\Lambda_q^{\mathbf{u}}(\mathbf{F}_1),\sigma}$ where $\mathbf{F}_1 = [\mathbf{A} \mid \mathbf{B}]$.

Lemma 2.5 [11]**.** Let $q > 2$ and $m > n$. There exists a PPT algorithm **SampleRight** ($\mathbf{A}$, $\mathbf{AR} + \mathbf{B}$, $\mathbf{T_B}$, $\mathbf{u}$, σ) that, on input two matrices $\mathbf{A}$, $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, a uniform random matrix $\mathbf{R} \in \{-1, 1\}^{m \times m}$, a basis $\mathbf{T_B}$ of $\Lambda_q^{\perp}(\mathbf{B})$, a vector $\mathbf{u} \in \mathbb{Z}_q^n$ and a parameter $\sigma \geq \| GS(\mathbf{T_B}) \| \cdot \sqrt{m} \cdot \omega(\log m)$, outputs a sample $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{F}_2)$ distributed statistically close to $D_{\Lambda_q^{\mathbf{u}}(\mathbf{F}_2),\sigma}$ where $\mathbf{F}_2 = [\mathbf{A} \mid \mathbf{AR} + \mathbf{B}]$.

2.3 Learning with Errors

Regev [20] introduced the Learning With Errors (LWE) problem, and showed it is as hard as the worst-case SIVP and GapSVP under a quantum reduction. Applebaum et al. [21] further showed that the LWE problem remains equivalently hard even if the secret vector is sampled from the noise distribution.

Definition 2.6 (**LWE**). For any real $\alpha \in (0, 1)$, $q \geq 2\sqrt{n}/\alpha$, let $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ be the group of reals [0, 1) with addition modulo 1, and let ψ_α be the distribution over $\mathbb{T}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$. Define $\bar{\psi}_\alpha$ as the discrete distribution of the random variable $\lceil qX \rfloor \bmod q$, where the random variable $X \in \mathbb{T}$ has the distribution ψ_α.

Let $n, m \geq 1$. Given a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, let $A_{\mathbf{s},\bar{\psi}_\alpha}$ be a pseudo-random distribution obtained by uniformly random sampling $\mathbf{a} \in \mathbb{Z}_q^n$, sampling $x \leftarrow \bar{\psi}_\alpha$ and outputting $(\mathbf{a}, \mathbf{a}^{\mathrm{T}}\mathbf{s} + x) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The $(\mathbb{Z}_q, n, \bar{\psi}_\alpha)$-LWE problem asks to distinguish m samples chosen according to $A_{\mathbf{s},\bar{\psi}_\alpha}$ and m samples from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Lemma 2.7 [21]. Let $q = p^e$ be a prime power. There is a deterministic polynomial time transformation that, for arbitrary $\mathbf{s} \in \mathbb{Z}_q^n$ and noise distribution χ, maps $A_{\mathbf{s},\chi}$ to $A_{\bar{\mathbf{x}},\chi}$ where $\bar{\mathbf{x}} \leftarrow \chi^n$, and maps the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ to itself.

To prove the security of our construction, we also need two lemmas presented by Agrawal et al. [11].

Lemma 2.8 [11]**.** Let $\mathbf{e} \in \mathbb{Z}^m$ and $\mathbf{x} \leftarrow \bar{\psi}_\alpha^m$. Then the quanity $|\mathbf{e}^{\mathrm{T}}\mathbf{x}|$ treated as an integer in $[0, q-1]$ satisfies

$$|\mathbf{e}^{\mathrm{T}}\mathbf{x}| \leq \| \mathbf{e} \| q\alpha\omega(\sqrt{\log m}) + \| \mathbf{e} \| \sqrt{m}/2$$

with all but negligible probability in m.

Lemma 2.9 [11]**.** Let q be prime and $m > (n+1)\log q + \omega(\log n)$. Matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{R}$ are randomly chosen from $\mathbb{Z}_q^{n \times m}$, $\mathbb{Z}_q^{n \times m}$ and $\{-1, 1\}^{m \times m}$. Then for all vectors $\mathbf{u} \in \mathbb{Z}_q^m$, the distribution ($\mathbf{A}$, $\mathbf{AR}$, $\mathbf{R}^{\mathrm{T}}\mathbf{u}$) is statistically close to the distribution ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{R}^{\mathrm{T}}\mathbf{u}$).

3 Definitions

3.1 Fuzzy CPRE

Definition 3.1 A (single-hop) fuzzy conditional proxy re-encryption over a keyword universe U consists of the following six algorithms.

- **Setup**(λ, l). On input a security parameter λ and the size of keyword universe l, outputs some public parameters PP.
- **KeyGen**(i). On input an user index i, outputs a public/secret key pair (pk_i, sk_i).
- **Enc**(pk_i, W, m). On input a public key pk_i, a keyword set $W \subseteq U$ and a message $m \in \{0, 1\}$, outputs an original ciphertext CT_W^i.
- **ReKeyGen**(pk_i, sk_i, pk_j, S, t). On input a delegator's public/secret key pair (pk_i, sk_i), a delegatee's public key pk_j, a keyword set $S \subseteq U$ and a threshold t such that $t \leq |S| \leq l$, outputs a re-encryption key $rk_{i \to j,S}$.
- **ReEnc**(CT_W^i, $rk_{i \to j,S}$). On input an original ciphertext CT_W^i and a re-encryption key $rk_{i \to j,S}$, outputs a re-encryption ciphertext CT_W^j under the public key pk_j if $|W \cap S| \geq t$, or $\perp$ otherwise.
- **Dec**(sk_i, CT_W^i). On input a secret key sk_i and a (original or re-encryption) ciphertext CT_W^i, outputs either a plaintext m or $\perp$.

