# Attribute-Based Conditional Proxy Re-Encryption in the Standard Model under LWE

Xiaojian Liang, Jian Weng, Anjia Yang, Lisha Yao, Zike Jiang, and Zhenghao Wu

College of Cyber Security, Jinan University, China
{im.liangxj,cryptjweng,anjiayang,yaolishaqh,aptx1255,apxf1996}@gmail.com

**Abstract.** Attribute-based conditional proxy re-encryption (AB-CPRE) allows delegators to carry out attribute-based control on the delegation of decryption by setting policies and attribute vectors. The fine-grained control of AB-CPRE makes it suitable for a variety of applications, such as cloud storage and distributed file systems. However, all existing AB-CPRE schemes are constructed under classical number-theoretic assumptions, which are vulnerable to quantum cryptoanalysis. Therefore, we propose the first AB-CPRE scheme based on the learning with errors (LWE) assumption. Constructed from fully key-homomorphic encryption (FKHE) and key-switching techniques, our scheme is unidirectional, single-hop, and enables a polynomial-deep boolean circuit as its policy. Furthermore, we split the ciphertext into two independent parts to avoid two-level or multi-level encryption/decryption mechanisms. Taking advantage of it, we then extend our single-hop AB-CPRE into an efficient and concise multi-hop one. No matter how many transformations are performed, the re-encrypted ciphertext is in constant size, and only one encryption/decryption algorithm is needed. Both of our schemes are proved to be selective secure against chosen-plaintext attacks (CPA) in the standard model.

**Keywords:** Conditional proxy re-encryption · Learning with errors · Fine-grained control

## 1 Introduction

Proxy re-encryption (PRE) allows a semi-trusted proxy with a re-encryption key to transform a ciphertext intended for Alice (i.e. delegator) to another ciphertext intended for Bob (i.e.delegatee) without revealing the underlying plaintext [5]. PRE schemes can be classified into two types: one is single-hop, whose ciphertext can be transformed at most once, e.g., a ciphertext can be converted from Alice to Bob and cannot be further converted; the other is multi-hop, which means a ciphertext can be transformed multiple times, e.g., a ciphertext can be converted from Alice to Bob and to Carol, and so on. Based on the direction of transformation, PRE can be further categorized into bidirectional and unidirectional. In a

bidirectional scheme, a re-encryption key enables the transformation from Alice to Bob and vise versa. Whereas, in the unidirectional setting, a re-encryption key only supports the transformation from Alice to Bob. Notice that a bidirectional scheme can be built by running a unidirectional one in both directions.

PRE has found lots of applications that require delegation, but it may not be sufficient to facilitate flexible delegation. More specifically, once the proxy obtains a re-encryption key, it can re-encrypt all ciphertexts for delegator into the ciphertexts for delegatee without any discrimination. Suppose that some of Alice's ciphertexts are highly confidential, and they should remain secret from Bob. To implement the delegation control, a trusted proxy is needed. Such a trusted model makes PRE unrealistic in complex applications.

To address the above problem, conditional proxy re-encryption (CPRE) was introduced by Weng et al [32]. A CPRE is a variant of PRE supporting control on re-encryption. The ciphertext is associated with a condition, and the proxy can perform a transformation correctly only if the re-encryption key is associated with the same condition. The delegation control of CPRE makes it applicable to complex applications, such as encrypted email systems [30], online medical systems [13], distributed files systems [33] and cloud storage systems [19, 20].

An open problem left by Weng et al. is how to construct a CPRE scheme supporting expressive predicates over the condition [32]. To address this problem, two types of CPRE are proposed: one is fuzzy conditional proxy re-encryption (F-CPRE), which does not require the condition in the re-encryption key and ciphertext to exactly match [12]; the other is attribute-based conditional proxy re-encryption (AB-CPRE), which supports attribute-based control on delegation [24, 33, 34]. Accurately, AB-CPRE is a kind of CPRE with fine-grained control, in which the ciphertext is associated with an attribute vector $\mathbf{x}$ and the re-encryption key is related to a policy $f$. The proxy is able to perform a transformation if $f(\mathbf{x}) = 0$ only. However, as far as our knowledge, there only exist CPRE [23] and F-CPRE [18] based on learning with errors (LWE). In other words, there is no quantum-resistant AB-CPRE construction to date.

On the other hand, several multi-hop PRE schemes are available in the literature [3,9,15,17,22,29,31]. However, the majority of them do not capture conditional re-encryption property. To achieve delegation control, Mo et al. proposed a unidirectional multi-hop conditional proxy re-encryption [26]. Liang et al. suggested a bidirectional multi-hop identity-based conditional proxy re-encryption (IBCPRE) with constant-size ciphertexts [19]. But, how to construct a multi-hop CPRE with fine-grained control remains open.

The existing lattice-based CPRE and multi-hop IBCPRE leave us two interesting problems: AB-CPRE over lattices and multi-hop AB-CPRE with constant-size ciphertexts. Therefore, the new scheme should be secure under lattice-based assumptions, e.g., LWE, but it should also enjoy constant-size ciphertexts no matter how many transformations are performed.

### 1.1 Contribution

We first formalize the definition and security notation for unidirectional multi-hop AB-CPRE. Specially, to achieve multi-hop, we require that a ciphertext with

**Table 1.** Comparison between Ours and Existing Schemes

| Schemes | Types | Policy | Assumption | Security | Key Privacy | Standard Model | Direction | Multi-hop |
|---|---|---|---|---|---|---|---|---|
| [23] | IBCPRE | \ | LWE | CPA | ✓ | × | → | × |
| [18] | F-CPRE | Threshold | LWE | sCPA | × | ✓ | → | × |
| [34] | AB-CPRE | Access Tree | 3-QDBDH | sCCA | × | ✓ | → | × |
| [33] | AB-CPRE | Access Tree | BDH | CPA | × | × | → | × |
| [24] | AAB-CPRE | LSSS | 3-wDBDHI | RCCA | × | ✓ | → | × |
| Scheme I | AB-CPRE | Boolean Circuit | LWE | sCPA | ✓ | ✓ | → | × |
| [26] | CPRE | \ | DDH | CCA | × | × | → | ✓ |
| [19] | IBCPRE | \ | 3-wBDHI | CCA | × | ✓ | ↔ | ✓ |
| Scheme II | AB-CPRE | Boolean Circuit | LWE | sCPA | ✓ | ✓ | → | ✓ |

an attribute vector $\mathbf{x}$ for user $\alpha$ could be transformed into another ciphertext with a different attribute vector $\mathbf{y}$ for user $\beta$. Regarding security notation of AB-CPRE, we define a selective security and key privacy against chosen-plaintext attacks.

We also present two LWE-based unidirectional AB-CPRE schemes: the first one is single-hop; the second one is multi-hop. Our designs are obtained from fully key-homomorphic encryption (FKHE) and key-switching techniques. Table 1 makes a comparison between existing lattice-based CPRE, multi-hop CPRE, and ours. Our LWE-based constructions have the following features:

- The majority of PRE schemes are built with 2-level encryption/decryption mechanisms, where the second-level ciphertext allows to be transformed into the first-level one. To make CPRE more concise, we split ciphertext $CT$ into two parts: one is $ct$, the ciphertext for message; the other is $cc$, the ciphertext for attributes. As a result, only one encryption/decryption is needed. Moreover, by reconstructing $cc$, our single-hop AB-CPRE can be extended into a multi-hop one with constant-size ciphertexts.
- Combining with FKHE, the delegation policy enables any polynomial-deep boolean circuit. As a consequence, our schemes support fine-grained delegation control.
- Our schemes enjoy selective indistinguishability of re-encryption keys and encryptions against chosen-plaintext attacks (sKP-CPA, sIND-CPA) in the standard model.

### 1.2 Related Work

In 2010, Zhao et al. [34] supposed the first AB-CPRE scheme to improve the expressiveness and flexibility of the condition construction. Later, Yang et al. [33] presented a ciphertext-policy AB-CPRE, whose re-encryption key is related to a set of attributes whereas the ciphertext is associated with a policy. In 2018, Mao et al. [24] constructed the first anonymous AB-CPRE (AAB-CPRE) by linear secret sharing schemes (LSSS), which achieved replayable CCA (RCCA) security [10] in the standard model. But all stated AB-CPRE schemes were constructed under classical number-theoretic assumptions and none of them considers multi-hop case. We are thus motivated to propose AB-CPRE schemes over lattices that is secure against quantum attacks.

### 1.3 Organization

The rest of paper is organized as follows. The introductions of lattices, such as LWE, lattice trapdoor, and Gaussian sampling are presented in Section 2. The definition and security notation of universal AB-CPRE are formalized in Section 3. In Section 4, we propose our single-hop AB-CPRE scheme in the standard model and give the corresponding security proof. In Section 5, we give a multi-hop AB-CPRE scheme as an extension from our single-hop one. Finally, Section 6 concludes this paper.

## 2 Preliminaries

In this paper, we use a lower-case bold letter to denote a column vector $\mathbf{a}$, while an upper-case bold letter to denote a matrix $\mathbf{A}$. The (centered) discrete Gaussian distribution over $\mathcal{L}$ with parameter $\sigma$ is denoted $\mathcal{D}_\sigma(\mathcal{L})$. For vector $\mathbf{u}$, we let $\|\mathbf{u}\|$ denote its $\ell_2$ norm. For matrix $\mathbf{R} \in \mathbb{Z}^{k\times m}$, we denote by $\|\mathbf{R}\|$ the maximum length of column vector of $\mathbf{R}$. $\|\mathbf{R}\|_{GS} := \|\tilde{\mathbf{R}}\|$ where $\tilde{\mathbf{R}}$ is the Gram-Schmidt(GS) orthogonalization of $\mathbf{R}$, and $\|\mathbf{R}\|_2 := \sup_{\|\mathbf{e}\|=1} \|\mathbf{R}\mathbf{e}\|$. Then, we have $\|\mathbf{R}\|_{GS} \leq \|\mathbf{R}\| \leq \|\mathbf{R}\|_2 \leq \sqrt{k}\|\mathbf{R}\|$ and $\|\mathbf{R}\mathbf{S}\|_2 \leq \|\mathbf{R}\|_2 \cdot \|\mathbf{S}\|_2$. Moreover, we denote horizontal concatenation of vectors and/or matrices using a vertical bar, e.g., $[\mathbf{A}|\mathbf{B}]$, and vertical concatenation of vectors and/or matrices using a semicolon, e.g., $[\mathbf{A};\mathbf{B}]$.

### 2.1 Lattice Background

We use $m$-dimensional full-rank integers lattices $\Lambda$, which are discrete additive subgroups of $\mathbb{Z}^m$. A $q$-ary integer lattice and a "shift" integer lattice are defined as follows.

**Definition 1.** *(q-ary Lattices) Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for some positive integers $n, m, q$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$. We define the lattices as:*

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

**Definition 2.** *A noise distribution $\chi$ over $\mathbb{Z}$ is $B$-bounded, if $\Pr_{x\leftarrow\chi}[|x| \geq B] \leq 2^{-\tilde{\Omega}(n)}$.*

**Definition 3.** *(Decisional learning with errors) Given integers $n$, $q \geq 1$ , $m \geq O(n \log q)$, and a noise distribution $\chi$ over integers, the $LWE_{n,q,\chi}$ problem is to distinguish the following two distributions:*

$$(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) \text{ and } (\mathbf{A}, \mathbf{u})$$

*where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{s} \leftarrow \mathbb{Z}_p^n, \mathbf{e} \leftarrow \chi^m, \mathbf{u} \leftarrow \mathbb{Z}_q^m$ are sampled independently.*

**Theorem 1.** *( [8, 27, 28]) If there exists an efficient algorithm for deciding the $LWE_{n,q,\chi}$ problem for some $B = B(n), q/B \geq 2^{n^{\varepsilon}}, m = poly(n)$, then there is an efficient quantum algorithm for $SIVP_{\gamma}$ and a classical algorithm for $GapSVP_{\gamma}$ for $\gamma = 2^{\Omega(n^{\varepsilon})}$ in the worst case.*

**Corollary 1.** *(Hermite normal form [4]) There exists a useful transformation that reduces $LWE_{n,q,\chi}$ problem into one where the secret is chosen from its noise distributions $\chi$, which illustrates that distinguish the following two distributions is no easier than solving $LWE_{n,q,\chi}$ problem.*

$$(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e}) \text{ and } (\mathbf{A}, \mathbf{u})$$

*where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{s} \leftarrow \chi^n, \mathbf{e} \leftarrow \chi^m, \mathbf{u} \leftarrow \mathbb{Z}_q^m$ are sampled independently.*

**Corollary 2.** *( [21]) Applying standard hybrid argument, these distributions below are computational indistinguishable. Otherwise, there exists an efficient algorithm to solve $LWE_{n,q,\chi}$ problem.*

- $(\mathbf{A}, \mathbf{AX} + \mathbf{E})$ *and* $(\mathbf{A}, \mathbf{U})$*, where* $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{X} \leftarrow \chi^{m\times \ell}, \mathbf{E} \leftarrow \chi^{n\times \ell}, \mathbf{U} \leftarrow \mathbb{Z}_q^{n\times \ell}$.
- $(\mathbf{A}, \mathbf{D}, \mathbf{AX} + \mathbf{E}, \mathbf{DX} + \mathbf{E}')$ *and* $(\mathbf{A}, \mathbf{D}, \mathbf{AX} + \mathbf{E}, \mathbf{DX}' + \mathbf{E}')$ *where* $\mathbf{A}, \mathbf{D} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{X}, \mathbf{X}' \leftarrow \chi^{m\times \ell}, \mathbf{E}, \mathbf{E}' \leftarrow \chi^{n\times \ell}$
- $(\mathbf{A}, \{\mathbf{AX}_i + \mathbf{E}_i\}_{i\in[t]})$ *and* $(\mathbf{A}, \{\mathbf{U}_i\}_{i\in[t]})$*, where* $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{X}_i \leftarrow \chi^{m\times \ell}, \mathbf{E}_i \leftarrow \chi^{n\times \ell}, \mathbf{U}_i \leftarrow \mathbb{Z}_q^{n\times \ell}$ *for all* $i \in [t], t = poly(n)$.

