HRA-Secure Proxy Re-encryption with Re-encryption Simulatability under Lwe in Standard Model

Bimei Wang , Li Hou , Lisha Yao , Feixiang Zhao , Jian Weng *

Posted Date: 28 November 2023

doi: 10.20944/preprints202311.1769.v1

Keywords: Proxy re-encryption; re-encryption simulatability; honest re-encryption attacks; learning with errors; quantum-resistant

Article

HRA-Secure Proxy Re-Encryption with Re-Encryption Simulatability under LWE in Standard Model

Bimei Wang, Lin Hou, Lisha Yao, Feixiang Zhao and Jian Weng *

College of Cyber Security, Jinan University, Guangzhou, 510632, China

* Correspondence: cryptjweng@gmail.com; Tel.: +86-15920484703

Abstract: Proxy re-encryption (PRE) is a momentous and widely used cryptographic technique. It enables a proxy to forward ciphertext without the need of decryption. PRE has received significant interest in applications like cloud computing, blockchain, and the Internet of Things. Despite its wide range of uses, PRE has also been subject to new security and privacy regulations. In PKC'19, Cohen *et al*. first drew attention to the weakness in PRE's security against chosen-plaintext attacks (CPA) and put up a more stringent security concept known as security against honest re-encryption attacks (HRA). Notably, Cohen provides a beneficial conclusion as well. It is proveded that PRE schemes with re-encryption simulatability property can be elevated from CPA to HRA security. It is also proved that CPA-secure PRE schemes with re-encryption simulatability property can be directly elevated to those satisfying HRA security. However, those PRE schemes with re-encryption simulatability are almost always based on pairings. In this study, to the best of our knowledge, we directly construct HRA-secure PRE with re-encryption simulatability for the first time based on the learning with errors (LWE) assumption, which was widely believed to be quantum-resistant. Based on the re-encryption key generation algorithm and the re-encryption algorithm construction method of the above scheme, we can modify attribute-based conditional proxy re-encryption (AB-CPRE) as well as the corresponding attribute-based proxy re-encryption (AB-PRE) algorithm to make them have re-encryption simulatability properties. Finally, by using this property, We boost the security of AB-CPRE scheme of ESORICS'21 from CPA to HRA and simplify the HRA-security proof for the AB-PRE of ESORICS'21.

Keywords: proxy re-encryption; re-encryption simulatability; honest re-encryption attacks; learning with errors; quantum-resistant

1. Introduction

Proxy re-encryption (PRE) was first proposed at the EUROCRYPT'98 [1]. It can transform ciphertexts between keys without decrypting, enableing powerful cryptographic workflows while maintaining confidentiality. PRE has numerous applications, such as electronic medical systems [2], data sharing [3], email systems [4], and has drawn increasing attention in applications of more fields, e.g., cloud computing [5–7], Internet of Things [8], and blockchain [9,10]. In these industrial scenarios, the data involved is usually sensitive business secrets, such as customer data and financial data. The leakage or tampering of these data can cause severe economic and reputational losses to enterprises or individuals, and even lead to legal disputes. Therefore, in the above industrial scenarios, the security requirements for PRE are higher, and more stringent security standards must be met to ensure the confidentiality of data.

It is challenging to design a PRE scheme with security under chosen-ciphertext attacks (CCA), while PRE schemes with security under chosen-plaintext attacks (CPA) fail to consider that there may exist the re-encryption from honest user to corrupted user in real-world scenarios. Therefore, Cohen *et al*. [11] proposed security under honest re-encryption attacks (HRA), which better captures the inadequacy of CPA-secure PRE for various applications. Generally speaking, HRA is more robust than CPA. Cohen *et al*. [11] also proposed an exciting property called re-encryption simulatability to enhance PRE schemes from CPA security to HRA security.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

Re-encryption simulatability refers to the ciphertext generated by the re-encryption algorithm can be simulated without knowing the private key of the delegator. Although the re-encryption simulatability property is not necessary for HRA security, it can not only boost the security of PRE from CPA to HRA but also significantly reduce the difficulty of proof of HRA-secure PRE schemes. Taking advantage of re-encryption simulatability, the oracle can effortlessly answer the adversary's re-encryption queries, transforming a ciphertext under an honest user to a ciphertext under a corrupted user. This greatly motivates us to explore PRE schemes with this property.