3.2 Security Model

In the fuzzy CPRE of [4], since the first encryption algorithm and the re-encryption algorithm output exactly same first level ciphertexts, one security game is defined to consider these two kinds of ciphertexts, where the challenge ciphertext is generated by the first encryption algorithm. In our fuzzy CPRE, however, the encryption algorithm and the re-encryption algorithm output LWE-based ciphertexts which are different in noise levels and distributions. If we directly follow this security model, the indistinguishability of re-encryption ciphertexts can not be guaranteed. Therefore, we strength this security model by defining two security games, namely sIND-OC-CPA and sIND-RC-CPA (selective indistinguishability of original/re-encryption ciphertexts under chosen-plaintext attacks), to capture the respective indistinguishability of original ciphertexts and re-encryption ciphertexts, where the challenge ciphertext in the second game is generated by the re-encryption algorithm.

Both games are selective secure in the sense that the challenge user index and the challenge keyword set should be declared at the beginning of the games.

sIND-OC-CPA Game. Consider the following game between a challenger C and an adversary $\mathcal{A}$.

- **Init**. $\mathcal{A}$ declares an user index i^* and a keyword set W^* to be challenged upon.
- **Setup.** C generates $PP \leftarrow \text{Setup}(\lambda, l)$ and returns them to $\mathcal{A}$.
- **Query.** $\mathcal{A}$ makes the following types of queries:

(1) Corrupted key query on $i \neq i^*$. C generates $(pk_i, sk_i) \leftarrow \text{KeyGen}(i)$ and returns (pk_i, sk_i) to $\mathcal{A}$.

(2) Uncorrupted key query on i. C generates $(pk_i, sk_i) \leftarrow \text{KeyGen}(i)$ and returns pk_i to $\mathcal{A}$.

(3) Re-encryption key query on $\langle i, j, S\rangle$. C ignores the request if $i = i^*$ and $|W^* \cap S| \geq t$. If C responds with a re-encryption key to this query, $\mathcal{A}$ could change the challenge public key pk_{i^*} to pk_j, which is inconsistent with our selective security model. Otherwise, C generates the re-encryption key $rk_{i\rightarrow j,S} \leftarrow \text{ReKeyGen}(pk_i, sk_i, pk_j, S, t)$ and returns it to $\mathcal{A}$.

- **Challenge.** $\mathcal{A}$ outputs (m_0, m_1). C tosses a coin $b \in \{0, 1\}$ and returns $CT^{i^*}_{W^*} \leftarrow \text{Enc}(pk_{i^*}, W^*, m_b)$ to $\mathcal{A}$.
- **Guess.** $\mathcal{A}$ outputs a guess b' of b.

sIND-RC-CPA Game. Same as the sIND-OC-CPA game except the Challenge phase.

- **Challenge.** $\mathcal{A}$ outputs $(m_0, m_1, pk_j, sk_j, S, t)$ such that $|W^* \cap S| \geq t$. C tosses a coin $b \in \{0, 1\}$, computes $CT^{j}_{W^*} \leftarrow \text{Enc}(pk_j, W^*, m_b)$, $rk_{j\rightarrow i^*,S} \leftarrow \text{ReKeyGen}(pk_j, sk_j, pk_{i^*}, S, t)$ and returns $CT^{i^*}_{W^*} \leftarrow \text{ReEnc}(CT^{j}_{W^*}, rk_{j\rightarrow i^*,S})$ to $\mathcal{A}$.

Definition 3.2 (sIND-CPA) A fuzzy CPRE scheme is selective secure against chosen plaintext attack if the advantages of any PPT adversary in above two games are both negligible in λ, where the advantage is defined as $\Pr[b' = b] - 1/2$.

4 Our Construction

4.1 Construction

- **Setup**(λ, l). The algorithm first set the parameters n, m, q, σ, α as specified in Section 4.2. Then it chooses $(l + 1)$ uniformly random matrices $\mathbf{B}, \mathbf{E}_1, \ldots, \mathbf{E}_l \in \mathbb{Z}_q^{n\times m}$ and outputs $PP := (\mathbf{B}, \mathbf{E}_1, \ldots, \mathbf{E}_l)$.
- **KeyGen**(i). This algorithm runs $(\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}, \mathbf{T}_{\mathbf{A}_i}) \leftarrow$ **TrapGen** (n, q, m). It also picks a random vector $\mathbf{u}_i \in \mathbb{Z}_q^n$ and outputs $pk_i := (\mathbf{A}_i, \mathbf{u}_i)$, $sk_i := \mathbf{T}_{\mathbf{A}_i}$.
- **Enc**(pk_i, W, m). Given $pk_i := (\mathbf{A}_i, \mathbf{u}_i)$, this algorithm does following steps:

(1) Choose a uniformly random vector $\mathbf{s} \leftarrow \bar{\psi}_\alpha^n$.

(2) For each keyword $w \in W$, choose a uniformly random matrix $\mathbf{R}_w \in \{-1, 1\}^{m\times m}$.