**Lemma 1.** *( [1]) Given $q > 2$ and $m > (n+1)\log q + \omega(\log n)$, for some polynomial $k = k(n)$, choose three uniformly random matrices $\mathbf{U} \in \{-1, 1\}^{m\times k}$ , $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and $\mathbf{B} \in \mathbb{Z}_q^{n\times k}$. For all vectors $\mathbf{r} \in \mathbb{Z}_q^m$, the distributions $(\mathbf{A}, \mathbf{AU}, \mathbf{U}^T\mathbf{r})$ and $(\mathbf{A}, \mathbf{B}, \mathbf{U}^T\mathbf{r})$ are statistically indistinguishable.*

### 2.2 Trapdoor and Sampling

The following lemmas show the properties of lattice trapdoor and efficient Gaussian preimage sampling respectively.

**Definition 4.** *(Gadget matrix [6, 25] ) For integers $q \geq 2$ and $n \geq 1$, there is a special, structured matrix $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^T \in \mathbb{Z}_q^{n\times kn}$ where $k = \lceil \log(q) \rceil, \mathbf{g} = (1, 2, ..., 2^{k-1}) \in \mathbb{Z}_q^k$.*

- *The lattice $\Lambda_q^{\perp}(\mathbf{G})$ has a public known basis $\mathbf{T_G} \in \mathbb{Z}^{kn\times kn}$ with $\|\mathbf{T_G}\|_{GS} \leq \sqrt{5}$.*
- *For any $m \geq kn$, $\mathbf{G} \in \mathbb{Z}_q^{n\times kn}$ can be extended to a matrix $\mathbf{G}' \in \mathbb{Z}_q^{n\times m}$ by adding zero columns on the right of $\mathbf{G}$.*

**Lemma 2.** *( [1, 6, 11]) Given $n \geq 1$ , $q \geq 2$ and $m \geq \lceil 6n \log q \rceil$, we have the following polynomial-time algorithms:*

- *There is a randomized algorithm* $\mathbf{TrapGen}(1^n, 1^m, q)$ *that outputs a full-rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and a short basis* $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ *for* $\Lambda_q^{\perp}(\mathbf{A})$ *such that* $\mathbf{A}$ *is statistically close to uniform and* $\|\mathbf{T_A}\|_{GS} = O(\sqrt{n\log q})$*, with all but negligible probability in* $n$*.*
- *There is a deterministic algorithm* $\mathbf{ExtendRight}(\mathbf{A}, \mathbf{T}_A, \mathbf{B})$ *that given matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{n\times m}$ *and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *outputs a basis* $\mathbf{T_{A|B}}$ *of* $\Lambda_q^{\perp}(\mathbf{A}|\mathbf{B})$ *such that* $\|\mathbf{T_{A|B}}\|_{GS} = \|\mathbf{T_A}\|_{GS}$*.*
- *There is a deterministic algorithm* $\mathbf{ExtendLeft}(\mathbf{A}, \mathbf{G}, \mathbf{T_G}, \mathbf{S})$ *that given full-rank matrices* $\mathbf{A}, \mathbf{G} \in \mathbb{Z}_q^{n\times m}$ *and a basis* $\mathbf{T_G}$ *of* $\Lambda_q^{\perp}(\mathbf{G})$ *outputs a basis* $\mathbf{T_{A|AS+G}}$ *of* $\Lambda_q^{\perp}(\mathbf{A}|\mathbf{AS}+\mathbf{G})$ *such that* $\|\mathbf{T_{A|AS+G}}\|_{GS} \leq \|\mathbf{T_G}\|_{GS}\cdot(1+\|\mathbf{S}\|_2)$*.*

**Lemma 3.** *( [2, 11, 14]) Given integers* $n, q > 2$ *and* $m > n$*. Let* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and* $\mathbf{T_A}$ *be a basis for* $\Lambda_q^{\perp}(\mathbf{A})$*, for any* $\sigma \geq \|\mathbf{T_A}\|_{GS} \cdot \omega(\sqrt{\log m})$*. We have*

- *A random vector* $\mathbf{x}$ *sampled from* $D_\sigma(\Lambda_q^{\mathbf{u}}(\mathbf{A}))$ *has* $\ell_2$ *norm less than* $\sigma\sqrt{m}$ *all but with negiligible probability in* $m$*.*
- *There is a randomized algorithm* $\mathbf{SamplePre}(\mathbf{A}, \mathbf{T_A}, \mathbf{D}, \sigma)$*, which outputs a random matrix* $\mathbf{X} \in \Lambda_q^{\mathbf{D}}(\mathbf{A})$ *such that* $\mathbf{AX} = \mathbf{D}$ *from a distribution that is statistically close to* $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{D}}(\mathbf{A}))$*.*
- *There is a randomized algorithm* $\mathbf{RandBasis}(\mathbf{A}, \mathbf{T_A}, \sigma)$*, which outputs a random basis* $\mathbf{T'_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$ *sampled from a distribution that is statistically close to* $\mathcal{D}_\sigma(\Lambda_q^{\mathbf{D}}(\mathbf{A}))$*.*

### 2.3 Key Homomorphism and Vector Decomposition

Let us recall some notions, used in fully homomorphic encryption.

**Definition 5.** *For any positive integers* $\ell, d$*, we define the family of functions* $\mathcal{F}_{\ell,d} = \{f : \{0,1\}^\ell \to \{0,1\}\}$*, where* $f$ *is a boolean circuit of depth* $\leq d$*.*

**Lemma 4.** *( [6, 22]) Given positive integers* $n, q, \ell, d, m$ *where* $m \geq \lceil n\log(q)\rceil$ *and a* $B$*-bounded noise distribution* $\chi$*, for any matrices* $\mathbf{B}_1, ..., \mathbf{B}_\ell \in \mathbb{Z}_q^{n\times m}$*, any boolean circuit* $f \in \mathcal{F}_{\ell,d}$ *and any* $\mathbf{x} \in \{0,1\}^\ell$*, if*

$$\forall i \in [\ell] : \mathbf{c}_i = (x_i\mathbf{G} + \mathbf{B}_i)^T\mathbf{s} + \mathbf{e}_i$$

*where* $\mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e}_i \leftarrow \chi^m$ *for* $i \in [\ell]$*, then we have,*

- *A deterministic algorithm* $\mathbf{Eval}_{\mathrm{pk}}(f, \{\mathbf{B}_i\}_{i\in[\ell]})$ *that given a circuit* $f$ *and* $\ell$ *matrices* $\{\mathbf{B}_i\}_{i\in[\ell]}$*, outputs a matrix* $\mathbf{B}_f$*.*
- *A deterministic algorithm* $\mathbf{Eval}_{\mathrm{ct}}(f, \{(x_i, \mathbf{B}_i, \mathbf{c}_i)\}_{i\in[\ell]})$ *that given a circuit* $f$*, a vector* $\mathbf{x} \in \{0,1\}^\ell$ *,* $\ell$ *matrices* $\{\mathbf{B}_i\}_{i\in[\ell]}$ *and* $\ell$ *vectors* $\{\mathbf{c}_i\}_{i\in[\ell]}$*, outputs a vector* $\mathbf{c}_f$*, satisfying*

$$\mathbf{c}_f = (f(\mathbf{x})\mathbf{G} + \mathbf{B}_f)^T\mathbf{s} + \mathbf{e}_f$$

*where* $\mathbf{B}_f = \mathbf{Eval}_{\mathrm{pk}}(f, \{\mathbf{B}_i\}_{i\in[\ell]})$ *and* $\|\mathbf{e}_f\| \leq B\sqrt{m}(m+1)^d$ *with all but negligible probability.*

– *For all* $i \in [\ell]$*, it holds that* $\mathbf{B}_i = \mathbf{A}\mathbf{S}_i - x_i^*\mathbf{G}$ *where* $\mathbf{x}^* = \{x_i^*\}_{i\in[\ell]} \in \{0,1\}^\ell$ *and* $\mathbf{S}_i \in \{-1,1\}^{m\times m}$*. A deterministic algorithm* $\mathbf{Eval}_{\mathrm{sim}}(f, \{x_i^*, \mathbf{S}_i\}_{i\in[\ell]}, \mathbf{A})$ *that given a circuit* $f$*, a vector* $\mathbf{x}^* \in \{0,1\}^\ell$ *,* $\ell$ *matrices* $\{\mathbf{S}_i\}_{i\in[\ell]}$ *and a matrix* $\mathbf{A}$*, outputs a matrix* $\mathbf{S}_f$ *satisfying*

$$\mathbf{A}\mathbf{S}_f - f(\mathbf{x}^*)\mathbf{G} = \mathbf{B}_f$$

*where* $\mathbf{B}_f = \mathbf{Eval}_{\mathrm{pk}}(f, \{\mathbf{B}_i\}_{i\in[\ell]})$ *and* $\|\mathbf{S}_f\|_2 \leq 20\sqrt{m}(m+1)^d$ *with all but negligible probability.*

**Definition 6.** *(Vector Decomposition [3, 7]) We define the function mapping vectors to their bit representations as below:*

– *A deterministic function* $\mathbf{Bits}_q(\mathbf{v})$ *that given a vector* $\mathbf{v} \in \mathbb{Z}_q^n$*, let* $\mathbf{v}_i \in \{0,1\}^n$ *be such that* $\mathbf{v} = \sum_{i=1}^{\lceil \log q\rceil - 1} 2^i \mathbf{v}_i$*, outputs a vector* $\widetilde{\mathbf{v}} \in \{0,1\}^{n\cdot\lceil\log q\rceil}$*, where* $\tilde{\mathbf{v}} = (\mathbf{v}_0; \ldots; \mathbf{v}_{\lceil\log q\rceil-1})$*.*
– *A deterministic function* $\mathbf{Power2}_q(\mathbf{X})$ *that given a matrix* $\mathbf{X} \in \mathbb{Z}_q^{n\times m}$*, outputs a matrix* $\overline{\mathbf{X}} \in \mathbb{Z}_q^{n\lceil\log q\rceil\times m}$*, where* $\overline{\mathbf{X}} = [\mathbf{X}; 2\mathbf{X}; \ldots; 2^{\lceil\log q\rceil-1}\mathbf{X}]$*.*
– *For all positive integers* $q, n, m \in \mathbb{Z}$ *, a vector* $\mathbf{v} \in \mathbb{Z}_q^n$ *and a matrix* $\mathbf{X} \in \mathbb{Z}^{n\times m}$*, it holds that* $\mathbf{v}^T\mathbf{X} = \mathbf{Bits}_q(\mathbf{v})^T \cdot \mathbf{Power2}_q(\mathbf{X}) = \widetilde{\mathbf{v}}^T\overline{\mathbf{X}} \in \mathbb{Z}_q^{1\times m}$*.*

# 3 Model of Attribute-based CPRE

In this section, we present the formalization of unidirectional AB-CPRE and its corresponding security notation. We start with multi-hop AB-CPRE, which implies single-hop AB-CPRE.

## 3.1 Multi-hop AB-CPRE

**Definition 7.** *(Multi-hop AB-CPRE) A unidirectional multi-hop attribute-based conditional proxy re-encryption scheme comprises the following six algorithms:*

– $\mathbf{Setup}(n)$*: the setup algorithm is run by a semi-trusted agent. Given a security parameter* $n$ *as input, it outputs the public parameters* $pp$*.*
– $\mathbf{KeyGen}(pp, \alpha)$*: the key generation algorithm is run by a user in the system. Given the public parameters* $pp$*, it generates the public/private key pair* $(pk_\alpha, sk_\alpha)$ *for user* $\alpha$*.*
– $\mathbf{Enc}(pp, pk_\alpha, \boldsymbol{\mu}, \mathbf{x})$*: the encryption algorithm, takes as input the public parameters* $pp$*, a public key* $pk_\alpha$*, a plaintext* $\boldsymbol{\mu}$*, and an attribute vector* $\mathbf{x}$*. It outputs a ciphertext* $CT_\alpha$ *associated with* $\mathbf{x}$ *under public key* $pk_\alpha$*.*
– $\mathbf{Dec}(pp, sk_\alpha, CT_\alpha)$*: the decryption algorithm, takes as input the public parameters* $pp$*, a private key* $sk_\alpha$ *and a ciphertext* $CT_\alpha$ *under public key* $pk_\alpha$*. It outputs a message* $\boldsymbol{\mu}$*.*
– $\mathbf{ReKeyGen}(pp, sk_\alpha, pk_\beta, f, \mathbf{y})$*: the re-encryption key generation algorithm is run by user* $\alpha$*, takes as input the public parameters* $pp$*, the private key* $sk_\alpha$ *for user* $\alpha$*, the public key* $pk_\beta$ *for another user* $\beta$*, a control policy/function* $f$ *and an attribute vector* $\mathbf{y}$*. It outputs a re-encryption key* $rk_{\alpha, f\to\beta, \mathbf{y}}$ *associated with* $f$*.*

– **ReEnc**$(pp, CT_\alpha, rk_{\alpha,f\to\beta,\mathbf{y}})$*: the re-encryption algorithm run by the proxy, takes as input a ciphertext* $CT_\alpha$ *associated with* $\mathbf{x}$ *under a public key* $pk_\alpha$ *for user* $\alpha$*, a public key* $pk_\beta$ *for user* $\beta$ *and a re-encryption key* $rk_{\alpha,f\to\beta,\mathbf{y}}$*. It outputs a ciphertext* $CT_\beta$ *associated with* $\mathbf{y}$ *under the public key* $pk_\beta$ *when* $f(\mathbf{x}) = 0$ *holds, otherwise outputs* $\perp$*.*

*Remark.* In **Enc**, **ReKeyGen**, and **ReEnc** algorithms, attribute vector $\mathbf{y}$ or $\mathbf{x}$ may be a null vector. Specially, a ciphertext with null attribute cannot be re-encrypted. For simplification, if the attribute vector is a null vector, we will omit it, e.g., $rk_{\alpha,f\to\beta} \leftarrow \mathbf{ReKeyGen}(pp, sk_\alpha, pk_\beta, f)$.