Until now, only a few schemes have this desirable property. However, these schemes either lack resistance against quantum attacks or suffer from inefficiency. Cohen *et al*. [11] presented two schemes with this property. One is a pairing-based PRE scheme that is not resistant to quantum computer attacks [12]. The other is a quantum-safe PRE constructed from fully homomorphic encryption (FHE) based bootsrapping but is inefficient [13]. One is based on pairings that is vulnerable to attacks by quantum computers [12]. The other is a quantum-safe PRE from fully homomorphic encryption (FHE) based bootsrapping [13], which is inefficient. Although, in recent years, there has been a proliferation of PRE schemes based on key switching technique under the learning with errors (LWE) assumption, which are efficient and resistant to quantum computer's attacks. To the best of our knowledge, none of them possesses re-encryption simulatability property.

1.1. Our Contributions

To address these aforementioned challenges, we primarily develop a succinct lattice-based PRE scheme, which is both quantum resistant and has the property of re-encryption simulatability. The design idea can also be applied to other types of PRE schemes. We show a comparison between existing PRE schemes and ours, as shown in Table 1. Contributions are as follows:

Table 1. Evaluating our three schemes in comparison to existing methods.

Scheme	Type	Assumptions	Security	Quantum-resistant	Standard Model	Re-encryption Simulatability
[12]	PRE	DBDH	CPA	×	✓	×
[11][1]	PRE	DBDH	HRA	×	✓	✓
[13]	*PRE*⋆	–	CPA	✓	✓	×
[11][2]	*PRE*⋆	–	HRA	✓	✓	✓
Scheme I(Ours)	PRE	LWE	HRA	✓	✓	✓
[14]	AB-CPRE	LWE	CPA	✓	✓	×
Scheme II(Ours)	AB-CPRE	LWE	HRA	✓	✓	✓
[15]	AB-PRE	LWE	CPA	✓	✓	×
[16]	AB-PRE	LWE	HRA	✓	✓	×
Scheme III(Ours)	AB-PRE	LWE	HRA	✓	✓	✓

1 PRE scheme based on [12]; 2 PRE scheme based on [13]; *PRE*⋆ is a sufficiently somewhat homomorphic encryption scheme that incorporates bootstrapping and implies CPA secure PRE

- **PRE with Re-encryption Simulatability.** We construct a concise lattice-based PRE scheme directly, scheme I,using key switching technique. Compared with other schemes, the main attraction of our proposed scheme is its re-encryption simulatability property. This property has been easily proven to be HRA-secure in the standard model under the LWE assumption. Most importantly, the methods for constructing re-encryption key generation and re-encryption algorithms can be extended to other schemes related to PRE to make them have re-encryption simulatability.
- **AB-CPRE with Re-encryption Simulatability.** We apply the methods above to AB-CPRE scheme presented at ESORICS'21 [14]. We first formalize the HRA security model for AB-CPRE in this work and obtain a modified AB-CPRE scheme, scheme III. Besides, we also prove that scheme III has re-encryption simulatability and boost the security of AB-CPRE from selective CPA to selective HRA.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

- **AB-PRE with Re-encryption Simulatability.** We apply the methods above to AB-PRE presented at ESORICS'21 [16]. We obtain an improved AB-PRE scheme, scheme III, which has the re-encryption simulatability property.

1.2. Organization

In Section 2, we present the preliminaries, which encompass some algorithms of lattice, as well as relevant functions. Section 3 introduces some definitions about PRE, including the definition of re-encryption simulatability. The PRE scheme we proposed, along with its re-encryption simulatability property and security proof of HRA-secure, is presented in Section 4. Section 5 provides the selective HRA security model for AB-CPRE and a modified AB-CPRE scheme. We also boost the security of AB-CPRE from selective CPA to selective HRA in Section 5. Additionally, Section 6 discusses the re-encryption simulatability property of AB-PRE. In the end, the conclusion of this work is provided in Section 7.

2. Preliminaries

First, we give a description of notations mainly involved in this study, as shown in Table 2. Moreover, this section contains the Decisional Learning with Errors (DLWE) assumption, a few functions and algorithms. For instance, the vector decomposition function, preimage sampling (SamplePre) algorithm, and trapdoor generation (TranGen) algorithm.

Table 2. Main notations in this work.

Notation	Description
$\mathbf{A}$	A matrix.
$\mathbf{s}$	A vector.
$(\cdot\|\|\cdot)$	Horizontal concatenation.
$\xleftarrow{\$}$	Sample a matrix or vector randomly.
$[m]$	$1, 2, \cdots, m$.
$\Lambda(\mathbf{A})$	A lattice.
$\Lambda_q^{\perp}(\mathbf{A})$	A q-ary integer lattice.
$\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\tau}$	A discrete Gaussian distribution.