(3) Sample $x \leftarrow \bar{\psi}_\alpha$, $\mathbf{x} \leftarrow \bar{\psi}_\alpha^m$, define $D := (l!)^2$ and compute:

$$\mathbf{c}_0 = \mathbf{A}_i^{\mathrm{T}}\mathbf{s} + D\mathbf{x}.$$
$$\mathbf{c}_{1,w} = (\mathbf{E}_w + \mathbf{B})^{\mathrm{T}}\mathbf{s} + D\mathbf{R}_w^{\mathrm{T}}\mathbf{x},\ w \in W.$$
$$c_2 = \mathbf{u}_i^{\mathrm{T}}\mathbf{s} + Dx + m\lfloor q/2\rfloor.$$

(4) Output $CT^i_W := (\mathbf{c}_0, \{\mathbf{c}_{1,w}\}_{w\in W}, c_2)$.

- **ReKeyGen** $((pk_i, sk_i, pk_j, S, t)$. Let $pk_i = (\mathbf{A}_i, \mathbf{u}_i)$, $sk_i = \mathbf{T}_{\mathbf{A}_i}$, $pk_j = (\mathbf{A}_j, \mathbf{u}_j)$ and $|S| = d \geq t$. This algorithm does following steps:

(1) Sample two noisy matrices $\mathbf{Y} \leftarrow \bar{\psi}_\alpha^{n\times n}$, $\mathbf{Z} \leftarrow \bar{\psi}_\alpha^{n\times m}$, a noisy vector $\mathbf{z} \leftarrow \bar{\psi}_\alpha^n$, and compute $\bar{\mathbf{A}}_j = \mathbf{Y}\mathbf{A}_j + \mathbf{Z}$, $\bar{\mathbf{u}}_j = \mathbf{Y}\mathbf{u}_j + \mathbf{z}$.

(2) For each column $\bar{\mathbf{a}}_k$ of $\bar{\mathbf{A}}_j$, $k \in [m]$, construct d shares. That is, choose n uniformly random degree $t - 1$ polynomials $p_{k,1}, \ldots, p_{k,n} \in \mathbb{Z}_q[x]$ such that $(p_{k,1}(0), \ldots, p_{k,n}(0))^{\mathrm{T}} = \bar{\mathbf{a}}_k$, and compute a share $\hat{\mathbf{a}}_{k,w} = (p_{k,1}(w), \ldots, p_{k,n}(w))$ for each $w \in S$.

(3) For each $k \in [m]$ and $w \in S$, sample a vector $\mathbf{e}_{k,w} \in \mathbb{Z}^{2m} \leftarrow \mathsf{SampleLeft}(\mathbf{A}_i, \mathbf{E}_w + \mathbf{B}, \mathbf{T}_{\mathbf{A}_i}, \hat{\mathbf{a}}_{k,w}, \sigma)$ such that $[\mathbf{A}_i | \mathbf{E}_w + \mathbf{B}]\mathbf{e}_{k,w} = \hat{\mathbf{a}}_{k,w}$.

(4) Construct d shares of $\bar{\mathbf{u}}_j - \mathbf{u}_i$ by choose n uniformly random degree $t - 1$ polynomials $g_1, \ldots, g_n \in \mathbb{Z}_q[x]$ such that $(g_1(0), \ldots, g_n(0))^{\mathrm{T}} = \bar{\mathbf{u}}_j - \mathbf{u}_i$, and compute a share $\hat{\mathbf{u}}_w = (g_1(w), \ldots, g_n(w))$ for each $w \in S$.

(5) Sample a vector $\mathbf{v}_w \in \mathbb{Z}^{2m} \leftarrow \mathsf{SampleLeft}(\mathbf{A}_i, \mathbf{E}_w + \mathbf{B}, \mathbf{T}_{\mathbf{A}_i}, \hat{\mathbf{u}}_w, \sigma)$ such that $[\mathbf{A}_i | \mathbf{E}_w + \mathbf{B}]\mathbf{v}_w = \hat{\mathbf{u}}_w$.

(6) Output $rk_{i\rightarrow j,S} := (\{\mathbf{e}_{k,w}\}_{k\in[m], w\in S}, \{\mathbf{v}_w\}_{w\in S})$.

- **ReEnc**$(CT^i_W, rk_{i\rightarrow j,S})$. If $|W \cap S| \geq t$, this algorithm does following steps:

(1) Choose a subset $S' \subseteq W \cap S$ such that $|S'| = t$.

(2) Compute $\mathbf{c}'_0 = [\mathbf{c}'_{0,1} | \ldots | \mathbf{c}'_{0,m}]^{\mathrm{T}}$, where $\mathbf{c}'_{0,k} = \sum_{w\in S'} L_w \mathbf{e}_{k,w}^{\mathrm{T}}[\mathbf{c}_0 | \mathbf{c}_{1,w}]$ and $L_w = \prod_{w'\in S', w'\neq w} \frac{-w'}{w-w'}$.

(3) Compute $c'_2 = \sum_{w\in S'} L_w \mathbf{v}_w^{\mathrm{T}}[\mathbf{c}_0 | \mathbf{c}_{1,w}] + c_2$.

(4) Output $CT^j_W := (\mathbf{c}'_0, \{\mathbf{c}_{1,w}\}_{w\in W}, c'_2)$.

- **Dec**(sk_i, CT^i_W). Given $sk_i = \mathbf{T}_{\mathbf{A}_i}$, this algorithm does following steps:

(1) Sample a vector $\mathbf{e}_i \in \mathbb{Z}^m \leftarrow \mathsf{SamplePre}(\mathbf{A}_i, \mathbf{T}_{\mathbf{A}_i}, \mathbf{u}_i, \sigma)$ such that $\mathbf{A}_i\mathbf{e}_i = \mathbf{u}_i$.