*Correctness.* In a unidirectional multi-hop attribute-based proxy re-encryption scheme. We require the correctness for encryption and re-encryption as follows,

– For any key pair $(pk_\alpha, sk_\alpha) \leftarrow \mathbf{KeyGen}(pp, \alpha)$, any attribute vector $\mathbf{x}$ and any message $\boldsymbol{\mu}$, it holds that

$$\Pr[\mathbf{Dec}(pp, sk_\alpha, \mathbf{Enc}(pp, pk_\alpha, \boldsymbol{\mu}, \mathbf{x})) = \boldsymbol{\mu}] = 1 - negl(n).$$

– For any attribute vectors $\mathbf{y}_1, ..., \mathbf{y}_t$, any key pairs $(pk_{\beta_1}, sk_{\beta_1})...(pk_{\beta_t}, sk_{\beta_t})$, and any message $\boldsymbol{\mu}$, for all $i \in \{2, ..., t\}$, $f_{i-1}(\mathbf{y}_{i-1}) = 0$, it holds that

$$\begin{gathered} rk_{\beta_{i-1},f_{i-1}\to\beta_i,\mathbf{y}_i} \leftarrow \mathbf{ReKeyGen}(pp, sk_{\beta_{i-1}}, pk_{\beta_i}, f_{i-1}, \mathbf{y}_i), \\ CT_{\beta_i}^{(i-1)} = \mathbf{ReEnc}(pp, CT_{\beta_{i-1}}^{(i-2)}, rk_{\beta_{i-1},f_{i-1}\to\beta_i,\mathbf{y}_i}), \\ \Pr[\mathbf{Dec}(pp, sk_{\beta_i}, CT_{\beta_i}^{(i-1)}) = \boldsymbol{\mu}] = 1 - negl(n). \end{gathered}$$

where $t = poly(n), CT_{\beta_1}^{(0)} = \mathbf{Enc}(pp, pk_{\beta_1}, \boldsymbol{\mu}, \mathbf{y}_1)$.

### 3.2 Single-hop AB-CPRE

Unidirectional single-hop AB-CPRE, whose ciphertext can be transformed at most once, can be viewed as a weak concept of unidirectional multi-hop AB-CPRE. CPRE scheme does not require the attribute vector (or conditional vector) as an input to decrypt the transformed ciphertext. Thus, different from multi-hop one, single-hop AB-CPRE does not require delegator to set an attribute vector $\mathbf{y}$ in **ReKeyGen** and **ReEnc** algorithms. Particularly, in single-hop scheme, we would call the ciphertext with attributes as original ciphertext, and the ciphertext with null attribute as transfromed ciphertext.

### 3.3 Security Notation

In this section, we concentrate on formulating the universal security notation for unidirectional AB-CPRE. Before proceeding, we define the notations used in security definitions.

- **Delegation chain.** Suppose in an unidirectional AB-CPRE scheme there is a re-encryption key set $RK = \{rk_{\beta_1, f_1 \to \beta_2, \mathbf{y}_2}, ..., rk_{\beta_{t-1}, f_{t-1} \to \beta_t, \mathbf{y}_t}\}$, or $RK' = \{rk_{\beta_1, f_1 \to \beta_2, \mathbf{y}_2}, ..., rk_{\beta_{t-2}, f_{t-2} \to \beta_{t-1}, \mathbf{y}_{t-1}}, rk_{\beta_{t-1}, f_{t-1} \to \beta_t}\}$, where $t \geq 2$ and $f_i(\mathbf{y}_i) = 0$ for all $i \in \{1, ..., t-1\}$. Specially, we can learn that users $\beta_1, \beta_2, ..., \beta_t$ are able to decrypt all ciphertexts with $\mathbf{y}_1$ for user $\beta_1$. Thus, we say that there exists a delegation chain under $(\beta_1, \mathbf{y}_1)$ from user $\beta_1$ to user $\beta_t$. For convenience, we denote this delegation chain as $(\mathbf{y}_1, \beta_1, ..., \beta_t)$.
- **Uncorrupted/corrupted user.** If the private key of a user is compromised by an adversary, then we consider this user as a corrupted user. Otherwise, this user is an uncorrputed user.
- **Uncorrupted/corrupted delegation chain.** Suppose there exists a delegation chain $(\mathbf{y}_1, \beta_1, ..., \beta_t)$. If all users on the chain are uncorrputed users, then it is an uncorrputed chain. Otherwise, it is a corrupted chain, which implies at least one corrupted user could decrypt all ciphertexts with $\mathbf{y}_1$ for user $\beta_1$.

*Remark.* In single-hop AB-CPRE, the delegation chain at most contains two users, e.g., $(\mathbf{x}, \alpha, \beta)$. Whereas, in multi-hop one, the delegation chain could contain $O(n)$ users.

**sIND-CPA Game.** The selective security of AB-CPRE on ciphertext is defined through the following security game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Init Adversary $\mathcal{A}$ announces an attributes vector $\mathbf{x}^*$ before seeing public parameters $pp$.

Setup Challenger $\mathcal{C}$ runs the **Setup** algorithm to generate public parameters $pp$, and then executes **KeyGen** algorithm with a random user identity $\theta$ to get a key pair $(pk_\theta, sk_\theta)$. Finally, the challenger passes $pp$ and $pk_\theta$ to the adversary $\mathcal{A}$.

Phase 1 $\mathcal{C}$ initializes three empty collections $\Psi_u$, $\Psi_c$, and $\Psi_{rk}$. Then, $\mathcal{C}$ inserts $(pk_\theta, sk_\theta)$ into $\Psi_u$. $\mathcal{A}$ sends queries $q_1, ..., q_t$ to $\mathcal{C}$. Each query is one of the following:

1) Uncorrupted key generation query $\mathcal{O}_u(\beta)$: $\mathcal{C}$ first runs algorithm $\mathbf{KeyGen}(pp, \beta)$ to get a key pair $(pk_\beta, sk_\beta)$, and then inserts it into collection $\Psi_u$. Finally, $\mathcal{C}$ output a public key $pk_\beta$.
2) Corrupted key generation query $\mathcal{O}_c(\beta)$: $\mathcal{C}$ first executes algorithm $\mathbf{KeyGen}(pp, \beta)$ to get a key pair $(pk_\beta, sk_\beta)$, and then inserts it into collection $\Psi_c$. Finally, $\mathcal{C}$ output a key pair $(pk_\beta, sk_\beta)$.
3) Re-encryption key query $\mathcal{O}_{rk}(pk_\alpha, pk_\beta, f, \mathbf{y})$: If $\alpha = \beta$ or $pk_\alpha \notin \Psi_u \cup \Psi_c$ or $pk_\beta \notin \Psi_u \cup \Psi_c$, then $\mathcal{C}$ outputs $\perp$. $\mathcal{C}$ generates a re-encryption key $rk_{\alpha, f \to \beta, \mathbf{y}}$ by executing $\mathbf{ReKeyGen}(pp, sk_\alpha, pk_\beta, f, \mathbf{y})$. If there exists a corrupted delegation chain $(\mathbf{x}^*, \theta, ...)$ in $\Psi_{rk} \cup \{rk_{\alpha, f \to \beta, \mathbf{y}}\}$, then $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ inserts $rk_{\alpha, f \to \beta, \mathbf{y}}$ into $\Psi_{rk}$ and then outputs $rk_{\alpha, f \to \beta, \mathbf{y}}$.
4) Re-encryption query $\mathcal{O}_{re}(CT_\alpha, rk_{\alpha, f \to, \beta, \mathbf{y}})$: If $rk_{\alpha, f \to, \beta, \mathbf{y}} \in \Psi_{rk}$, then $\mathcal{C}$ outputs $\mathbf{ReEnc}(pp, CT_\alpha, rk_{\alpha, f \to \beta, \mathbf{y}})$. Otherwise, $\mathcal{C}$ outputs $\perp$.

Challenge $\mathcal{A}$ submits two equal-length messages $\boldsymbol{\mu}_0^*$ and $\boldsymbol{\mu}_1^*$. $\mathcal{C}$ flips a random coin $b \in \{0,1\}$, executes $CT^* \leftarrow \mathbf{Enc}(pp, pk_\theta, \mathbf{x}^*, \boldsymbol{\mu}_b^*)$, and returns the original ciphertext $CT^*$ to $\mathcal{A}$.

Phase 2 The same as Phase 1.

Guess $\mathcal{A}$ outputs a bit $b'$, which is a guess on $b$.

**sKP-CPA Game.** The selective security of AB-CPRE on re-encryption key is the same as sIND-CPA game, except the Challenge phase.

Challenge $\mathcal{A}$ submits an uncorrupted user's public key $pk_\beta$ and a policy $f$. If there exists a re-encryption key $rk_{\beta,f\to\theta} \in \Psi_{rk}$ or $rk_{\beta,f\to\theta,\mathbf{y}} \in \Psi_{rk}$ where $\mathbf{y}$ is an attribute vector, then $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ tosses a random coin $b \in \{0,1\}$, outputs a re-encryption key $rk_{\beta,f\to\theta}$ by executing $\mathbf{ReKeyGen}(pp, sk_\beta, pk_\theta, f)$ if $b = 1$, or returns random re-encryption key $rk^*$ in re-encryption key space if $b = 0$.

**Definition 8.** *(sIND-CPA Security) An attribute-based CPRE scheme is selective secure against chosen-plaintext attacks if for any PPT adversary $\mathcal{A}$, it holds that $Pr[b' = b] = 1/2 + negl(n)$ in sIND-CPA game, where negl is a negligible function.*

**Definition 9.** *(sKP-CPA security) An attribute-based CPRE scheme is selective key privacy under chosen-plaintext attacks if for any PPT adversary $\mathcal{A}$, it holds that $Pr[b' = b] = 1/2 + negl(n)$ in sKP-CPA game, where negl is a negligible function.*

## 4 Single-hop AB-CPRE Scheme

In this section, we propose the single-hop AB-CPRE scheme. Firstly, we introduce the core techniques and the main idea behind our scheme. Then, we present our concrete scheme, its correctness as well as security proof.

### 4.1 Technique Review

We start with a brief overview of fully key-homomorphic public-key encryption (FKHE) [6] and key switching [3], which are the core techniques of our scheme.

In [6], Boneh et al. put forward a kind of FKHE. For any boolean circuit $f : \{0,1\}^\ell \to \{0,1\}$ and its $\ell$ bits input $\mathbf{x} \in \{0,1\}^\ell$, there exist three efficient algorithms $\mathbf{Eval}_{\mathrm{pk}}$, $\mathbf{Eval}_{\mathrm{ct}}$ and $\mathbf{Eval}_{\mathrm{sim}}$ (See Lemma 4 for more details).

Applying FKHE, a KP-ABE system can be constructed. The master public key contains $\ell$ attribute matrices $\{\mathbf{B}_i\}_{i\in[\ell]}$ and two matrices $\mathbf{A}, \mathbf{D}$. The master secret key is a short basis $\mathbf{T}$ for lattice $\Lambda^\perp(\mathbf{A})$.

- For a user with policy $f$, use $\mathbf{T}$ to extract a secret key $\mathbf{R}_f$ such that $[\mathbf{A}|\mathbf{B}_f]\mathbf{R}_f = -\mathbf{D}$, where $\mathbf{B}_f = \mathbf{Eval}_{\mathrm{pk}}(f, \{\mathbf{B}_i\}_{i\in[\ell]})$.
- For a ciphertext $(\mathbf{A}^T\mathbf{s} + \mathbf{e}_{in}, \mathbf{D}^T\mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}, \{(x_i\mathbf{G} + \mathbf{B}_i)^T\mathbf{s} + \mathbf{e}_i\}_{i\in[\ell]})$ of a message $\boldsymbol{\mu}$ with an attribute vector $\mathbf{x}$, the user can execute the $\mathbf{Eval}_{\mathrm{ct}}$ to assemble a ciphertext $\mathbf{c}_f = (\mathbf{B}_f + f(\mathbf{x})\mathbf{G})^T\mathbf{s} + \mathbf{e}_f$. The user can recover the message $\boldsymbol{\mu}$ correctly by secret key $\mathbf{R}_f$ if his policy $f$ satisfies $f(\mathbf{x}) = 0$.

**First attempt**. Easily, we can construct a naive AB-CPRE scheme by FKHE. Firstly, a random matrix $\mathbf{D}$ and $\ell$ attribute matrices $\{\mathbf{B}_i\}_{i\in[\ell]}$ are chosen and shared among users. Then, each user chooses his public key $\mathbf{A}$ and the corresponding private key $\mathbf{T}$, the short basis of lattice $\Lambda^{\perp}(\mathbf{A})$. At last, if user $\alpha$ wants to delegate the decryption right with policy $f$ to user $\beta$, user $\alpha$ could use $\mathbf{T}_\alpha$ to extract the re-encryption key $\mathbf{R}_{\alpha,f\to\beta}$ such that $[\mathbf{A}_\alpha|\mathbf{B}_f]\mathbf{R}_{\alpha,f\to\beta}=\mathbf{A}_\beta$.

Although this naive scheme seems to work, there is no formal proof to show the indistinguishability under chosen plaintext attack. The FKHE system of [6] only achieves selective IND-CPA secure. In other words, in FKHE system, $\mathcal{A}$ would announce an attribute vector $\mathbf{x}^*$ in the beginning, and $\mathcal{C}$ does not need to answer the query on function $f$ such that $f(\mathbf{x}^*)=0$. But the security notation of AB-CPRE needs $\mathcal{C}$ to answer the query on $\mathcal{O}_{rk}(pk_\theta, pk_\beta, f)$. In the case that $f(\mathbf{x}^*)=0$ and $pk_\beta\in\Psi_u$, $\mathcal{C}$ cannot generate the corresponding re-encryption key by **ExtendLeft**, and then abort.