Definition 1. *The Decisional Learning with Errors (DLWE) problem refers to* $(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e})$ *and* $(\mathbf{A}, \mathbf{u})$ *are computationally indistinguishable, where* $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ *or* $\mathbf{s} \xleftarrow{\$} \chi^n$, $\mathbf{e} \xleftarrow{\$} \chi^m$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m$, *and* $m = poly(n)$.

Definition 2 (B-bounded Noise Distribution)**.** *If* $\Pr_{x\leftarrow\chi}[|x| \geq B] \leq 2^{-\tilde{\Omega}(n)}$, *then* χ *over* $\mathbb{Z}$ *is* B*-bounded.*

Definition 3. *The TrapGen algorithm [17,18] takes as input* 1^n, m, q *where* n, m, q *be integers and* $q \geq 3$ *be odd and* $m = \lceil 6n \log q \rceil$. *It generates* $(\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{T_A} \in \mathbb{Z}^{m\times m})$. *It is ensured that* $\|\widetilde{\mathbf{T_A}}\| \leq O(\sqrt{n \log q})$ *with overwhelmingly high probability in* n.

Definition 4. *The SamplePre algorithm [19] algorithm takes as input* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, *a basis* $\mathbf{T_A}$ *for lattice* $\Lambda_q^{\perp}(\mathbf{A})$, $\mathbf{u} \in \mathbb{Z}_q^n$, *and a Gaussian parameter* $\tau \geq \|\widetilde{\mathbf{T_A}}\|\omega(\sqrt{\log m})$. *It outputs a vector* $\mathbf{e} \in \mathbb{Z}^m$ *sampled from a distribution that is* $2^{-\Omega(n)}$*-close to* $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\tau}$.

In the subsequent construction of the scheme in this paper, we use the following two functions of key switching techinique [20].

Definition 5. *There are two deterministic functions that map vectors to a higher dimension,* $\boldsymbol{BD}(\mathbf{v})$ *and* $\boldsymbol{P2}(\mathbf{x})$, *respectively.*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

First, let us review the $\boldsymbol{BD}(\mathbf{v})$ *function. For a vector* $\mathbf{v} \in \mathbb{Z}_q^n$*, let* $\mathbf{v}_i \in \{0,1\}^n$ *be* $\mathbf{v} = \sum_{i=0}^{\lceil \log q \rceil - 1} 2^i \mathbf{v}_i$*,* $\boldsymbol{BD}(\mathbf{v})$ *inputs a vector* $\mathbf{v}$ *and outputs a vector* $\widetilde{\mathbf{v}} = (\mathbf{v}_0; \cdots; \mathbf{v}_{\lceil \log q \rceil - 1}) \in \{0,1\}^{n\lceil \log q \rceil}$*.*

The second function $\boldsymbol{P2}(\mathbf{x})$ *inputs vector* $\mathbf{x} \in \mathbb{Z}_q^n$ *and outputs* $\bar{\mathbf{x}} = (\mathbf{x}; 2\mathbf{x}; \cdots; 2^{\lceil \log q \rceil - 1}\mathbf{x}) \in \mathbb{Z}_q^{n\lceil \log q \rceil}$*.*

These two functions hold that $\mathbf{v}^T\mathbf{x} = \boldsymbol{BD}(\mathbf{v})^T \cdot \boldsymbol{P2}(\mathbf{x}) = \widetilde{\mathbf{v}}^T\bar{\mathbf{x}}$*.*

A very important lemma, named Leftover Hash Lemma, needs to be used in the proof of schemes.

Definition 6. *Leftover Hash Lemma [21] refers to two distributions* $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^T\mathbf{e})$ *and* $(\mathbf{A}, \mathbf{B}, \mathbf{R}^T\mathbf{e})$*, which are statistically close for all* $\mathbf{e} \in \mathbb{Z}_q^m$ *if* $m > (n+1)\log_2 q + \omega(\log n)$*,* $q > 2$*, and* $\mathbf{R} \in \{1, -1\}^{m \times k}$ *where* $k = k(n)$*. Matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *are chosen uniformly from* $\mathbb{Z}^{n \times m}$ *and* $\mathbb{Z}^{n \times k}$*, respectively.*

3. Re-encryption Simulatability of PRE

In this part, we define Proxy Re-Encryption (PRE) and elucidate the concept of re-encryption simulatability. Six algorithms make up a PRE scheme, as listed below:

$\boldsymbol{Setup}(1^\lambda) \to pp$, $\boldsymbol{KeyGen}(pp, i) \to (pk_i, sk_i)$, $\boldsymbol{Enc}(pk_i, \mu) \to c_i$, $\boldsymbol{Dec}(sk_i, c_i) \to \mu$, $\boldsymbol{ReKeyGen}(sk_i, pk_j) \to rk_{i\to j}$, $\boldsymbol{ReEnc}(rk_{i\to j}, c_i) \to c_j$.