(2) Compute $r = c_2 - \mathbf{e}_i^{\mathrm{T}}\mathbf{c}_0 \bmod q$.

(3) Output 0 if $|r| < q/4$ or 1 otherwise.

4.2 Correctness and Parameters

Given an original ciphertext $CT^i_W := (\mathbf{c}_0, \{\mathbf{c}_{1,w}\}_{w\in W}, c_2)$, the algorithm ReEnc first computes:

$$\mathbf{c}'_{0,k} = \sum_{w\in S'} L_w \mathbf{e}_{k,w}^{\mathrm{T}}([\mathbf{A}_i | \mathbf{E}_w + \mathbf{B}]^{\mathrm{T}}\mathbf{s} + D\mathbf{y}_w)$$
$$= (\bar{\mathbf{a}}_k)^{\mathrm{T}}\mathbf{s} + \sum_{w\in S'} DL_w \mathbf{e}_{k,w}^{\mathrm{T}}\mathbf{y}_w$$

where $\mathbf{y}_w = (\mathbf{x}; \mathbf{R}_w{}^{\mathrm{T}}\mathbf{x})^{\mathrm{T}}$. Then it generates the re-encryption ciphertext as $CT'^{j}_{W} := (\mathbf{c}'_0, \{\mathbf{c}_{1,w}\}_{w\in W}, c'_2)$, where:

$\mathbf{c}'_0 = [\mathbf{c}'_{0,1}|\ldots|\mathbf{c}'_{0,m}]^{\mathrm{T}} = \mathbf{A}_j{}^{\mathrm{T}}(\mathbf{Y}^{\mathrm{T}}\mathbf{s}) + \mathbf{Z}^{\mathrm{T}}\mathbf{s} + (y_1;\ldots;y_m)^{\mathrm{T}}$, and $y_k = \sum_{w\in S'} DL_w \mathbf{e}_{k,w}^{\mathrm{T}}\mathbf{y}_w$ for $k \in [m]$, and

$$\begin{aligned} c'_2 &= \sum_{w\in S'} L_w \mathbf{v}_w^{\mathrm{T}}([\mathbf{A}_i \,|\, \mathbf{E}_w + \mathbf{B}]^{\mathrm{T}}\mathbf{s} + D\mathbf{y}_w) + c_2 \\ &= (\bar{\mathbf{u}}_j - \mathbf{u}_i)^{\mathrm{T}}\mathbf{s} + \sum_{w\in S'} DL_w \mathbf{v}_w^{\mathrm{T}}\mathbf{y}_w + \mathbf{u}_i{}^{\mathrm{T}}\mathbf{s} + Dx + m\lfloor q/2 \rfloor \\ &= \mathbf{u}_j{}^{\mathrm{T}}(\mathbf{Y}^{\mathrm{T}}\mathbf{s}) + \mathbf{z}^{\mathrm{T}}\mathbf{s} + \sum_{w\in S'} DL_w \mathbf{v}_w^{\mathrm{T}}\mathbf{y}_w + Dx + m\lfloor q/2 \rfloor \end{aligned}$$

If we consider $\mathbf{Y}^{\mathrm{T}}\mathbf{s}$ as the secret vector, this ciphertext has the same form as an original ciphertext, and thus the decryption algorithm first samples a short $\mathbf{e}_j$ such that $\mathbf{A}_j\mathbf{e}_j = \mathbf{u}_j$ and then computes:

$$\begin{aligned} r &= c'_2 - \mathbf{e}_j{}^{\mathrm{T}}\mathbf{c}'_0 \\ &= \mathbf{z}^{\mathrm{T}}\mathbf{s} + \sum_{w\in S'} DL_w \mathbf{v}_w^{\mathrm{T}}\mathbf{y}_w + Dx - \mathbf{e}_j{}^{\mathrm{T}}\mathbf{Z}^{\mathrm{T}}\mathbf{s} - \mathbf{e}_j{}^{\mathrm{T}}(y_1;\ldots;y_m)^{\mathrm{T}} \\ &\quad + m\lfloor q/2 \rfloor \bmod q. \end{aligned} \tag{1}$$

If the noise term $|\mathbf{z}^{\mathrm{T}}\mathbf{s} + \sum_{w\in S'} DL_w \mathbf{v}_w^{\mathrm{T}}\mathbf{y}_w + Dx - \mathbf{e}_j{}^{\mathrm{T}}\mathbf{Z}^{\mathrm{T}}\mathbf{s} - \mathbf{e}_j{}^{\mathrm{T}}(y_1;\ldots;y_m)^{\mathrm{T}}| < q/4$ with overwhelming probability, the decryption algorithm will output correct message m.

The above re-encryption ciphertext can not be further re-encrypted since secret vectors (e.g., $\mathbf{Y}^{\mathrm{T}}\mathbf{s}$ or $\mathbf{s}$) in $\mathbf{c}'_0$, c'_2 and $\mathbf{c}_{1,w}$ are different..

To ensure the correctness of our construction, we should set parameters under following constraints.

- Constraints on m due to Lemma 2.2, 2.4, 2.5 and 2.9.
- Constraints on σ due to Lemma 2.3, 2.4 and 2.5
- Constraints on q due to Lemma 2.6 and 2.7.
- The noise term in Equation (1) has magnitude less than $q/4$.