To address the constrain in the naive scheme, we have to apply the key-switching technique, which was originally used in fully homomorphic encryption [7]. Aono et al. [3] constructed an interactive PRE with key privacy using key-switching. Intuitively, we can convert it into a non-interactive one as follows,

- For user $\alpha$, the public key is a pair of LWE instance $(\mathbf{A}_\alpha, \mathbf{D}_\alpha)$ while the private key is $\mathbf{S}_\alpha$, where $\mathbf{D}_\alpha=\mathbf{R}_\alpha-\mathbf{A}_\alpha\mathbf{S}_\alpha$ and $\mathbf{R}_\alpha, \mathbf{S}_\alpha$ are sampled from error distribution.
- The re-encryption key is a matrix $\mathbf{Q}_{\alpha\to\beta}$ as below,

$$\mathbf{Q}_{\alpha\to\beta}=\begin{bmatrix}\mathbf{E}_1\mathbf{A}_\beta+\mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta+\mathbf{E}_3+\mathbf{Power2}_q(\mathbf{S}_\alpha)\\ \mathbf{0} & \mathbf{I}\end{bmatrix}.$$

  where $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$ are chosen from error distribution.
- In the transformation process, the proxy converts user $\alpha$'s ciphertext $(\mathbf{c}_{in}, \mathbf{c}_{out})$ into $(\mathbf{Bits}_q(\mathbf{c}_{in}), \mathbf{c}_{out})$ and then returns $[\mathbf{Bits}(\mathbf{c}_{in})^T|\mathbf{c}_{out}^T]\mathbf{Q}_{\alpha\to\beta}$ as transformed ciphertext ($\mathbf{Power2}_q$ and $\mathbf{Bits}_q$ are defined as Definition 6 ).

Combining key-switching technique with our naive scheme, we propose a provably-secure single-hop AB-CPRE scheme. The main idea is showed as follows,

- $\ell$ attribute matrices $\{\mathbf{B}_i\}_{i\in[\ell]}$ are chosen uniformly at random and shared among users.
- Each user chooses two matrices $\mathbf{A}, \mathbf{D}$ as their public key, and the short basis $\mathbf{T}$ for lattice $\Lambda^{\perp}(\mathbf{A})$ as their private key.
- Ciphertext of message $\boldsymbol{\mu}$ with attribute vector $\mathbf{x}$ under $pk_\alpha$ is $CT_\alpha=(ct, cc)$,

$$ct=(\mathbf{A}_\alpha^T\mathbf{s}+\mathbf{e}_{in}, \mathbf{D}_\alpha^T\mathbf{s}+\mathbf{e}_{out}+\lfloor q/2\rfloor\boldsymbol{\mu})\ ,\ \ cc=\{(x_i\mathbf{G}+\mathbf{B}_i)^T\mathbf{s}+\mathbf{e}_i\}_{i\in[\ell]}$$

  where $\mathbf{A}_\alpha, \mathbf{D}_\alpha$ is the public key of user $\alpha$ and $\mathbf{s}$ is selected uniformly at random.
- Since user $\alpha$ has the short basis $\mathbf{T}_\alpha$, only $ct$ is needed in decryption process. Whereas $cc$ only works for delegation of decryption.
- If user $\alpha$ wants to delegate the decryption right with policy $f$ to user $\beta$, then user $\alpha$ extracts a matrix $\mathbf{R}_{\alpha,f}$ with small norm such that $(\mathbf{A}_\alpha|\mathbf{B}_f)\mathbf{R}_{\alpha,f}=-\mathbf{D}_\alpha$ and returns a matrix $\mathbf{Q}_{\alpha,f\to\beta}$ as re-encryption key.

### 4.2 Construction

Before giving our AB-CPRE scheme, we list the parameters that will be used.

- $(n, q, m, \chi)$ - lattice parameters, where $m \geq \lceil 6n \log q \rceil$ , $q/4 \geq B \cdot (m+1)^{O(d)}$ and $\chi$ is a B-bounded ($B \geq \sqrt{n} \cdot \omega(\log n)$ ) distribution.
- $\ell$ - number of attributes.
- $d$ - the maximum depth of the boolean circuit.
- $\sigma$ - Gaussian parameter, where $\sigma = \omega((m+1)^{d+1}) \cdot \omega(\sqrt{\log m})$.

Our scheme works for $\ell, d, q = poly(n), k = \lceil \log q \rceil$.

- $\mathbf{Setup}(n)$: Choose $\ell$ random uniform matrices $\mathbf{B}_1, ..., \mathbf{B}_\ell \leftarrow \mathbb{Z}_q^{n \times m}$ and an error sampling algorithm $\chi$, which is a $B$-bounded distribution. Return public parameters $pp := (\{\mathbf{B}_i\}_{i \in [\ell]}, \chi)$
- $\mathbf{KeyGen}(pp, \alpha)$: Select a matrix $\mathbf{D}_\alpha \leftarrow \mathbb{Z}_q^{n \times m}$ uniformly at random and generate a pair $(\mathbf{A}_\alpha, \mathbf{T}_\alpha) \leftarrow \mathbf{TrapGen}(1^n, 1^m, q)$. Then run

$$\mathbf{R}_\alpha \leftarrow \mathbf{SamplePre}(\mathbf{A}_\alpha, \mathbf{T}_\alpha, -\mathbf{D}_\alpha, \sigma) \text{ s.t. } \mathbf{A}_\alpha \mathbf{R}_\alpha = -\mathbf{D}_\alpha.$$

  Output public key $pk_\alpha = (\mathbf{A}_\alpha, \mathbf{D}_\alpha)$ and private key $sk_\alpha = (\mathbf{T}_\alpha, \mathbf{R}_\alpha)$.
- $\mathbf{Enc}(pp, pk_\alpha, \boldsymbol{\mu}, \mathbf{x})$: Given $pp = (\{\mathbf{B}_i\}_{i \in [\ell]}, \chi)$, $pk_\alpha = (\mathbf{A}_\alpha, \mathbf{D}_\alpha)$, a plaintext $\boldsymbol{\mu} \in \{0,1\}^m$ and an attribute vector $\mathbf{x} = \{x_i\}_{i \in [\ell]}$. Choose a random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and error vectors $\mathbf{e}_{in}, \mathbf{e}_{out} \leftarrow \chi^m$. Compute $ct = (\mathbf{c}_{in}, \mathbf{c}_{out})$ as

$$\mathbf{c}_{in} = \mathbf{A}_\alpha^T \mathbf{s} + \mathbf{e}_{in}, \mathbf{c}_{out} = \mathbf{D}_\alpha^T \mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}.$$

  If $\mathbf{x}$ is none or null, then set $cc = \emptyset$. Otherwise, choose $\ell$ uniformly random matrices $\mathbf{S}_i \leftarrow \{-1, 1\}^{m \times m}$ and compute

$$cc = (\{\mathbf{c}_i = (x_i \mathbf{G} + \mathbf{B}_i)^T \mathbf{s} + \mathbf{S}_i^T \mathbf{e}_{in}\}_{i \in [\ell]}) \in \mathbb{Z}_q^{\ell m}.$$

  Output ciphertext $CT_\alpha := (ct, cc)$.
- $\mathbf{Dec}(pp, sk_\alpha, CT_\alpha)$: Parse $sk_\alpha = (\mathbf{T}_\alpha, \mathbf{R}_\alpha)$ and $CT_\alpha = (ct, cc)$. Let $ct = (\mathbf{c}_{in}, \mathbf{c}_{out})$, then compute

$$\widehat{\boldsymbol{\mu}} = \begin{bmatrix} \mathbf{c}_{in}^T & \mathbf{c}_{out}^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_\alpha \\ \mathbf{I}_{m \times m} \end{bmatrix}.$$

  For $j \in [m]$, set $\boldsymbol{\mu}_j = 1$ if $|\widehat{\boldsymbol{\mu}}_j - \lfloor q/2 \rfloor| < q/4$, otherwise set $\boldsymbol{\mu}_j = 0$. Finally, output $\boldsymbol{\mu} \in \{0,1\}^m$.
- $\mathbf{ReKeyGen}(pp, sk_\alpha, pk_\beta, f)$: Given $pp = (\{\mathbf{B}_i\}_{i \in [\ell]}, \chi)$ , $sk_\alpha = (\mathbf{T}_\alpha, \mathbf{R}_\alpha)$, $pk_\beta = (\mathbf{A}_\beta, \mathbf{D}_\beta)$ and a policy $f \in \mathcal{F}_{\ell,d}$. Let $\mathbf{B}_f = \mathbf{Eval}_{\mathrm{pk}}(f, \{\mathbf{B}_i\}_{i \in [\ell]})$ and $\mathbf{F} = (\mathbf{A}_\alpha | \mathbf{B}_f) \in \mathbb{Z}^{n \times 2m}$. To construct $\mathbf{R}_{\alpha,f}$, build the basis $\mathbf{T}_{\alpha,f}$ for $\mathbf{F}$ as $\mathbf{T}_{\alpha,f} \leftarrow \mathbf{ExtendRight}(\mathbf{A}_\alpha, \mathbf{T}_\alpha, \mathbf{B}_f)$. Then run $\mathbf{SamplePre}(\mathbf{F}, \mathbf{T}_{\alpha,f}, -\mathbf{D}_\alpha, \sigma)$ to generate $\mathbf{R}_{\alpha,f}$ such that $\mathbf{F}\mathbf{R}_{\alpha,f} = -\mathbf{D}_\alpha$ where $\mathbf{R}_{\alpha,f} \in \mathbb{Z}^{2m \times m}$. Set $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$, sample matrices $\mathbf{E}_1 \leftarrow \chi^{2km \times n}, \mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km \times m}$ and build matrix

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1 \mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1 \mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\alpha,f} \\ \mathbf{0}_{m \times m} & \mathbf{I}_{m \times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m) \times 2m}.$$

  Output $rk_{\alpha, f \to \beta} = \mathbf{Q}$ as re-encryption key.

– $\mathbf{ReEnc}(pp, rk_{\alpha,f\to\beta}, CT_\alpha)$: Parse $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$, $rk_{\alpha,f\to\beta} = \mathbf{Q}$, and $CT_\alpha = (ct, cc)$. If $f(\mathbf{x}) \neq 0$ or $cc = \emptyset$ then output $\perp$, otherwise let $ct = (\mathbf{c}_{in}, \mathbf{c}_{out}), cc = \{\mathbf{c}_i\}_{i\in[\ell]}$, set $\mathbf{c}_f = \mathbf{Eval}_{\mathrm{ct}}(f, \{(x_i, \mathbf{B}_i, \mathbf{c}_i)\}_{i\in[\ell]})$ and $\widetilde{\mathbf{c}}_{in,f} = \mathbf{Bits}_q([\mathbf{c}_{in}; \mathbf{c}_f])$,

$$(\mathbf{c}'^T_{in} | \mathbf{c}'^T_{out}) = [\widetilde{\mathbf{c}}^T_{in,f} | \mathbf{c}^T_{out}] \cdot \mathbf{Q}.$$

Output $CT_\beta = (ct' = (\mathbf{c}'_{in}, \mathbf{c}'_{out}), cc' = \emptyset)$ as transformed ciphertext.

### 4.3 Correctness

According to the parameters given at the beginning, the correctness is as follows.

*Original Ciphertext.* $(\mathbf{c}_{in}, \mathbf{c}_{out})$ is the $cc$ of ciphertext under $pk_\alpha$ as follows,

$$\mathbf{c}_{in} = \mathbf{A}^T_\alpha \mathbf{s} + \mathbf{e}_{in}, \mathbf{c}_{out} = \mathbf{D}^T_\alpha \mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}.$$

Since, $\mathbf{A}_\alpha \cdot \mathbf{R}_\alpha = -\mathbf{D}_\alpha$ where $\|\mathbf{R}_\alpha\|_2 \leq m\sigma$ with overwhelming probability. Therefore, we have

$$\begin{bmatrix} \mathbf{c}^T_{in} & \mathbf{c}^T_{out} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_\alpha \\ \mathbf{I}_{m\times m} \end{bmatrix} = \mathbf{e}^T_{in}\mathbf{R}_\alpha + \mathbf{e}^T_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}^T$$

where $\|\mathbf{e}^T_{in}\mathbf{R}_\alpha + \mathbf{e}^T_{out}\| \leq m\sqrt{m}\sigma B + \sqrt{m}B \leq B \cdot (m+1)^{O(d)} \leq q/4$ with overwhelming probability, which ensures correct decryption of $\boldsymbol{\mu}$.

*Transformed Ciphertext.* $(cc = (\mathbf{c}_{in}, \mathbf{c}_{out}), ct = (\{\mathbf{c}_i\}_{i\in[\ell]}))$ is the original ciphertext associated with attribute vector $\mathbf{x}$ under $pk_\alpha$. $rk_{\alpha,f\to\beta}$ is a re-encryption key, where $f(\mathbf{x}) = 0$. By Lemma 6 and Lemma 4, we have

$$\widetilde{\mathbf{c}}^T_{in,f} \cdot \overline{\mathbf{R}}_{\alpha,f} = (\mathbf{s}^T[\mathbf{A}_\alpha | \mathbf{B}_f] + [\mathbf{e}^T_{in} | \mathbf{e}^T_f])\mathbf{R}_{\alpha,f} = -\mathbf{s}^T\mathbf{D}_\alpha + [\mathbf{e}^T_{in} | \mathbf{e}^T_f]\mathbf{R}_{\alpha,f}$$

where $\mathbf{R}_{\alpha,f} \leq \sqrt{2}m\sigma$ and $\|\mathbf{e}_f\| \leq B\sqrt{m}(m+1)^d$ with overwhelming probability. Then, the transformed ciphertext is computed as follows,

$$\begin{aligned}
[\mathbf{c}'^T_{in} | \mathbf{c}'^T_{out}] &= [\widetilde{\mathbf{c}}^T_{in,f} | \mathbf{c}^T_{out}] \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\alpha,f} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\
&= [\widetilde{\mathbf{c}}^T_{in,f}(\mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2) \,|\, \widetilde{\mathbf{c}}^T_{in,f}(\mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3) + [\mathbf{e}^T_{in} | \mathbf{e}^T_f]\mathbf{R}_{\alpha,f} + \mathbf{e}^T_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}]
\end{aligned}$$

where $\mathbf{A}_\beta$ and $\mathbf{D}_\beta$ is public key of user $\beta$, $\|\mathbf{E}_1\| \leq \sqrt{2km}B$, $\|\mathbf{E}_2\| \leq \sqrt{2km}B$ and $\|\mathbf{E}_3\| \leq \sqrt{2km}B$ with overwhelming probability. Therefore, we have

$$[\mathbf{c}'^T_{in} | \mathbf{c}'^T_{out}] \cdot \begin{bmatrix} \mathbf{R}_\beta \\ \mathbf{I} \end{bmatrix} = \widetilde{\mathbf{c}}^T_{in,f}\mathbf{E}_2\mathbf{R}_\beta + \widetilde{\mathbf{c}}^T_{in,f}\mathbf{E}_3 + [\mathbf{e}^T_{in} | \mathbf{e}^T_f]\mathbf{R}_{\alpha,f} + \mathbf{e}^T_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}$$

where $\|\widetilde{\mathbf{c}}^T_{in,f}\mathbf{E}_2\mathbf{R}_\beta + \widetilde{\mathbf{c}}^T_{in,f}\mathbf{E}_3 + [\mathbf{e}^T_{in} | \mathbf{e}^T_f]\mathbf{R}_{\alpha,f} + \mathbf{e}^T_{out}\| \leq 2km^2\sqrt{m}\sigma B + 2km\sqrt{m}B + 2m\sqrt{m}(m+1)^d\sigma B + \sqrt{m}B \leq B(m+1)^{O(d)} \leq q/4$ with overwhelming probability, which means that decryption of $\boldsymbol{\mu}$ is correct.