In the above algorithms, 1^λ and pp represent security parameter and public parameters; i, pk_i and sk_i stand for a user's identity, public key and secret key of user i; μ represents a message; c_i represents a ciphertext under pk_i and $rk_{i\to j}$ represents the re-encryption key.

Correctness. The correctness includes the following two validations

- **Original Ciphertext.** It satisfies the equation $\boldsymbol{Dec}(sk_i, \boldsymbol{Enc}(sk_i, \mu)) = \mu$.
- **Re-encryption Ciphertext.** It satisfies $\boldsymbol{Dec} = (sk_j, \boldsymbol{ReEnc}(rk_{i\to j}, c_i)) = \mu$.

A detailed description of the widely used CPA security model of proxy re-encryption can be found in reference [22]. This paper directly describe the HRA security model [11] for PRE, which implies the CPA security model. Re-encryption simulatability plays an essential role in proving the PRE scheme's HRA security, which was first proposed by Cohen et al. [11] at PKC'19. In a nutshell, re-encryption simulatability is the ability to simulate ciphertexts produced by computing $\boldsymbol{ReEnc}(rk_{i\to j}, c_i)$ without knowing the secret key sk_i of the sender (but knowing the plaintext message μ and the secret key sk_j of the recipient). Furthermore, Cohen et al. also put forth the following crucial theorem regarding the re-encryption simulatability. Let's take a detailed look below.

Definition 7. *A PRE scheme possesses the re-encryption simulatability property[11] if there is a PPT algorithm* $\boldsymbol{Sim.ReEnc}$ *that satisfies the condition* $(\boldsymbol{Sim.ReEnc}(aux), aux)$ *is statistically indistinguishable from* $(\boldsymbol{ReEnc}(rk_{i\to j}, c_i), aux)$*.* c_i *and* aux *are sampled according to*

$$\boldsymbol{Setup}(1^\lambda) \to pp;$$

$$\boldsymbol{KeyGen}(pp, i) \to (pk_i, sk_i);$$

$$\boldsymbol{KeyGen}(pp, j) \to (pk_j, sk_j);$$

$$\boldsymbol{ReKeyGen}(pp, sk_i, pk_j) \to rk_{i\to j};$$

$$\boldsymbol{Enc}(pp, pk_i, \mu) \to c_i;$$

$$aux = (pp, pk_i, pk_j, sk_j, c_i, \mu).$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

It should be noted that a special case arises when $\textbf{\textit{Sim.ReEnc}}(aux) = \textbf{\textit{Enc}}(pk_j, \mu)$. *This indicates that the distribution of the ciphertext after re-encryption is the same as that of the original ciphertext, then the scheme possesses re-encryption simulatability property.*

Lemma 1 ([11]). *If a PRE scheme with security under CPA possesses re-encryption simulatability property, this PRE scheme is HRA-secure.*

4. Construction of PRE with Re-encryption Simulatability

Now, we introduce our innovative lattice-based Proxy Re-Encryption (PRE) scheme, designed to uphold re-encryption simulatability without bootstrapping. We integrate the concept of key switching to develop a novel re-encryption key generation algorithm and re-encryption algorithm. The resulting ciphertexts after re-encryption, maintain a distribution nearly identical to that of the original ciphertexts. Consequently, in conjunction with the Chosen-Plaintext Attack (CPA) security of the foundational dual-Regev encryption, our novel PRE scheme achieves security against Honest Re-Encryption Attacks (HRA). The detailed construction and the corresponding security analysis are provided below. Our innovation focuses on the ***ReKeyGen*** and ***ReEnc*** algorithms.

4.1. Construction (Scheme I)

Prior to presenting our PRE scheme, we provide a list of the parameters employed in the scheme.