We determine these parameters by first estimating the magnitude of the noise term. Let $\mathbf{e}_{k,w}^{\mathrm{T}} = (\mathbf{e}_1{}^{\mathrm{T}}; \mathbf{e}_2{}^{\mathrm{T}})$, and thus $\mathbf{e}_{k,w}^{\mathrm{T}}\mathbf{y}_w = \mathbf{e}_1{}^{\mathrm{T}}\mathbf{x} + \mathbf{e}_2{}^{\mathrm{T}}\mathbf{R}_w{}^{\mathrm{T}}\mathbf{x} = (\mathbf{e}_1{}^{\mathrm{T}} + \mathbf{e}_2{}^{\mathrm{T}}\mathbf{R}_w{}^{\mathrm{T}})\mathbf{x}$. Since $||\mathbf{e}_1||, ||\mathbf{e}_2|| \le \sqrt{m}\sigma$ due to Lemma 2.3 and the facts of $||\mathbf{R}_w|| = \sqrt{m}$, $DL_w \le D^2$, by Lemma 2.8 we have $|y_k| = |\sum_{w\in S'} DL_w \mathbf{e}_{k,w}^{\mathrm{T}}\mathbf{y}_w| \le lD^2 m\sigma\,(q\alpha\,\omega(\sqrt{\log m}) + \sqrt{m}/2)$. By further applying the Cauchy-Schwarz inequality, we have $|\mathbf{e}_j{}^{\mathrm{T}}(y_1;\ldots;y_m)^{\mathrm{T}}| \le ||\mathbf{e}_j||\cdot||(y_1;\ldots;y_m)|| \le m\sigma|y_k|$. Since this is largest subterm in the noise term, we have

$$\begin{aligned} &|\mathbf{z}^{\mathrm{T}}\mathbf{s} + \sum_{w\in S'} L_w \mathbf{v}_w^{\mathrm{T}} D\mathbf{y}_w + Dx - \mathbf{e}_j{}^{\mathrm{T}}\mathbf{Z}^{\mathrm{T}}\mathbf{s} - \mathbf{e}_j{}^{\mathrm{T}}(y_1;\ldots;y_m)^{\mathrm{T}}| \\ &\le 5|\mathbf{e}_j{}^{\mathrm{T}}(y_1;\ldots;y_m)^{\mathrm{T}}| \\ &\le 2^{5l} m^2\sigma^2(q\alpha\omega(\sqrt{\log m}) + \sqrt{m}/2) \end{aligned}$$

where the last inequality follows from the fact that $5lD^2 \le 2^{5l}$.

Let δ be a real such that $n^{1+\delta} \ge (n+1)\log q + \omega(\log n)$. To meet the above constrains, we takes:

$$\begin{aligned} n &= \mathrm{poly}(\lambda),\ m = \lceil 5n^{1+\delta} \rceil,\ \sigma = m\omega(\log m) \\ q &= 2^{5l} m^5 \omega(\log^2 m),\ \alpha = (2^{5l} m^{4.5}\omega(\log m))^{-1} \end{aligned}$$

and round up q to the nearest larger integer such that $q = p^e$ for some integer p and prime e.

If we set $l = n^{\varepsilon}$ for some real $\varepsilon \in (0, 1/2)$, then the noise parameter $\alpha = 1/(2^{5n^{\varepsilon}} \cdot \mathrm{poly}(n))$. We get security under the hardness of $2^{O(n^{\varepsilon})}$-approximating gapSVP or SIVP on n-dimensional lattices.

4.3 Security Reduction

Lemma 4.1. If there exists a PPT adversary $\mathcal{A}$ with some advantage ξ against the sIND-OC-CPA security game, then there exists a simulator $\mathcal{B}$ that can solve the $(\mathbb{Z}_q, n, \bar{\psi}_\alpha)$-LWE problem with advantage $\xi/2$.

Proof. We construct the simulator $\mathcal{B}$ as follows.

- **Init**. $\mathcal{A}$ declares i^* and W^*.
- **Setup**. $\mathcal{B}$ requests $(m+1)$ LWE instances and rearrange them as $(\mathbf{A}_{i^*}, \mathbf{v}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, $(\mathbf{u}_{i^*}, v) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. Then it sets $pk_{i^*} := (\mathbf{A}_{i^*}, \mathbf{u}_{i^*})$ and generates public parameters as follows.

  (1) Run $(\mathbf{B} \in \mathbb{Z}_q^{n\times m}, \mathbf{T}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(n, q, m)$.

  (2) For each keyword $w \in W^*$, choose a uniformly random matrix $\mathbf{R}_w^* \in \{-1, 1\}^{m\times m}$ and set $\mathbf{E}_w = \mathbf{A}_{i^*}\mathbf{R}_w^* - \mathbf{B}$.

  (3) For each keyword $w \notin W^*$, choose a uniformly random matrix $\mathbf{R}_w^* \in \{-1, 1\}^{m\times m}$ and set $\mathbf{E}_w = \mathbf{A}_{i^*}\mathbf{R}_w^*$.

  (4) Return the public parameters $PP = (\mathbf{B}, \mathbf{E}_1, \ldots, \mathbf{E}_l)$ to $\mathcal{A}$.
- **Corrupted or uncorrupted Key query**. For any $i \ne i^*$, $\mathcal{B}$ generates $(pk_i, sk_i) \leftarrow \mathrm{KeyGen}(i)$ and returns (pk_i, sk_i) or pk_i to $\mathcal{A}$.
- **Re-encryption key query**. We divide these queries into following two cases:

  (1) If $i \ne i^*$, $\mathcal{B}$ generates $rk_{i\to j,S} \leftarrow \mathrm{ReKeyGen}(pk_i, sk_i, pk_j, S, t)$ and returns it to $\mathcal{A}$.