### 4.4 Security Proof

In this section, we show that our AB-CPRE scheme is sIND-CPA secure and sKP-CPA secure in standard model.

**Theorem 2.** *Our single-hop AB-CPRE scheme is sIND-CPA secure and sKP-CPA secure in the standard model under $LWE_{n,q,\chi}$ assumption.*

The full proof can be found in Appendix A. Here, we outline our proof sketch only. Our security proof employs proof idea from [1,6]. We build a challenger $\mathcal{C}$, who solves $LWE_{n,q,\chi}$ problem by invocating a PPT adversary $\mathcal{A}$.

Given a random matrix $[\mathbf{A}_\theta|\mathbf{D}_\theta]$, $\mathcal{C}$ will be given a uniform vector $\mathbf{u}$ or an LWE instance $[\mathbf{A}_\theta|\mathbf{D}_\theta]^T\mathbf{s}+\mathbf{e}$, where $\mathbf{e}$ is sampled from error distribution $\chi$. Then $\mathcal{A}$ announces a challenge attribute vector $\mathbf{x}^* \in \{0,1\}^\ell$ before $\mathcal{C}$ selects the public parameters and the specific public key. After receiving $\mathbf{x}^*$, $\mathcal{C}$ generates $\ell$ matrices $\{\mathbf{S}_i^*\}_{i\in[\ell]}$ with small norm uniformly at random, computes $\{\mathbf{B}_i = \mathbf{A}_\theta\mathbf{S}_i^* - x_i^*\mathbf{G}\}_{i\in[\ell]}$, sets $(\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$ as the public parameters $pp$ and sets $(\mathbf{A}_\theta, \mathbf{D}_\theta)$ as the specific public key $pk_\theta$. When adversary $\mathcal{A}$ makes a query on $\mathcal{O}_{rk}(pk_\theta, pk_\beta, f)$ such that $f(\mathbf{x}^*) \neq 0$, challenger $\mathcal{C}$ would check whether there exists a corrupted delegation chain $(\mathbf{x}^*, \theta, ...)$. If not, $\mathcal{C}$ executes $\mathbf{Eval}_{\text{sim}}$, defined as Lemma 4, produces a short basis $\mathbf{T}_{\theta,f}$ for lattice $\Lambda^\perp(\mathbf{A}_\theta|\mathbf{B}_f)$ by **ExtendLeft** and then compute a re-encryption key $rk_{\theta,f\to\beta} = \mathbf{Q}$. In Challenge phase, challenger $\mathcal{C}$ assembles a challenge ciphertext by LWE instance $[\mathbf{A}_\theta|\mathbf{D}_\theta]^T\mathbf{s}+\mathbf{e}$ or a uniform vector $\mathbf{u}$. Finally, challenger $\mathcal{C}$ outputs adversary $\mathcal{A}$'s answer as result.

However, adversary $\mathcal{A}$ may make a query on $\mathcal{O}_{rk}(pk_\theta, pk_\beta, f)$ where $f(\mathbf{x}^*) = 0$ and $pk_\beta \in \Psi_u$. In this case, $\mathbf{B}_f = \mathbf{A}_\theta\mathbf{S}_f$, challenger $\mathcal{C}$ cannot generate the corresponding short basis $\mathbf{T}_{\theta,f}$ by **ExtendLeft**, which will make $\mathcal{C}$ abort.

To fix such a problem, we have to use key-switching technique to avoid to generate $\mathbf{T}_{\theta,f}$, where $f(\mathbf{x}^*) = 0$. By LWE assumption, $\mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$ is computational indistinguishable from uniform matrix $\mathbf{M}$. As a result, we will sample a random $\mathbf{M}$ instead of computing $\mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$ when asking for $rk_{\theta,f\to\beta}$, $f(\mathbf{x}^*) = 0$ and $pk_\beta \in \Psi_u$.

## 5 Extension: Multi-hop AB-CPRE Scheme

In this section, we construct a multi-hop AB-CPRE scheme from the single-hop scheme in Section 4.

Let us show transformed ciphertext $CT_\beta = (ct, cc = \emptyset)$ in single-hop AB-CPRE, detailedly,

$$\begin{aligned} ct^T =& [\widetilde{\mathbf{c}}_{in,f}^T | \mathbf{c}_{out}] \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\alpha,f} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ \approx& (\widetilde{\mathbf{c}}_{in,f}^T\mathbf{E}_1) \cdot \left[\mathbf{A}_\beta + error \,|\, \mathbf{D}_\beta + error'\right] + \left[\mathbf{0} \,|\, \lfloor q/2 \rfloor \boldsymbol{\mu}\right] \in \mathbb{Z}_q^{n\times 2m}. \end{aligned}$$

**Method 1** Obviously, $ct$ is in the form of dual Regev's ciphertext [14]. Thus, we can apply key-switching to generate a re-encryption key $rk_{\beta\to\pi}$ from user

$\beta$ to user $\pi$ (mentioned in Section 4.1). However, in such way, once the proxy obtains a re-encryption key $rk_{\beta\to\pi}$, the proxy could transform all ciphertext of user $\beta$ to user $\pi$ without any discrimination.

**Method 2** Compared to original ciphertext, transformed ciphertext does not contain any $cc = \{(x_i\mathbf{G} + \mathbf{B}_i)^T\mathbf{s} + \mathbf{e}_i\}_{i\in[\ell]}$, which plays an important role in delegation. Thus, we can make subtle change in **ReKeyGen** algorithm to achieve multi-hop capacity. **ReKeyGen** would return a re-encryption key in single-hop AB-CPRE together with an extra matrix,

$$\mathbf{P} = [\mathbf{E}_1(y_1\mathbf{G} + \mathbf{B}_1) + \mathbf{E}_{B_1}|...|\mathbf{E}_1(y_\ell\mathbf{G} + \mathbf{B}_\ell) + \mathbf{E}_{B_\ell}]$$

where $\mathbf{y}$ is the attribute vector set by delegator, $\mathbf{E}_1$ is the same as in **ReKeyGen**, and the elements of $\mathbf{E}_{B_i}$ are chosen from error distribution $\chi$. With matrix $\mathbf{P}$, the proxy could compute the new $cc$ for the transformed ciphertext

$$\begin{aligned} cc^T &= [\mathbf{c}_1; ...; \mathbf{c}_\ell]^T = \widetilde{\mathbf{c}}_{in,f}^T\mathbf{P} \\ &= \widetilde{\mathbf{c}}_{in,f}^T \cdot [\mathbf{E}_1(y_1\mathbf{G} + \mathbf{B}_1) + \mathbf{E}_{B_1}|...|\mathbf{E}_1(y_\ell\mathbf{G} + \mathbf{B}_\ell) + \mathbf{E}_{B_\ell}]. \end{aligned}$$

Therefore, the transformed ciphertext $(ct, cc)$ would be associated with a new attribute vector $\mathbf{y}$ set by delegator.

### 5.1 Construction

The parameters are the same as in section 4, and our scheme works for $\ell, d, q = poly(n)$, $m \geq \lceil 6n\log q\rceil$, $q/4 \geq B\cdot(m+1)^{O(d)}$, $\sigma = \omega((m+1)^{d+1})\cdot\omega\sqrt{\log m}$, $k = \lceil\log q\rceil$.

- **Setup**$(n)$: the same as **Setup**$(n)$ in Section 4.
- **Enc**$(pp, pk_\alpha, \boldsymbol{\mu}, \mathbf{x})$: the same as **Enc**$(pp, pk_\alpha, \boldsymbol{\mu}, \mathbf{x})$ in Section 4.
- **Dec**$(pp, sk_\alpha, CT_\alpha)$: the same as **Dec**$(pp, sk_\alpha, CT_\alpha)$ in Section 4.
- **ReKeyGen**$(pp, sk_\alpha, pk_\beta, f, \mathbf{y})$: Parse $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$ , $sk_\alpha = (\mathbf{T}_\alpha, \mathbf{R}_\alpha)$, $pk_\beta = (\mathbf{A}_\beta, \mathbf{D}_\beta)$, a policy $f \in \mathcal{F}_{\ell,d}$ and an attribute vector $\mathbf{y} = \{y_i\}_{i\in[\ell]}$. Let $\mathbf{B}_f = Eval_{\text{pk}}(f, \{\mathbf{B}_i\}_{i\in[\ell]})$ and $\mathbf{F} = (\mathbf{A}_\alpha|\mathbf{B}_f) \in \mathbb{Z}^{n\times 2m}$. To construct $\mathbf{R}_{\alpha,f}$, build the basis $\mathbf{T}_{\alpha,f}$ for $\mathbf{F}$ as $\mathbf{T}_{\alpha,f} \leftarrow ExtendRight(\mathbf{A}_\alpha, \mathbf{T}_\alpha, \mathbf{B}_f)$. Then run $\mathbf{R}_{\alpha,f} \leftarrow SamplePre(\mathbf{F}, \mathbf{T}_{\alpha,f}, -\mathbf{D}_\alpha, \sigma)$ s.t. $\mathbf{F}\mathbf{R}_{\alpha,f} = -\mathbf{D}_\alpha$ where $\mathbf{R}_{\alpha,f} \in \mathbb{Z}^{2m\times m}$. Set $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$, sample matrices $\mathbf{E}_1 \leftarrow \chi^{2km\times n}, \mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km\times m}$ and build matrix

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\alpha,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

If $\mathbf{y}$ is none or null, then set $\mathbf{P}$ as a null matirx. Otherwise, samples $\ell$ matrices $\mathbf{E}_{B_i}$ from error distribution $\chi^{2km\times m}$ and compute,

$$\mathbf{P} = [(\mathbf{E}_1(y_i\mathbf{G} + \mathbf{B}_1) + \mathbf{E}_{B_1}) \mid ... \mid (\mathbf{E}_1(y_i\mathbf{G} + \mathbf{B}_\ell) + \mathbf{E}_{B_\ell})] \in \mathbb{Z}_q^{2km\times\ell m}.$$

Output $rk_{\alpha,f\to\beta,\mathbf{y}} = (\mathbf{Q}, \mathbf{P})$ as re-encryption key.

– $\mathbf{ReEnc}(pp, rk_{\alpha,f\to\beta,\mathbf{y}}, CT_\alpha)$: Parse $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$, $rk_{\alpha,f\to\beta} = (\mathbf{Q}, \mathbf{P})$, and $CT_\alpha = (ct, cc)$. If $f(\mathbf{x}) \neq 0$ or $cc = \emptyset$ then $\perp$, otherwise let $ct = (\mathbf{c}_{in}, \mathbf{c}_{out}), cc = \{\mathbf{c}_i\}_{i\in[\ell]}$, set $\mathbf{c}_f = \mathbf{Eval}_{\mathrm{ct}}(f, \{(x_i, \mathbf{B}_i, \mathbf{c}_i)\}_{i\in[\ell]})$ and $\widetilde{\mathbf{c}}_{in,f} = \mathbf{Bits}_q([\mathbf{c}_{in}; \mathbf{c}_f])$, then compute

$$(\mathbf{c}'^T_{in} | \mathbf{c}'^T_{out}) = [\widetilde{\mathbf{c}}^T_{in,f} | \mathbf{c}^T_{out}] \cdot \mathbf{Q}.$$

If $\mathbf{P}$ is a null matrix, then set $cc' = \emptyset$. Otherwise, compute

$$[\mathbf{c}'_1; ...; \mathbf{c}'_\ell]^T = \widetilde{\mathbf{c}}^T_{in,f} \cdot \mathbf{P}.$$

and then set $cc' = \{\mathbf{c}'_i\}_{i\in[\ell]}$. Output $CT_\beta = (ct' = (\mathbf{c}'_{in}, \mathbf{c}'_{out}), cc')$ as transformed ciphertext.

### 5.2 Correctness and Security Proof

**Theorem 3.** *Our multi-hop scheme supports $O(n)$ times transformations.*

Suppose $t = O(n)$ and $(ct^{(t)} = (\mathbf{c}^{(t)}_{in}, \mathbf{c}^{(t)}_{out}), cc^{(t)} = \{\mathbf{c}^{(t)}_i\}_{i\in[\ell]})$ is the ciphertext that has been transformed $t$ times, then we have $\|\mathbf{e}^{(t)}_{out}\| \leq \sqrt{m}B + 2km\sqrt{m}Bt + 2\sqrt{2}km^2(m+1)^d\sigma Bt$ and $\|\mathbf{e}^{(t)}_{in}\| \leq 2km\sqrt{m}B$ (See Appendix B for more details).

Therefore, $\|{\mathbf{e}^{(t)}_{in}}^T \mathbf{R}_\alpha + {\mathbf{e}^{(t)}_{out}}^T\| \leq 2km^2\sqrt{m}\sigma B + \sqrt{m}B + 2km\sqrt{m}B \cdot O(n) + 2\sqrt{2}km^2(m+1)^d\sigma B \cdot O(n) \leq B \cdot (m+1)^{O(d)} \leq q/4$ holds with overwhelming probability, which ensures the correctness.