- 1^λ-security parameter.
- τ-discrete Gaussian distribution $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\tau}$ parameter, where τ is equal to $\omega((m+1)^{d+1}) \cdot \omega(\sqrt{\log m})$, larger than $\omega(\sqrt{\log m})$.
- (n, m, q, χ)-lattice parameter, where $m \geq \lceil 6n \lg q \rceil$, $\frac{q}{4} \geq B \cdot (m+1)^{O(d)}$, and χ is a B-bounded distribution.

The following describes lattice-based PRE scheme with re-encryption simulatability property:

- $\textbf{\textit{Setup}}(1^\lambda) \to pp$: This algorithm inputs 1^λ and outputs $pp = (n, m, q, \chi, \chi^m)$.
- $\textbf{\textit{KeyGen}}(pp, i) \to (pk_i, sk_i)$: This algorithm selects a random vector $\mathbf{U}_i \leftarrow \mathbb{Z}_q^{n \times m}$ and generates $(\mathbf{A}_i, \mathbf{T}_{\mathbf{A}_i})$ by executing $\textbf{\textit{TrapGen}}(1^n, m, q)$ for every user i where $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{T}_{\mathbf{A}_i} \in \mathbb{Z}_q^{m \times m}$, a basis of $\Lambda_q^\perp(\mathbf{A}_i)$. Then, it computes $\mathbf{E}_i \in \chi^{m \times m}$ by running $\mathbf{E}_i \leftarrow \textbf{\textit{SamplePre}}(\mathbf{A}_i, \mathbf{T}_{\mathbf{A}_i}, \mathbf{U}_i, \tau)$, where $\mathbf{U}_i = \mathbf{A}_i \mathbf{E}_i \in \mathbb{Z}_q^{n \times m}$. This algorithm outputs $pk_i = \{\mathbf{U}_i, \mathbf{A}_i\}$ and $sk_i = \mathbf{E}_i$.
- $\textbf{\textit{Enc}}(pk_i, \mu) \to c_i$: This algorithm selects $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ to encrypt $\mu \in \{0,1\}^m$. Let $pk_i = \{\mathbf{U}_i, \mathbf{A}_i\}$, and set $\mathbf{c}_{i_0} = \mathbf{s}^T \mathbf{A}_i + \mathbf{x}_0^T \in \mathbb{Z}_q^{1 \times m}$, $\mathbf{c}_{i_1} = \mathbf{s}^T \mathbf{U}_i + \mathbf{x}_1^T + \mu \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^{1 \times m}$, where Gaussian noise vectors $\mathbf{x}_0 \xleftarrow{\$} \chi^m$ and $\mathbf{x}_1 \xleftarrow{\$} \chi^m$. Finally, this algorithm outputs $c_i = (\mathbf{c}_{i_0}, \mathbf{c}_{i_1})$.
- $\textbf{\textit{Dec}}(sk_i, c_i) \to \mu / \perp$: Given $c_i = (\mathbf{c}_{i_0}, \mathbf{c}_{i_1})$ and $sk_i = \mathbf{E}_i$. This algorithm computes $\mu' = \mathbf{c}_{i_1} - \mathbf{c}_{i_0} \mathbf{E}_i$. For $j \in [m]$, the algorithm sets $\mu_j = 0$ if μ'_j is closer to 0 than to $\frac{q}{2}$ modulo q; otherwise outputs $\mu_j = 1$. Finally, this algorithm outputs $\mu \in \{0,1\}^m$.
- $\textbf{\textit{ReKeyGen}}(sk_i, pk_j) \to rk_{i \to j}$: On input $sk_i = \mathbf{E}_i$ and $pk_j = (\mathbf{U}_j, \mathbf{A}_j)$. The algorithm chooses matrices $\mathbf{R}_1 \xleftarrow{\$} \mathbb{Z}_q^{m\lceil \log q \rceil \times n}$, $\mathbf{R}_2 \xleftarrow{\$} \chi^{m\lceil \log q \rceil \times m}$, vectors $\mathbf{R}_3 \xleftarrow{\$} \chi^{m\lceil \log q \rceil \times m}$. Then, it computes

$$\mathbf{Z} = \begin{pmatrix} \mathbf{R}_1 \mathbf{A}_j + \mathbf{R}_2 & \mathbf{R}_1 \mathbf{U}_j + \mathbf{R}_3 - \boldsymbol{P2}(\mathbf{E}_i) \\ \mathbf{0}_{1 \times m} & 1 \end{pmatrix} \in \mathbb{Z}_q^{(m\lceil \log q \rceil + 1) \times 2m}.$$