  (2) If $i = i^*$ and $|W^* \cap S| = s < t$, $\mathcal{B}$ simulates the re-encryption key as follows.
  - Sample two noisy matrices $\mathbf{Y} \leftarrow \bar{\psi}_\alpha^{n\times n}$, $\mathbf{Z} \leftarrow \bar{\psi}_\alpha^{n\times m}$, a noisy vector $\mathbf{z} \leftarrow \bar{\psi}_\alpha^{n}$, and compute $\bar{\mathbf{A}}_j = \mathbf{Y}\mathbf{A}_j + \mathbf{Z}$, $\bar{\mathbf{u}}_j = \mathbf{Y}\mathbf{u}_j + \mathbf{z}$.
  - For each column $\bar{\mathbf{a}}_k$ of $\bar{\mathbf{A}}_j$, $k \in [m]$, $\mathcal{B}$ generates $\{\mathbf{e}_{k,w}\}_{w\in S}$ as follows. For each $w \in W^* \cap S$, $\mathcal{B}$ samples $\mathbf{e}_{k,w} \leftarrow D_{\mathbb{Z}^m,\sigma}$ and computes a vector $\hat{\mathbf{a}}_{k,w} = [\mathbf{A}_{i^*}|\mathbf{E}_w + \mathbf{B}]\mathbf{e}_{k,w}$. It also chooses $(t - s - 1)$ random vectors from $\mathbb{Z}_q^n$. Using these $(t-1)$ vectors as shares, $\mathcal{B}$ can determine n polynomials $p_{k,1}, \ldots, p_{k,n} \in \mathbb{Z}_q[x]$ of degree $t-1$, such that $(p_{k,1}(0), \ldots, p_{k,n}(0)) = \bar{\mathbf{a}}_k$. Then, for each $w \in S/W^*$, $\mathcal{B}$ computes a share $\hat{\mathbf{a}}_{k,w} = (p_{k,1}(w), \ldots, p_{k,n}(w))$ and samples $\mathbf{e}_{k,w} \leftarrow \mathsf{SampleRight}(\mathbf{A}_{i^*}, \mathbf{E}_w + \mathbf{B}, \mathbf{T}_{\mathbf{B}}, \hat{\mathbf{a}}_{k,w}, \sigma)$ such that $[\mathbf{A}_{i^*}|\mathbf{E}_w + \mathbf{B}]\mathbf{e}_{k,w} = \hat{\mathbf{a}}_{k,w}$.
  - $\mathcal{B}$ also generates $\{\mathbf{v}_w\}_{w\in S}$ in a very similar way. That is, for each $w \in W^* \cap S$, $\mathcal{B}$ samples $\mathbf{v}_w \leftarrow D_{\mathbb{Z}^m,\sigma}$ and computes $\hat{\mathbf{u}}_w = [\mathbf{A}_{i^*}|\mathbf{E}_w + \mathbf{B}]\mathbf{v}_w$. Then it chooses

(t − s − 1) random vectors from $\mathbb{Z}_q^n$ and thus can determine n polynomials $g_1,\ldots, g_n \in \mathbb{Z}_q[x]$ of degree $t-1$, such that $(g_1(0),\ldots, g_n(0)) = \mathbf{u}_j - \mathbf{u}_{i^*}$. Then, for each $w \in S/W^*$, $\mathcal{B}$ computes a share $\hat{\mathbf{u}}_w = (g_1(w),\ldots, g_n(w))$ and samples $\mathbf{v}_w \leftarrow \mathsf{SampleRight}(\mathbf{A}_{i^*}, \mathbf{E}_w + \mathbf{B}, \mathbf{T}_\mathbf{B}, \hat{\mathbf{u}}_w, \sigma)$ such that $[\mathbf{A}_{i^*} | \mathbf{E}_w + \mathbf{B}]\mathbf{v}_w = \hat{\mathbf{u}}_w$.

- $\mathcal{B}$ returns $rk_{i^*\to j,S} := (\{\mathbf{e}_{k,w}\}_{k\in[m],\, w\in S}, \{\mathbf{v}_w\}_{w\in S})$ to $\mathcal{A}$.

– **Challenge**. When receiving (m_0, m_1) from $\mathcal{A}$, $\mathcal{B}$ simulates the challenge ciphertext $CT^{i^*}_{W^*} := (\mathbf{c}_0, \{\mathbf{c}_{1,w}\}_{w\in W}, c_2)$ as follows.

$$\mathbf{c}_0 = D\mathbf{v}.$$
$$\mathbf{c}_{1,w} = D(\mathbf{R}^*_w)^{\mathrm{T}}\mathbf{v}, \text{ for } w \in W^*.$$
$$c_2 = Dv + m_b\lfloor q/2\rfloor.$$

– **Guess.** When $\mathcal{A}$ outputs a guess b', $\mathcal{B}$ outputs "pseudo-random" if $b' = b$, or outputs "random" otherwise.

By Lemma 2.2 and the fact that $\mathbf{A}_{i^*}$ and $\mathbf{u}_{i^*}$ come from the LWE instances, the distribution of $(\mathbf{A}_{i^*}, \mathbf{u}_{i^*}, \mathbf{B})$ is statistically close to the distribution in the real game. By Lemma 2.9, all of $\mathbf{E}_w$ are statistical close to uniform. Further by Lemma 2.5, the simulated re-encryption key is statistically close to the output from the algorithm ReKeyGen. Therefore, the simulator is statistically indistinguishable from the challenger in the real game.