**Theorem 4.** *Our multi-hop AB-CPRE scheme is sIND-CPA secure and sKP-CPA secure in standard model under $LWE_{n,q,\chi}$ assumption.*

Due to the space limitations, we just outline our proof sketch here. Our proof idea is similar to single-hop one. The difference between multi-hop scheme and single-hop scheme is the form of re-encryption key. In single-hop scheme, the re-encryption key $rk_{\theta,f\to\beta}$ contains a matrix $\mathbf{Q} \in \mathbb{Z}_q^{(2km+m)\times 2m}$. Whereas, in the multi-hop scheme, the re-encryption key $rk_{\theta,f\to\beta,\mathbf{y}}$ would contain an extra $\mathbf{P} \in \mathbb{Z}_q^{2km\times \ell m}$. Thus, in the sequence of sIND-CPA game or sKP-CPA game, $\mathcal{C}$ would generate an extra matrix $\mathbf{P}$ honestly, when asking for a re-encryption key (see Appendix C for more details).

## 6 Conclusion

In this paper, we propose two LWE-based AB-CPRE schemes against quantum-attack. Single-hop one is unidirectional, and supports fine-grained delegation of control as polynomial-deep circuit. Multi-hop one, an extension of single-hop scheme, is the first multi-hop AB-CPRE scheme. No matter how many transformation are performed, the ciphertext of multi-hop AB-CPRE is in constant size. Besides, we prove that both of our schemes are sIND-CPA and sKP-CPA without relying on random oracle.

At last, we leave two open problems. One is to construct an IND-CCA secure AB-CPRE scheme from lattices. Another is to construct a multi-hop lattice-based IND-CPA secure AB-CPRE scheme in adaptive model.

## References


1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (h) ibe in the standard model. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 553–572. Springer (2010)
2. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: 26th International Symposium on Theoretical Aspects of Computer Science STACS 2009. pp. 75–86. IBFI Schloss Dagstuhl (2009)
3. Aono, Y., Boyen, X., Wang, L., et al.: Key-private proxy re-encryption under lwe. In: International Conference on Cryptology in India. pp. 1–18. Springer (2013)
4. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Annual International Cryptology Conference. pp. 595–618. Springer (2009)
5. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 127–144. Springer (1998)
6. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 533–556. Springer (2014)
7. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical gapsvp. In: Annual Cryptology Conference. pp. 868–886. Springer (2012)
8. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. p. 575–584. STOC '13, Association for Computing Machinery, New York, NY, USA (2013)
9. Canetti, R., Hohenberger, S.: Chosen-ciphertext secure proxy re-encryption. In: Proceedings of the 14th ACM conference on Computer and communications security. pp. 185–194 (2007)
10. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Annual International Cryptology Conference. pp. 565–582. Springer (2003)
11. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 523–552. Springer (2010)
12. Fang, L., Wang, J., Ge, C., Ren, Y.: Fuzzy conditional proxy re-encryption. Science China Information Sciences **56**(5), 1–13 (2013)
13. Ge, C., Susilo, W., Wang, J., Fang, L.: Identity-based conditional proxy re-encryption with fine grain policy. Computer Standards & Interfaces **52**, 1–9 (2017)
14. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the fortieth annual ACM symposium on Theory of computing. pp. 197–206 (2008)
15. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: International Conference on Applied Cryptography and Network Security. pp. 288–306. Springer (2007)
16. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28**(4), 1364–1396 (1999)
17. Lai, J., Huang, Z., Au, M.H., Mao, X.: Constant-size cca-secure multi-hop unidirectional proxy re-encryption from indistinguishability obfuscation. Theoretical Computer Science **847**, 1–16 (2020)

18. Li, B., Xu, J., Liu, Y.: Lattice-based fuzzy conditional proxy re-encryption. Journal of Internet Technology **20**(5), 1379–1385 (2019)
19. Liang, K., Chu, C.K., Tan, X., Wong, D.S., Tang, C., Zhou, J.: Chosen-ciphertext secure multi-hop identity-based conditional proxy re-encryption with constant-size ciphertexts. Theoretical Computer Science **539**, 87–105 (2014)
20. Liang, K., Susilo, W., Liu, J.K., Wong, D.S.: Efficient and fully cca secure conditional proxy re-encryption from hierarchical identity-based encryption. The Computer Journal **58**(10), 2778–2792 (2015)
21. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Cryptographers' Track at the RSA Conference. pp. 319–339. Springer (2011)
22. Luo, F., Al-Kuwari, S., Wang, F., Chen, K.: Attribute-based proxy re-encryption from standard lattices. Theoretical Computer Science (2021)
23. Ma, C., Li, J., Ouyang, W.: Lattice-based identity-based homomorphic conditional proxy re-encryption for secure big data computing in cloud environment. International Journal of Foundations of Computer Science **28**(06), 645–660 (2017)
24. Mao, X., Li, X., Wu, X., Wang, C., Lai, J.: Anonymous attribute-based conditional proxy re-encryption. In: International Conference on Network and System Security. pp. 95–110. Springer (2018)
25. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 700–718. Springer (2012)
26. Mo, L., Yao, G.: Multi-use conditional proxy re-encryption. In: 2013 International Conference on Information Science and Cloud Computing Companion. pp. 246–251. IEEE (2013)
27. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. p. 333–342. STOC '09, Association for Computing Machinery, New York, NY, USA (2009)
28. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (Sep 2009)
29. Shao, J., Cao, Z.: Multi-use unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption. Information Sciences **206**, 83–95 (2012)
30. Shao, J., Wei, G., Ling, Y., Xie, M.: Identity-based conditional proxy re-encryption. In: 2011 IEEE International Conference on Communications (ICC). pp. 1–5. IEEE (2011)
31. Wang, H., Cao, Z., Wang, L.: Multi-use and unidirectional identity-based proxy re-encryption schemes. Information Sciences **180**(20), 4042–4059 (2010)
32. Weng, J., Deng, R.H., Ding, X., Chu, C.K., Lai, J.: Conditional proxy re-encryption secure against chosen-ciphertext attack. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security. pp. 322–332 (2009)
33. Yang, Y., Lu, H., Weng, J., Zhang, Y., Sakurai, K.: Fine-grained conditional proxy re-encryption and application. In: International Conference on Provable Security. pp. 206–222. Springer (2014)
34. Zhao, J., Feng, D., Zhang, Z.: Attribute-based conditional proxy re-encryption with chosen-ciphertext security. In: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010. pp. 1–6. IEEE (2010)

# A Full Proof for Single-hop AB-CPRE

In this section, we will present the full proof of single-hop AB-CPRE scheme. To make our proof more clear, the simulator algorithms are defined as follows, which will replace the original algorithms of AB-CPRE scheme gradually in the sequences of games.

- $\mathbf{Setup}_{SIM}(n, \mathbf{x}^*)$: Let $\mathbf{x}^* = \{x_i^*\}_{i\in[\ell]}$ to be the attribute vector selected by adversary $\mathcal{A}$. Sample a uniform matrix $\mathbf{D}_\theta \leftarrow \mathbb{Z}_q^{n\times m}$ and generate a random identity's public key $\mathbf{A}_\theta \leftarrow \mathbb{Z}_q^{n\times m}$ , then choose $\ell$ random matrices $\mathbf{S}_1^*, ..., \mathbf{S}_\ell^* \leftarrow \{-1, 1\}^{m\times m}$. Set $\mathbf{B}_i = \mathbf{A}_\theta \mathbf{S}_i^* - x_i^*\mathbf{G}$ for all $i \in [\ell]$. Select an error sampling algorithm $\chi$, which is a $B-$bounded distribution. Keep matrices $\{\mathbf{S}_i^*\}_{i\in[\ell]}$ as secret and return public parameters $pp := (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$ and specific public key $pk_\theta := (\mathbf{A}_\theta, \mathbf{D}_\theta)$.

- $\mathbf{Enc}_{SIM}(pp, pk_\theta, \boldsymbol{\mu}_b, \mathbf{x}^*)$: Let $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$, $pk_\theta = (\mathbf{A}_\theta, \mathbf{D}_\theta)$, a challenge message $\boldsymbol{\mu}_b \in \{0, 1\}^m$, and a selected attribute vector $\mathbf{x}^* = (\{x_i^*\}_{i\in[\ell]})$. Choose a random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and two error vectors $\mathbf{e}_{in}, \mathbf{e}_{out} \leftarrow \chi^m$. Compute $ct = (\mathbf{c}_{in}, \mathbf{c}_{out})$ as

$$\mathbf{c}_{in} = (\mathbf{A}_\theta)^T\mathbf{s} + \mathbf{e}_{in}, \mathbf{c}_{out} = (\mathbf{D}_\theta{}^T)\mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}_b.$$

  Use $\{\mathbf{S}_i^*\}_{i\in[\ell]}$ chosen in $Setup_{SIM}$ instead of uniform matrices in $\{-1, 1\}^{m\times m}$ and then assemble $cc^* = (\{\mathbf{c}_i = (x_i^*\mathbf{G} + \mathbf{B}_i)^T\mathbf{s} + (\mathbf{S}_i^*)^T\mathbf{e}_{in}\}_{i\in[\ell]}) \in \mathbb{Z}_q^{\ell m}$. Output a challenge ciphertext $CT^* = (ct^*, cc^*)$.

- $\mathbf{ReKeyGen}_{SIM-1}(pp, pk_\beta, f)$: Parse $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$ , $pk_\beta = (\mathbf{A}_\beta, \mathbf{D}_\beta)$, and a policy $f \in \mathcal{F}_{\ell,d}$. Let $\mathbf{B}_f = \mathbf{Eval}_{\mathrm{pk}}(f, \{\mathbf{B}_i\}_{i\in[\ell]})$ and a policy $\mathbf{F} = (\mathbf{A}_\theta|\mathbf{B}_f) \in \mathbb{Z}^{n\times 2m}$. To construct $\mathbf{R}_{\theta,f}$, build the basis $\mathbf{T}_{\theta,f}$ for $\mathbf{F}$ by executing $\mathbf{ExtendRight}(\mathbf{A}_\theta, \mathbf{T}_\theta, \mathbf{B}_f)$. Then run $\mathbf{SamplePre}(\mathbf{F}, \mathbf{T}_{\theta,f}, -\mathbf{D}_\theta, \sigma)$ to generate $\mathbf{R}_{\theta,f} \in \mathbb{Z}^{2m\times m}$ such that $\mathbf{F}\mathbf{R}_{\theta,f} = -\mathbf{D}_\theta$.
  1) In the case that $f(\mathbf{x}^*) \neq 0$, Set $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$, sample matrices $\mathbf{E}_1 \leftarrow \chi^{2km\times n}, \mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km\times m}$ and build matrix

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\theta,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

  2) In the case that $f(\mathbf{x}^*) = 0$, Set $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$, sample matrices $\mathbf{E}_1, \mathbf{E}_1' \leftarrow \chi^{2km\times n}, \mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km\times m}$ and build matrix

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1'\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\theta,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

  Output $rk_{\theta,f\to\beta} = \mathbf{Q}$ as re-encryption key.

- $\mathbf{ReKeyGen}_{SIM-2}(pp, pk_\beta, f)$: Parse $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$ , $pk_\beta = (\mathbf{A}_\beta, \mathbf{D}_\beta)$, and a policy $f \in \mathcal{F}_{\ell,d}$. Let $\mathbf{B}_f = \mathbf{Eval}_{\mathrm{pk}}(f, \{\mathbf{B}_i\}_{i\in[\ell]})$ and a policy $\mathbf{F} = (\mathbf{A}_\theta|\mathbf{B}_f) \in \mathbb{Z}^{n\times 2m}$. To construct $\mathbf{R}_{\theta,f}$, build the basis $\mathbf{T}_{\theta,f}$ for $\mathbf{F}$ by executing $\mathbf{ExtendRight}(\mathbf{A}_\theta, \mathbf{T}_\theta, \mathbf{B}_f)$. Then run $\mathbf{SamplePre}(\mathbf{F}, \mathbf{T}_{\theta,f}, -\mathbf{D}_\theta, \sigma)$ to generate $\mathbf{R}_{\theta,f} \in \mathbb{Z}^{2m\times m}$ such that $\mathbf{F}\mathbf{R}_{\theta,f} = -\mathbf{D}_\theta$.

1) In the case that $f(\mathbf{x}^*) \neq 0$, Set $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$, sample matrices $\mathbf{E}_1 \leftarrow \chi^{2km\times n}, \mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km\times m}$ and build matrix

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\theta,f}) \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

2) In the case that $f(\mathbf{x}^*) = 0$, Set $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$, sample matrices $\mathbf{E}_1 \leftarrow \chi^{2km\times n}, \mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km\times m}$, choose a matrix $\mathbf{M} \in \mathbb{Z}_q^{2km\times m}$ uniformly at random, and build matrix

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{M} + \overline{\mathbf{R}}_{\theta,f}) \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

Output $rk_{\theta,f\to\beta} = \mathbf{Q}$ as re-encryption key.

– $\mathbf{ReKeyGen}_{SIM-3}(pp, pk_\beta, f)$: Parse $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$ , $pk_\beta = (\mathbf{A}_\beta, \mathbf{D}_\beta)$, and a policy $f \in \mathcal{F}_{\ell,d}$. Let $\mathbf{S}_f^* = \mathbf{Eval}_{\mathrm{sim}}(f, (x_i^*, \mathbf{S}_i^*)_{i\in[\ell]}, \mathbf{A}_\theta)$.