The general key generation algorithm of PRE is only composed of $\mathbf{Z}$, but in this study, we innovatively introduce $\mathbf{g}^T$. The purpose of introducing is to make the scheme have the re-encryption simulatability. We define $\mathbf{g}^T$ as

$$\mathbf{g}^T = \mathbf{r}_1^T (\mathbf{A}_j \| \mathbf{U}_j) + (\tilde{\mathbf{e}}_0^T \| \tilde{\mathbf{e}}_1^T) \in \mathbb{Z}_q^{1 \times 2m},$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

where $\mathbf{r}_1 \xleftarrow{\$} \mathbb{Z}_q^n$, $\tilde{\mathbf{e}}_0 \xleftarrow{\$} \chi^m$, and $\tilde{\mathbf{e}}_1 \xleftarrow{\$} \chi^m$. Finally, it outputs $rk_{i\to j} = \{\mathbf{g}^T, \mathbf{Z}\}$.

– ***ReEnc***$(rk_{i\to j}, c_i) \to c_j$: Given $rk_{i\to j} = \{\mathbf{g}^T, \mathbf{Z}\}$ and $c_i = (\mathbf{c}_{i_0}, \mathbf{c}_{i_1})$. This algorithm consists of three steps.

First, sample a small random number $a \in \chi$ and compute

$$(\bar{\mathbf{c}}_{j_0}, \bar{\mathbf{c}}_{j_1}) = a\mathbf{g}^T = a(\mathbf{r}_1^T(\mathbf{A}_j \| \mathbf{U}_j) + (\tilde{\mathbf{e}}_0^T \| \tilde{\mathbf{e}}_1^T)).$$

Since a is random, $\bar{\mathbf{c}}_{j_0}$ and $\bar{\mathbf{c}}_{j_1}$ are also random.

Second, calculate

$$\begin{aligned}(\mathbf{c}'_{j_0}, \mathbf{c}'_{j_1}) &= (\boldsymbol{BD}(\mathbf{c}_{i_0}) \| \mathbf{c}_{i_1}) \cdot \mathbf{Z} \\ &= (\boldsymbol{BD}(\mathbf{c}_{i_0}) \| \mathbf{c}_{i_1}) \begin{pmatrix} \mathbf{R}_1\mathbf{A}_j + \mathbf{R}_2 & \mathbf{R}_1\mathbf{U}_j + \mathbf{R}_3 - \boldsymbol{P2}(\mathbf{E}_i) \\ \mathbf{0}_{1\times m} & 1 \end{pmatrix}.\end{aligned}$$

The $\mathbf{c}'_{j_0}$ and c'_{j_1} are determined for a specific ciphertext c_i and for a specific $rk_{i\to j}$.
Third, obtain the random re-encryption ciphertext as follows:

$$\mathbf{c}_{j_0} = \bar{\mathbf{c}}_{j_0} + \mathbf{c}'_{j_0},$$

and

$$\mathbf{c}_{j_1} = \bar{\mathbf{c}}_{j_1} + \mathbf{c}'_{j_1}.$$

Let can simplify the above steps as follows

$$c_j = a\mathbf{g}^T + (\boldsymbol{BD}(\mathbf{c}_{i_0}) \| \mathbf{c}_{i_1}) \cdot \mathbf{Z}.$$

Finally, this algorithm outputs $c_j = (\mathbf{c}_{j_0}, \mathbf{c}_{j_1})$ as re-encryption ciphertext.

Correctness. Based on the provided parameters, the correctness of the above PRE summarized as follows.

- **Original Ciphertext.** $c_i = (\mathbf{c}_{i_0}, \mathbf{c}_{i_1})$ is the ciphertext of μ under pk_i. $\mathbf{c}_{i_0} = \mathbf{s}^T\mathbf{A}_i + \mathbf{x}_0^T \in \mathbb{Z}_q^{1\times m}$, $\mathbf{c}_{i_1} = \mathbf{s}^T\mathbf{U}_i + \mathbf{x}_1^T + \mu\lfloor\frac{q}{2}\rfloor \in \mathbf{Z}_q^{1m}$. Therefore, we have the decryption as below.