If the LWE problem instances are random, the probability that the adversary guesses the right b is 1/2. If the LWE problem instances are pseudo-random, the challenge cipheretext has following form:

$$\begin{aligned}\mathbf{c}_0 &= D\mathbf{v} = \mathbf{A}_{i^*}{}^{\mathrm{T}} D\,\mathbf{s}^* + D\mathbf{x},\ \mathbf{x} \leftarrow \bar{\psi}_\alpha^m\\ \mathbf{c}_{1,w} &= D(\mathbf{R}^*_w)^{\mathrm{T}}\mathbf{v}\\ &= (\mathbf{E}_w + \mathbf{B})^{\mathrm{T}} D\,\mathbf{s}^* + D(\mathbf{R}^*_w)^{\mathrm{T}}\mathbf{x}\\ c_2 &= Dv + m_b\lfloor q/2\rfloor\\ &= \mathbf{u}_{i^*}{}^{\mathrm{T}} D\,\mathbf{s}^* + Dx + m_b\lfloor q/2\rfloor,\ x \leftarrow \bar{\psi}_\alpha\end{aligned}$$

When we regard the secret vector as $\mathbf{s} = D\,\mathbf{s}^*$, this is a valid original cipheretext, and thus the simulator can solve the LWE problem with the same advantage.

Lemma 4.2. Our fuzzy PRE is sIND-RC-CPA secure provided that the $(\mathbb{Z}_q, n, \bar{\psi}_\alpha)$-LWE assumption holds.

Proof. We prove this lemma via a hybrid argument over a sequence of games as follows.

– **Game 0**: The real sIND-RC-CPA game.

– **Game 1**: Same as Game 0 except that the challenge re-encryption key is simulated by sampling $\mathbf{e}_{k,w}, \mathbf{v}_w \leftarrow D_{\mathbb{Z}^m,\sigma}$ for $k \in [m]$ and $w \in S$.

– **Game 2**: Same as Game 0 except that $\mathbf{c}'_0$ and c'_2 in the challenge re-encryption ciphertext are uniformly chosen from $\mathbb{Z}_q^m \times \mathbb{Z}_q$.

Since the adversary in Game 2 has advantage of exact zero, we only need to show that the adversary cannot distinguish between every two consecutive games with non-negligible advantage.

Game 0 and 1. In Game 0, by Lemma 2.7, the distributions of $\bar{\mathbf{A}}_{i^*} = \mathbf{Y}\mathbf{A}_{i^*} + \mathbf{Z}$ and $\bar{\mathbf{u}}_{i^*} = \mathbf{Y}\mathbf{u}_i + \mathbf{z}$ are computationally indistinguishable from the uniform. Thus the distributions of their shares $\hat{\mathbf{a}}_{k,w}$ and $\hat{\mathbf{u}}_w$ are also computationally indistinguishable from the uniform. Further by Lemma 2.4, $\mathbf{e}_{k,w}$ and $\mathbf{v}_w$ are statistically close to $D_{\Lambda_q^{\hat{\mathbf{a}}_{k,w}}(\mathbf{F}),\sigma}$ and $D_{\Lambda_q^{\hat{\mathbf{u}}_w}(\mathbf{F}),\sigma}$, respectively, where $\mathbf{F} = [\mathbf{A}_j | \mathbf{E}_w + \mathbf{B}]$. In Game 1, since both $\mathbf{e}_{k,w}$ and $\mathbf{v}_w$ are sampled from $D_{\mathbb{Z}^m,\sigma}$, by Lemma 2.3 the distributions of $\hat{\mathbf{a}}_{k,w} = \mathbf{F}\mathbf{e}_{k,w}$ and $\hat{\mathbf{u}}_w = \mathbf{F}\mathbf{v}_w$ are statistically close to the uniform, and the conditional distribution of $\mathbf{e}_{k,w}$ and $\mathbf{v}_w$ given $\hat{\mathbf{a}}_{k,w}$ and $\hat{\mathbf{u}}_w$ are also $D_{\Lambda_q^{\hat{\mathbf{a}}_{k,w}}(\mathbf{F}),\sigma}$ and $D_{\Lambda_q^{\hat{\mathbf{u}}_w}(\mathbf{F}),\sigma}$, respectively.

Game 1 and 2. In Game 1, since $\mathbf{e}_{k,w}$ and $\mathbf{v}_w$ are sampled from $D_{\mathbb{Z}^m,\sigma}$, by Lemma 2.3, both $\mathbf{c}'_{0,k} = \sum_{w\in S'} L_w \mathbf{e}_{k,w}^{\mathrm{T}}[\mathbf{c}_0 | \mathbf{c}_{1,w}]$ and $c'_2 = \sum_{w\in S'} L_w \mathbf{v}_w^{\mathrm{T}}[\mathbf{c}_0 | \mathbf{c}_{1,w}] + c_2$ are also statistically close to the uniform.

5 Conclusion

In this paper, we construct the first lattice-based fuzzy CPRE under the LWE assumption. Our fuzzy CPRE is single-hop unidirectional, and allows to re-encryption ciphertexts under some flexible t-out-of-d threshold conditions. In addition, original ciphertexts and re-encryption ciphertexts in our construction have the same form, thus only one algorithm is needed to decrypt these two kinds of ciphertexts.