1) In the case that $f(\mathbf{x}^*) \neq 0$, let $\mathbf{B}_f = \mathbf{A}_\theta\mathbf{S}_f^* - f(\mathbf{x}^*)\mathbf{G}$ and set $\mathbf{F} = [\mathbf{A}_\theta|\mathbf{B}_f - f(\mathbf{x}^*)\mathbf{G}] \in \mathbb{Z}^{n\times 2m}$. Compute the basis $\mathbf{T}_{\theta,f}$ for $\mathbf{F}$ as $\mathbf{T}_{\theta,f} \leftarrow \mathbf{ExtendLeft}(\mathbf{A}_\theta, \mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{S}_f)$. Then generate a matrix $\mathbf{R}_{\theta,f} \in \mathbb{Z}^{2m\times m}$ such that $\mathbf{F}\mathbf{R}_{\theta,f} = -\mathbf{D}_\theta$ by executing $\mathbf{SamplePre}(\mathbf{F}, \mathbf{T}_{\theta,f}, -\mathbf{D}_\theta, \sigma)$, set $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$, sample $\mathbf{E}_1 \leftarrow \chi^{2km\times n}, \mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km\times m}$ and compute

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\theta,f}) \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

2) In the case that $f(\mathbf{x}^*) = 0$,sample two matrices $\mathbf{E}_1 \leftarrow \chi^{2km\times n}, \mathbf{E}_2 \leftarrow \chi^{2km\times m}$, choose a matrices $\mathbf{M}' \leftarrow \mathbb{Z}^{2km\times m}$ uniformly at random, and compute

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{M}' \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

Output $rk_{\theta,f\to\beta} = \mathbf{Q}$ as re-encryption key.

### A.1 sIND-CPA

We now give a security proof of sIND-CPA as a sequence of games below.

**Game 0** This is the real sIND-CPA securtiy game.

**Game 1** Modify Phase 1 and Phase 2 in Game 0. When adversary $\mathcal{A}$ access to key generation query $\mathcal{O}_{rk}(pk_\theta, pk_\beta, f)$, where $pk_\beta \in \Psi_u, f(\mathbf{x}^*) = 1$, the challenger $\mathcal{C}$ would execute $\mathbf{ReKeyGen}_{SIM-1}$ to generate a re-encryption key $rk_{\theta,f\to\beta}$ instead.

**Game 2** This game is same as Game 1 except that in Phase 1 and Phase 2. When $\mathcal{C}$ receives a re-encryption key query $\mathcal{O}_{rk}(pk_\theta, pk_\beta, f)$ where $f(\mathbf{x}^*) = 0$ and $pk_\beta \in \Psi_u$, then $\mathcal{C}$ generates a re-encryption key $rk_{\theta,f\to\beta}$ by executing $\mathbf{ReKeyGen}_{SIM-2}$ instead.

**Game 3** We now continue to change Phase 1 and Phase 2 in Game 2. When $\mathcal{A}$ access to $\mathcal{O}_{rk}(pk_\theta, pk_\beta, f)$, $\mathcal{C}$ would use $\mathbf{ReKeyGen}_{SIM-3}$ algorithm to generate re-encryption key $rk_{\theta,f\to\beta}$.

**Game 4** Modify Setup phase in Game 2. Using $\mathbf{Setup}_{SIM}$ algorithm instead to generate the public parameters $pp$ and the specific identity's public key $(\mathbf{A}_\theta, \mathbf{D}_\theta)$.

**Game 5** We now change how challenge ciphertext is generated. In Challenge phase, challenger would use $\mathbf{Enc}_{SIM}$ to generate the challenge ciphertext.

**Game 6** This game is identitcal to Game 4 except that the challenge ciphertext $CT^* = (ct^*, cc^*) \in \mathbb{Z}^{(\ell+2)m}$ is chosen uniformly at random in $\mathbb{Z}^{(\ell+2)m}$. Since challenge ciphertext is a random element in ciphertext space, which is independent of $\boldsymbol{\mu}_0^*$ and $\boldsymbol{\mu}_1^*$, $\mathcal{A}$'s advantage in this game is zero.

**Theorem 5.** *Our scheme is sIND-CPA secure under $LWE_{n,q,\chi}$ assumption.*

*Proof.* Combine the following Lemma 5, 6, 7, 8, 9 and 10 together, then we could prove that our scheme is sIND-CPA secure.

**Lemma 5.** *Game 0 is computational indistinguishable from Game 1.*

*Proof.* Recalling that the difference between Game 0 and Game 1 is how the re-encryption key is generated. When $pk_\beta \in \Psi_u$ and $f(\mathbf{x}^*) = 0$ hold, the re-encryption keys $rk_{\theta,f\to\beta}$ are showed as follows,

$$rk_{\theta,f\to\beta} = \begin{cases} \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\theta,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} & \text{in Game 0} \\ \\ \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1'\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\theta,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} & \text{in Game 1} \end{cases}.$$

By Corollary 2, we can learn that under LWE assumption, Game 0 is computational indistinguishable from Game 1.

**Lemma 6.** *Game 1 is computational indistinguishable from Game 2, otherwise there exists an efficient algorithm to solve $LWE_{n,q,\chi}$ problem.*

*Proof.* When $f(\mathbf{x}^*) = 0$ and $pk_\beta \in \Psi_u$ hold, in Game 1, the re-encryption key is generated by $\mathbf{ReKeyGen}_{SIM-1}$ algorithm. In the meanwhile, in Game 2, re-encryption key is produced by $\mathbf{ReKeyGen}_{SIM-2}$. We show the re-encryption key $rk_{\theta,f\to\beta}$ in details:

$$rk_{\theta,f\to\beta} = \begin{cases} \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\theta,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} & \text{in Game 1} \\ \\ \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{M} + \overline{\mathbf{R}}_{\theta,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} & \text{in Game 2} \end{cases}$$

where $\mathbf{M} \leftarrow \mathbb{Z}^{2km\times m}, \mathbf{E}_1 \leftarrow \chi^{2km\times n}, \mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km\times m}$. Noted that, matrix $\mathbf{D}_\beta$ is selected uniformly at random, which means that $(\mathbf{D}_\beta, \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{2km\times m}$ is an LWE pair in Hermite normal form. Then, if $\mathcal{A}$ can distinguish Game 1 and Game 2, we can construct an Algorithm 1 to solve $LWE_{n,q,\chi}$ problem.

**Algorithm 1**

**Input:** $(\{\mathbf{Y}_\beta, \mathbf{Z}_{\beta,1}, ..., \mathbf{Z}_{\beta,t}\}_{\beta\in[t]})$

**Output:** Distinguish whether $(\{\mathbf{Y}_\beta, \mathbf{Z}_{\beta,1}, ..., \mathbf{Z}_{\beta,t}\}_{\beta\in[t]})$ is LWE pair.

1: Execute the Init and Setup phases in real sIND-CPA game.

2: Answer $\mathcal{A}$'s queries as Game 1. But in Phase 1 and Phase 2,

1) When $\mathcal{A}$ submits a query $q_j$ on $\mathcal{O}_u(\beta)$ for $j \in [t]$, execute **TrapGen** algorithm to get a matrix $\mathbf{A}_\beta$, set $\mathbf{D}_\beta = \mathbf{Y}_\beta$, then outputs $(\mathbf{A}_\beta, \mathbf{D}_\beta)$ as user $\beta$'s public key.
2) When $\mathcal{A}$ submits a query $q_j$ on $\mathcal{O}_{rk}(pk_\theta, pk_\beta, f)$ such that $f(\mathbf{x}^*) = 0$ and $pk_\beta \in \Psi_u$, sample matrices $\mathbf{E}_1 \leftarrow \chi^{2km\times n}, \mathbf{E}_2 \leftarrow \chi^{2km\times m}$, set $\mathbf{M} = \mathbf{Z}_{\beta,j}$, and then assemble the re-encryption key as following:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{M} + \overline{\mathbf{R}}_{\theta,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times m}.$$

3: **Return** $\mathcal{A}$'s answer.

If $\mathbf{M} = \mathbf{Z}_{\beta,j} = \mathbf{E}_1'\mathbf{Y}_\beta + \mathbf{E}_3 = \mathbf{E}_1'\mathbf{D}_\beta + \mathbf{E}_3$, then Algorithm 1 is exactly act as Game 1. Otherwise, it act as Game 2 when $f(\mathbf{x}^*) = 0$ and $pk_\beta \in \Psi_u$ hold. In short, by Corollary 2, under $LWE_{n,q,\chi}$ assumption, Game 1 is computational indistinguishable from Game 2.

**Lemma 7.** *Game 2 is statistically indistinguishable from Game 3.*

*Proof.* Considering that in Game 3, when $\mathcal{A}$ queries a function $f$ such that $f(\mathbf{x}^*) = 1$, which means $\mathbf{B}_f = \mathbf{A}_\theta\mathbf{S}_f - f(\mathbf{x}^*)\mathbf{G}$, we could use **ExtendLeft** algorithm to generate a trapdoor for $\mathbf{B}_f$. Since, Gaussian parameter $\sigma$ is the same in both games, by Lemma 3, $\mathbf{R}_{\theta,f}$ in Game 2 and Game 3 are statistically close to $\mathcal{D}_\sigma(\Lambda_q^D(\mathbf{A}_\theta|\mathbf{B}_f))$. Hence, Game 2 is statistically indistinguishable from Game 3.

**Lemma 8.** *Game 3 is statistically indistinguishable from Game 4.*

*Proof.* Recall that public matrices $\{\mathbf{B}_i\}_{i\in[\ell]}$ in Game 3 are chosen uniformly, whereas in Game 4 public matrices are $\{\mathbf{B}_i = \mathbf{A}_\theta\mathbf{S}_i^* - x_i^*\mathbf{G}\}_{i\in[\ell]}$. By lemma 1, we can learn that $\mathbf{A}_\theta\mathbf{S}_i^*$ is statistically indistingushable from uniform in $\mathbb{Z}_q^{n\times m}$. Therefore, Game 3 is statistically indistinguishable from Game 4.

**Lemma 9.** *Game 4 is statistically indistinguishable from Game 5.*

*Proof.* In Game 5, matrices $\mathbf{S}_i^*$ in $\mathbf{Enc}_{SIM}$ share the randomness with $\mathbf{S}_i^*$ in $\mathbf{Setup}_{SIM}$. However, through sIND-CPA game, encryption algorithm is only executed once by challenger to generate the challenge ciphtertext. So we have Game 4 is statistically indistinguishable from Game 5.

**Lemma 10.** *Game 5 is computational indistinguishable from Game 6, otherwise there exists an efficient algorithm to solve $LWE_{n,q,\chi}$ problem.*

*Proof.* We show the challenge ciphertext in Game 5.

$$\mathbf{c}_{in} = (\mathbf{A}_\theta)^T\mathbf{s} + \mathbf{e}_{in} \ , \ \mathbf{c}_{out} = (\mathbf{D}_\theta)^T\mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}_b^*,$$
$$cc = \{\mathbf{c}_i = (x_i^*\mathbf{G} + \mathbf{B}_i)^T\mathbf{s} + (\mathbf{S}_i^*)^T\mathbf{e}_{in}\}_{i\in[\ell]}).$$

Then, by $\mathbf{B}_i = \mathbf{A}_\theta\mathbf{S}_i^* - x_i^*\mathbf{G}$, we have that

$$\begin{aligned}\left[\mathbf{c}_1^T|.....|\mathbf{c}_\ell^T\right] &= \left[\mathbf{s}^T\mathbf{A}_\theta\mathbf{S}_1^* + \mathbf{e}_{in}^T\mathbf{S}_1^*|.....|\mathbf{s}^T\mathbf{A}_\theta\mathbf{S}_\ell^* + \mathbf{e}_{in}^T\mathbf{S}_\ell^*\right] \\ &= (\mathbf{s}^T\mathbf{A}_\theta + \mathbf{e}_{in}^T)\left[\mathbf{S}_1^*|.....|\mathbf{S}_\ell^*\right].\end{aligned}$$

We build the Algorithm 2 that uses $\mathcal{A}$ to solve $LWE_{n,q,\chi}$ problem. Algorithm 2 is given an LWE instance $(\mathbf{Y}, \mathbf{b}) \in \mathbb{Z}_q^{n\times 2m} \times \mathbb{Z}_q^{2m}$, answers whether $(\mathbf{Y}, \mathbf{b})$ is selected randomly or $\mathbf{b} = \mathbf{Y}^T\mathbf{s} + \mathbf{e}$ holds for some noise $\mathbf{e} \in \chi^m$.

**Algorithm 2**

**Input:** $(\mathbf{Y}, \mathbf{z})$

**Output:** Distinguish whether $(\mathbf{Y}, \mathbf{z})$ is LWE pair.

1: Let $[\mathbf{A}_\theta|\mathbf{D}_\theta] := \mathbf{Y}$

2: Using LWE instance and $\mathbf{Setup}_{SIM}$ to generate the public parameters $pp = (\{\mathbf{B}_i = \mathbf{A}_\theta\mathbf{S}_i^* - \mathbf{x}_i^*\mathbf{G}\}_{i\in[\ell]}, \chi)$ and public key $pk := (\mathbf{A}_\theta, \mathbf{D}_\theta)$.

3: Answer $\mathcal{A}$'s queries as in Game 4.

4: In the challenger phase, generate the challenge ciphertext by LWE instance

$$[\mathbf{c}_{in}; \mathbf{c}_{out}] := \mathbf{z},$$

$$\left[\mathbf{c}_1^T|...|\mathbf{c}_\ell^T\right] = \mathbf{c}_{in}^T\left[\mathbf{S}_1^*|...|\mathbf{S}_\ell^*\right].$$

5: **Return** $\mathcal{A}$'s answer.

In the case that, $\mathbf{z} = [\mathbf{c}_{in}; \mathbf{c}_{out}] = \mathbf{Y}^T\mathbf{s} + \mathbf{e} = [\mathbf{A}_\theta|\mathbf{D}_\theta]^T\mathbf{s} + [\mathbf{e}_{in}; \mathbf{e}_{out}]$, where $\mathbf{Y} \leftarrow \mathbb{Z}_q^{n\times 2m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{e} \leftarrow \chi^{2m}$. The challenge ciphertext is the same distribution in Game 5. In the other case, if $\mathbf{Y}$ and $\mathbf{z}$ are chosen uniformly, by leftover hash lemma [16], the challenge ciphertext is statistically indistinguishable from uniform.

In other words, if $\mathcal{A}$ can distinguish Game 5 and Game 6, then there exists an efficient Algorithm 2 to solve $LWE_{n,q,\chi}$ problem, which have demonstrated that Game 5 is computational indistinguishable from Game 6.

### A.2 sKP-CPA

This proof is barely based on sIND-CPA game sequences. We make some subtle changes in Challenge phase from Game 3 to complete our security proof. The securtiy proof of sKP-CPA is done through a sequence of games below.