$$\begin{aligned}\mathbf{c}_{i_1} - \mathbf{c}_{i_0}\mathbf{E}_i &= \mathbf{s}^T\mathbf{U}_i + \mathbf{x}_1^T + \mu\lfloor\frac{q}{2}\rfloor - (\mathbf{s}^T\mathbf{A}_i + \mathbf{x}_0^T)\mathbf{E}_i \\ &= \mathbf{s}^T\mathbf{U}_i + \mathbf{x}_1^T + \mu\lfloor\frac{q}{2}\rfloor - \mathbf{s}^T\mathbf{A}_i\mathbf{E}_i - \mathbf{x}_0^T\mathbf{E}_i \\ &= \mu\lfloor\frac{q}{2}\rfloor + \underbrace{\mathbf{x}_1^T - \mathbf{x}_0^T\mathbf{E}_i}_{\text{error}\quad\text{term}}.\end{aligned}$$

In order to obtain an accurate decryption, the error term norm needs to be smaller than $q/4$, i.e., $\|\mathbf{x}_1^T - \mathbf{x}_0^T\mathbf{E}_i\| \le q/4$. Because of $\mathbf{x}_0 \xleftarrow{\$} \chi^m$ and $\mathbf{x}_1 \xleftarrow{\$} \chi^m$, $\mathbf{E}_i \in \chi^{m\times m}$ and $\mathbf{A}_i\mathbf{E}_i = \mathbf{U}_i$, we have $\|\mathbf{x}_0^T\| \le \sqrt{m}B$, $\|\mathbf{x}_1^T\| \le \sqrt{m}B$, and $\|\mathbf{E}_i\| \le m\tau$. Then, we can compute $\|\mathbf{x}_1^T - \mathbf{x}_0^T\mathbf{E}_i\| \le \|\mathbf{x}_1^T + \mathbf{x}_0^T\mathbf{E}_i\| \le \|\mathbf{x}_1^T\| + \|\mathbf{x}_0^T\mathbf{E}_i\| \le \|\mathbf{x}_1^T\| + \|\mathbf{x}_0^T\|\|\mathbf{E}_i\| \le \sqrt{m}B + \sqrt{m}B \cdot m\tau \le \sqrt{m}B + m\sqrt{m}\tau B \le B \cdot (m+1)^{O(d)} \le q/4$. Therefore, the initial ciphertext can be decrypted correctly.
- **Re-encryption Ciphertext.** The re-encrypted ciphertext represented as $c_j = (\mathbf{c}_{j_0}, \mathbf{c}_{j_1})$ can be computed as follows.

$$\begin{aligned}c_j &= a\mathbf{g}^T + (\boldsymbol{BD}(\mathbf{c}_{i_0}) \| \mathbf{c}_{i_1}) \cdot \mathbf{Z} \\ &= a\mathbf{r}_1^T(\mathbf{A}_j \| \mathbf{U}_j) + a(\tilde{\mathbf{e}}_0^T \| \tilde{\mathbf{e}}_1^T) + (\boldsymbol{BD}(\mathbf{c}_{i_0}) \| \mathbf{c}_{i_1}) \begin{pmatrix} \mathbf{R}_1\mathbf{A}_j + \mathbf{R}_2 & \mathbf{R}_1\mathbf{U}_j + \mathbf{R}_3 - \boldsymbol{P2}(\mathbf{E}_i) \\ \mathbf{0}_{1\times m} & 1 \end{pmatrix}.\end{aligned}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

Therefore, we have

$$\begin{cases} \mathbf{c}_{j_0} = (a\mathbf{r}_1^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_1)\mathbf{A}_j + a\tilde{\mathbf{e}}_0^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_2, \\ \mathbf{c}_{j_1} = (a\mathbf{r}_1^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_1)\mathbf{U}_j + a\tilde{\mathbf{e}}_1^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_3 - \mathbf{x}_0^T\mathbf{E}_i + \mathbf{x}_1^T + \mu\lfloor\frac{q}{2}\rfloor. \end{cases} \tag{1}$$

Let

$$\bar{\mathbf{s}}^{\mathbf{T}} = a\mathbf{r}_1^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_1 \in \mathbb{Z}_q^{1\times n},$$
$$\bar{\mathbf{x}}_0^T = a\tilde{\mathbf{e}}_0^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_2 \in \chi^{1\times m},$$
$$\bar{\mathbf{x}}_1^T = a\tilde{\mathbf{e}}_1^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_3 - \mathbf{x}_0^T\mathbf{E}_i + \mathbf{x}_1^T \in \chi^{1\times m}.$$

Equation (1) can be simplified to Equation (2) as follows:

$$\begin{cases} \mathbf{c}_{j_0} = \bar{\mathbf{s}}^T\mathbf{A}_j + \bar{\mathbf{x}}_0^T, \\ \mathbf{c}_{j_1} = \bar{\mathbf{s}}^T\mathbf{U}_j + \bar{\mathbf{x}}_1^T + \mu\lfloor\frac{q}{2}\rfloor. \end{cases} \tag{2}$$