Since Boolean formulas are more flexible to express re-encryption conditions, attribute-based CPRE under lattice assumptions is more desirable in some applications. We leave it as an interesting open problem.

References

[1] M. Blaze, G. Bleumer, M. Strauss, Divertible Protocols and Atomic Proxy Cryptography, *Advances in Cryptology – EUROCRYPT 1998*, Espoo, Finland, 1998, pp. 127-144.

[2] J. Weng, R. H. Deng, X. Ding, C.-K. Chu, J. Lai, Conditional Proxy Re-encryption Secure Against Chosen-Ciphertext Attack, *The 4th International Symposium on Information, Computer, and Communications Security*, Sydney, Australia, 2009, pp. 322-332.

[3] M. Beraka, J. Al-Muhtadi, Critical Comparison of Access Control Models for Cloud Computing, *Journal of Internet Technology*, Vol 16, No 3, pp. 431-442, May, 2015.

[4] L. M. Fang, J. D. Wang, C. P. Ge, Y. J. Ren, Fuzzy Conditional Proxy Re-encryption, *Science China*, Vol. 56, No. 5, pp. 1-13, May, 2013.

[5] J. Zhao, D. Feng, Z. Zhang, Attribute-based Conditional Proxy Re-Encryption with Chosen-Ciphertext Security, *IEEE Global Telecommunications Conference*, Miami, FL, USA, 2010, pp. 1-6.

[6] K. Xagawa, *Cryptography with Lattices*, Ph.D. thesis, Department Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Japan, 2010.

[7] E. Kirshanova, Proxy Re-encryption from Lattices, *International Workshop on Public Key Cryptography*, Buenos Aires, Argentina, 2014, pp. 77-94.

[8] X. Fan, F. H. Liu, *Proxy Re-Encryption and Re-Signatures from Lattices*, IACR Cryptology ePrint Archive: Report 2017/456, January, 2017.

[9] C. G. Ma, J. Y. Li, W. P. Ouyang, Lattice-based Identity-based Homomorphic Conditional Proxy Re-encryption for Secure Big Data Computing in Cloud Environment, *International Journal of Foundations of Computer Science*, Vol. 28, No. 6, pp. 645-655, September, 2017.

[10] S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, H. Wee, Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices, *International Workshop on Public Key Cryptography*, Darmstadt, Germany, 2012, pp. 280-297.

[11] S. Agrawal, D. Boneh, X. Boyen, Efficient Lattice (H)IBE in the Standard Model, *Advances in Cryptology – EUROCRYPT 2010*, French Riviera, 2010, pp. 553-572.

[12] D. Cash, D. Hofheinz, E. Kiltz, C. Peikert, Bonsai Trees, or How to Delegate a Lattice Basis, *Journal of Cryptology*, Vol. 25, No. 4, pp. 601-639, October, 2012.

[13] Y. Aono, X. Boyen, L. T. Phong, L. Wang, Key-private Proxy Re-encryption under LWE, *International Conference on Cryptology in India*, Mumbai, India, 2013, pp. 1-18.

[14] K. Singh, C. P. Rangan, A. K. Banerjee, Cryptanalysis of Unidirectional Proxy Re-encryption Scheme, *Information and Communication Technology - EurAsia Conference*, Bali, Indonesia, 2014, pp. 564-575.

[15] N. Chandran, M. Chase, F. H. Liu, R. Nishimaki, K. Xagawa, Re-encryption, Functional Re-encryption, and Multi-hop Re-encryption: A Framework for Achieving Obfuscation-Based Security and Instantiations from Lattices, *International Workshop on Public Key Cryptography*, Buenos Aires, Argentina, 2014, pp. 95-112.

[16] M. Ajtai, Generating Hard Instances of Lattice Problems (extended abstract), *The 28th Annual ACM Symposium on Theory of Computing*, Philadelphia, Pennsylvania, USA, 1996, pp. 99-108.

[17] J. Alwen, C. Peikert, Generating Shorter Bases for Hard Random Lattices, *Theory of Computing Systems*, Vol. 48, No. 3, pp. 535-553, April, 2011.

[18] D. Micciancio, O. Regev, Worst-case to Average-case Reductions Based on Gaussian Measures, *SIAM Journal on Computing*, Vol. 37, No. 1, pp. 267-302, January, 2007.

[19] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for Hard Lattices and New Cryptographic Constructions, *The 40th Annual ACM Symposium on Theory of Computing*, Victoria, British Columbia, Canada, 2008, pp. 197-206.

[20] O. Regev, On Lattice, Learning with Errors, Random Linear Codes, and Cryptography, *The 37th Annual ACM Symposium on Theory of Computing*, Baltimore, MD, USA, 2005, pp. 84-93.

[21] B. Applebaum, D. Cash, C. Peikert, A. Sahai, Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems, *Annual International Cryptology Conference*, LNCS, Vol. 5677, Santa Barbara, CA, USA, 2009, pp. 595-618.

Biographies

BaoHong Li received his Ph.D. in computer science and technology from Xi'an Jiaotong University in 2006. He is now a Lecturer at School of Electronics and Information Engineering, Xi'an Jiaotong University. He currently focuses on network security and applied cryptography.

JieFei Xu is now a Master degree candidate for computer science and technology from Xi'an Jiaotong University. Her research interests include reputation computing and applied cryptography.

YanZhi Liu is now a Master degree candidate for computer science and technology from Xi'an Jiaotong University. His research interests include Blockchain technology and applied cryptography.