**Game 0'** This is the real sKP-CPA game.

**Game 1'** Modify Setup phase in Game 0'. Using $\mathbf{Setup}_{SIM}$ algorithm to generate the public parameters $pp$ and the specific identity's public key $\mathbf{A}_\theta$. Besides, in Phase 1 and Phase 2, challenger use $\mathbf{ReKeyGen}_{SIM-3}$ to generate the re-encryption keys.

**Game 2'** In Challenge phase, challenger would generate a random re-encryption key in re-encryption key space as challenge re-encryption key. In this game, $\mathcal{A}$'s advantage is zero.

**Theorem 6.** *Our AB-CPRE scheme is sKP-CPA secure under $LWE_{n,q,\chi}$ assumption.*

*Proof.* Combining the following Lemma 11 and 12 together, then we proof that our scheme is sKP-CPA secure.

**Lemma 11.** *Game 0' is computational indistinguishable from Game 1'.*

*Proof.* It is the same as proofing that Game 0 is computational indistinguishable from Game 3. By lemma 5, 6 and 7, we proof that Game 0' is computational indistinguishable from Game 1'.

**Lemma 12.** *Game 1' is computational indistinguishable from Game 2'.*

*Proof.* Compare the challenge re-encryption key in Game 1' and Game 2',

$$\mathbf{Q} = \begin{cases} \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\theta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\theta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\beta,f'} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} & \text{in Game 1'} \\ \begin{bmatrix} \mathbf{N} & \mathbf{M} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} & \text{in Game 2'} \end{cases}.$$

Given (Hermite normal form of) LWE instance $[\mathbf{A}_\theta|\mathbf{D}_\theta] := \mathbf{Y}$, $[\mathbf{N}|\mathbf{M}] := \mathbf{Z}$, if $\mathbf{N} = \mathbf{E}_1\mathbf{A}_\theta + \mathbf{E}_2$ and $\mathbf{M} = \mathbf{E}_1\mathbf{D}_\theta + \mathbf{E}_3$ hold, where the elements of $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$ are sampled from noise distribution $\chi$. Then it is exactly acting as in Game 1'. Otherwise, $\mathbf{N}, \mathbf{M}$ are sampled uniformly at random, then $\mathbf{M} + \overline{\mathbf{R}}_{\beta,f'}$ is statistically from uniform, which is exactly in Game 2'.

## B Correctness for Multi-hop AB-CPRE

The correstness of original ciphertext is the same as the correctness in Section 4. Then the correctness of transformed ciphertext is presented as follows.

*Transformed Ciphertext.* $(ct^{(t-1)} = (\mathbf{c}_{in}^{(t-1)}, \mathbf{c}_{out}^{(t-1)}), cc^{(t-1)} = \{\mathbf{c}_i^{(t-1)}\}_{i\in[\ell]})$ is the ciphertext which has been transformed $t-1$ times and associated with attribute vector $\mathbf{x}$ under $pk_\alpha$. We show $(cc^{(t-1)}, ct^{(t-1)})$ in detail as following;

$$\mathbf{c}_{in}^{(t-1)} = \mathbf{s}^{(t-1)^T}\mathbf{A}_\alpha + \mathbf{e}_{in}^{(t-1)} \ , \ \mathbf{c}_{out}^{(t-1)} = \mathbf{s}^{(t-1)^T}\mathbf{D}_\alpha + \mathbf{e}_{out}^{(t-1)} + \lfloor q/2 \rfloor \boldsymbol{\mu},$$
$$\{\mathbf{c}_i^{(t-1)} = \mathbf{s}^{(t-1)^T}(x_i\mathbf{G} + \mathbf{B}_i) + \mathbf{e}_i^{(t-1)}\}_{i\in[\ell]}.$$

For convenience, $rk_{\alpha,f\to\beta,\mathbf{y}} = (\mathbf{Q}, \mathbf{P})$ is the re-encryption key, where $f(\mathbf{x}) = 0$. Set $\widetilde{\mathbf{c}}_{in,f}^{(t-1)} = \mathbf{Bits}_q([\mathbf{c}_{in}^{(t-1)}; \mathbf{c}_f^{(t-1)}])$ and $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$. Then the $t$ times transformed ciphertext $(ct^{(t)}, cc^{(t)})$ is as following;

$$\begin{aligned}
\mathbf{c}_{in}^{(t)} &= \mathbf{s}^{(t)^T}\mathbf{A}_\beta + \mathbf{e}_{in}^{(t)} = (\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_1\mathbf{A}_\beta + (\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_2, \\
\mathbf{c}_{out}^{(t)} &= \mathbf{s}^{(t)^T}\mathbf{D}_\beta + \mathbf{e}_{out}^{(t)} + \lfloor q/2 \rfloor \boldsymbol{\mu} \\
&= (\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_1\mathbf{D}_\beta + (\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_3 + [\mathbf{e}_{in}^{(t-1)^T}|\mathbf{e}_f^{(t-1)^T}]\mathbf{R}_{\alpha,f} + \mathbf{e}_{out}^{(t-1)} + \lfloor q/2 \rfloor \boldsymbol{\mu}, \\
\{\mathbf{c}_i^{(t)} &= \mathbf{s}^{(t)^T}(y_i\mathbf{G} + \mathbf{B}_i) + \mathbf{e}_i^{(t)} = (\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_1(y_i\mathbf{G} + \mathbf{B}_i) + (\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_{B_i}\}_{i\in[\ell]}.
\end{aligned}$$

Obviously, for any $t > 0$, we can learn that,

$$\begin{aligned}
\|\mathbf{e}_{in}^{(t)}\| &\le \|(\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_2\| \le 2km\sqrt{m}B, \\
\|\mathbf{e}_{out}^{(t)}\| &\le \|(\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_3\| + \|[\mathbf{e}_{in}^{(t-1)^T}|\mathbf{e}_f^{(t-1)^T}]\mathbf{R}_{\alpha,f}\| + \|\mathbf{e}_{out}^{(t-1)}\|, \\
\|\mathbf{e}_i^{(t)}\| &\le \|(\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_{B_i}\| \le 2km\sqrt{m}B.
\end{aligned}$$

Because $\|\mathbf{e}_{out}^{(0)}\| \le \sqrt{m}B$ and $\|\mathbf{e}_f^{(t)}\| \le \sqrt{2}mkB(m+1)^d$, then we have,

$$\begin{aligned}
\|\mathbf{e}_{out}^{(t)}\| &\le \|\mathbf{e}_{out}^{(0)}\| + t\|(\widetilde{\mathbf{c}}_{in,f}^{(t-1)})^T\mathbf{E}_3\| + t\|[\mathbf{e}_{in}^{(t-1)^T}|\mathbf{e}_f^{(t-1)^T}]\mathbf{R}_{\alpha,f}\| \\
&\le \sqrt{m}B + 2km\sqrt{m}Bt + 2\sqrt{2}km^2(m+1)^d\sigma Bt.
\end{aligned}$$

Therefore, for the $t$ times transformed ciphertext $(ct^{(t)}, cc^{(t)})$,

$$\begin{bmatrix}\mathbf{c}_{in}^{(t)^T} & \mathbf{c}_{out}^{(t)^T}\end{bmatrix} \cdot \begin{bmatrix}\mathbf{R}_\alpha \\ \mathbf{I}_{m\times m}\end{bmatrix} = \mathbf{e}_{in}^{(t)^T}\mathbf{R}_\alpha + \mathbf{e}_{out}^{(t)^T} + \lfloor q/2 \rfloor \boldsymbol{\mu}^T.$$

where $\|\mathbf{e}_{in}^{(t)^T}\mathbf{R}_\alpha + \mathbf{e}_{out}^{(t)^T}\| \le \|\mathbf{e}_{in}^{(t)^T}\| \cdot \|\mathbf{R}_\alpha\| + \|\mathbf{e}_{out}^{(t)^T}\| \le 2km^2\sqrt{m}\sigma B + \sqrt{m}B + 2km\sqrt{m}Bt + 2\sqrt{2}km^2(m+1)^d\sigma Bt \le B \cdot (m+1)^{O(d)} \le q/4$ with overwhelming probability, which ensures the correctness.

## C Simulator Algorithms for Multi-hop AB-CPRE

The proof idea is quite similar to the proof of single-hop AB-CPRE. For the sake of breifness, simulator algorithms are showed as follows,

- $\mathbf{Setup}_{SIM}(n, \mathbf{x}^*)$: Let $\mathbf{x}^* = \{x_i^*\}_{i\in[\ell]}$ to be the attribute vector selected by adversary $\mathcal{A}$. Sample a uniform matrix $\mathbf{D}_\theta \leftarrow \mathbb{Z}_q^{n\times m}$ and generate a random identity's public key $\mathbf{A}_\theta \leftarrow \mathbb{Z}_q^{n\times m}$ , then choose $\ell$ random matrices $\mathbf{S}_1^*, ..., \mathbf{S}_\ell^* \leftarrow \{-1, 1\}^{m\times m}$. Set $\mathbf{B}_i = \mathbf{A}_\theta \mathbf{S}_i^* - x_i^*\mathbf{G}$ for all $i \in [\ell]$. Select an error sampling algorithm $\chi$, which is a $B-$bounded distribution. Keep matrices $\{\mathbf{S}_i^*\}_{i\in[\ell]}$ as secret and return public parameters $pp := (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$ and specific public key $pk_\theta := (\mathbf{A}_\theta, \mathbf{D}_\theta)$.

- $\mathbf{Enc}_{SIM}(pp, pk_\theta, \boldsymbol{\mu}_b, \mathbf{x}^*)$: Let $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$, $pk_\theta = (\mathbf{A}_\theta, \mathbf{D}_\theta)$, a challenge message $\boldsymbol{\mu}_b \in \{0, 1\}^m$, and a selected attribute vector $\mathbf{x}^* = (\{x_i^*\}_{i\in[\ell]})$. Choose a random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ and two error vectors $\mathbf{e}_{in}, \mathbf{e}_{out} \leftarrow \chi^m$. Compute $ct = (\mathbf{c}_{in}, \mathbf{c}_{out})$ as

$$\mathbf{c}_{in} = (\mathbf{A}_\theta)^T\mathbf{s} + \mathbf{e}_{in}, \mathbf{c}_{out} = (\mathbf{D}_\theta{}^T)\mathbf{s} + \mathbf{e}_{out} + \lfloor q/2 \rfloor \boldsymbol{\mu}_b.$$

  Use $\{\mathbf{S}_i^*\}_{i\in[\ell]}$ chosen in $Setup_{SIM}$ instead of uniform matrices in $\{-1, 1\}^{m\times m}$ and then assemble $cc^* = (\{\mathbf{c}_i = (x_i^*\mathbf{G} + \mathbf{B}_i)^T\mathbf{s} + (\mathbf{S}_i^*)^T\mathbf{e}_{in}\}_{i\in[\ell]}) \in \mathbb{Z}_q^{\ell m}$. Output a challenge ciphertext $CT^* = (ct^*, cc^*)$.

- $\mathbf{ReKeyGen}_{SIM}(pp, pk_\beta, f, \mathbf{y})$: Parse $pp = (\{\mathbf{B}_i\}_{i\in[\ell]}, \chi)$, $pk_\beta = (\mathbf{A}_\beta, \mathbf{D}_\beta)$, a policy $f \in \mathcal{F}_{\ell,d}$ and an attribute vector $\mathbf{y} = \{y_i\}_{i\in[\ell]}$. Compute $\mathbf{S}_f^*$ by executing $\mathbf{Eval}_{\text{sim}}(f, (x_i^*, \mathbf{S}_i^*)_{i\in[\ell]}, \mathbf{A}_\theta)$, sample matrices $\mathbf{E}_1 \leftarrow \chi^{2km\times n}$, $\mathbf{E}_2, \mathbf{E}_3 \leftarrow \chi^{2km\times m}$.
  1) In the case that $f(\mathbf{x}^*) \neq 0$, let $\mathbf{B}_f = \mathbf{A}_\theta\mathbf{S}_f^* - f(\mathbf{x}^*)\mathbf{G}$ and set $\mathbf{F} = [\mathbf{A}_\theta|\mathbf{B}_f - f(\mathbf{x}^*)\mathbf{G}] \in \mathbb{Z}^{n\times 2m}$. Compute the basis $\mathbf{T}_{\theta,f}$ for $\mathbf{F}$ as $\mathbf{T}_{\theta,f} \leftarrow \mathbf{ExtendLeft}(\mathbf{A}_\theta, \mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{S}_f)$. Then generate a matrix $\mathbf{R}_{\theta,f} \in \mathbb{Z}^{2m\times m}$ such that $\mathbf{F}\mathbf{R}_{\theta,f} = -\mathbf{D}_\theta$ by executing $\mathbf{SamplePre}(\mathbf{F}, \mathbf{T}_{\theta,f}, -\mathbf{D}_\theta, \sigma)$, set $\overline{\mathbf{R}}_{\alpha,f} = \mathbf{Power2}_q(\mathbf{R}_{\alpha,f})$, and compute

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \overline{\mathbf{R}}_{\theta,f} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

  2) In the case that $f(\mathbf{x}^*) = 0$, choose a matrices $\mathbf{M} \leftarrow \mathbb{Z}^{2km\times m}$ uniformly at random, and compute

$$\mathbf{Q} = \begin{bmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{M} \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{bmatrix} \in \mathbb{Z}_q^{(2km+m)\times 2m}.$$

  If $\mathbf{y}$ is none or null, then set $\mathbf{P}$ as a null matrix. Otherwise, samples $\ell$ matrices $\mathbf{E}_{B_i}$ from error distribution $\chi^{2km\times m}$ and compute,

$$\mathbf{P} = [(\mathbf{E}_1(y_i\mathbf{G} + \mathbf{B}_1) + \mathbf{E}_{B_1}) \mid ... \mid (\mathbf{E}_1(y_i\mathbf{G} + \mathbf{B}_\ell) + \mathbf{E}_{B_\ell})] \in \mathbb{Z}_q^{2km\times \ell m}.$$

  Output $rk_{\theta,f\to\beta,\mathbf{y}} = (\mathbf{Q}, \mathbf{P})$ as re-encryption key.