Therefore, we have the decryption as below.

$$\begin{aligned} \mathbf{c}_{j_1} - \mathbf{c}_{j_0}\mathbf{E}_j &= \bar{\mathbf{s}}^T\mathbf{U}_j + \bar{\mathbf{x}}_1^T + \mu\lfloor\frac{q}{2}\rfloor - (\bar{\mathbf{s}}^T\mathbf{A}_j + \bar{\mathbf{x}}_0^T)\mathbf{E}_j \\ &= \mu\lfloor\frac{q}{2}\rfloor + \underbrace{\bar{\mathbf{x}}_1^T - \bar{\mathbf{x}}_0^T\mathbf{E}_j}_{\text{error}\quad\text{term}}. \end{aligned}$$

In order to obtain an accurate decryption, the error term norm needs to be smaller than $q/4$, i.e., $\|\bar{\mathbf{x}}_1^T - \bar{\mathbf{x}}_0^T\mathbf{E}_j\| \leq q/4$. Because of $\bar{\mathbf{x}}_0^T \in \chi^{1\times m}$, $\bar{\mathbf{x}}_1^T \in \chi^{1\times m}$, $\mathbf{E}_j \in \chi^{m\times m}$ and $\mathbf{A}_j\mathbf{E}_j = \mathbf{U}_j$, we have $\|\mathbf{x}_0^T\| \leq \sqrt{m}B$, $\|\mathbf{x}_1^T\| \leq \sqrt{m}B$, and $\|\mathbf{E}_j\| \leq m\tau$. Similar to the first case, it is not difficult to compute $\|\bar{\mathbf{x}}_1^T - \bar{\mathbf{x}}_0^T\mathbf{E}_j\| \leq q/4$. As a result, the re-encryption ciphertext can be correctly decrypt.

4.2. Security Proof of Scheme I

According to Lemma 1, we now show that the above scheme satisfies CPA security and re-encryption simulatability respectively.

Theorem 1. *The scheme I has re-encryption simulatability property.*

Proof of Theorem 1. Referring to Equation 1 and Equation 2, when μ is known and $\bar{\mathbf{s}}$, $\bar{\mathbf{x}}_0$, and $\bar{\mathbf{x}}_1$ are randomized, we can directly sample to get $(\bar{\mathbf{s}}^T\mathbf{A}_j + \bar{\mathbf{x}}_0^T, \bar{\mathbf{s}}^T\mathbf{U}_j + \bar{\mathbf{x}}_1^T + \mu\lfloor\frac{q}{2}\rfloor)$. It is observed that the distributions of ciphertexts $\mathbf{c}_j$ and $\mathbf{c}_i$ are identical. Assuming that we know pp, $pk_j = \{\mathbf{U}_j, \mathbf{A}_j\}$ and plaintext μ, we can easily sample these ciphertexts as follows.

– ***Sim.ReEnc***$(pp, pk_j, sk_j, c_i, \mu) \rightarrow c_j$: On input $pp = (n, m, q, \chi, \chi^m)$, $pk_j = \{\mathbf{U}_j, \mathbf{A}_j\}$, $sk_j = \mathbf{E}_j$, $c_i = (\mathbf{c}_{i_0}, \mathbf{c}_{i_1})$. Sample $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{x}_0' \xleftarrow{\$} \chi^m$ and $\mathbf{x}_1' \xleftarrow{\$} \chi^m$. This algorithm outputs

$$\mathbf{c}'_{j_0 sim} = \mathbf{s}'^T\mathbf{A}_j + \mathbf{x}_0'^T,$$

and

$$\mathbf{c}'_{j_1 sim} = \mathbf{s}'^T\mathbf{U}_j + \mathbf{x}_1'^T + \mu\lfloor\frac{q}{2}\rfloor.$$

Because the re-encryption ciphertexts obtained by re-encryption algorithm are

$$\mathbf{c}_{j_0} = (a\mathbf{r}_1^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_1)\mathbf{A}_j + a\tilde{\mathbf{e}}_0^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_2 = \bar{\mathbf{s}}^T\mathbf{A}_j + \bar{\mathbf{x}}_0^T,$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

and

$$\mathbf{c}_{j_1} = (a\mathbf{r}_1^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_1)\mathbf{U}_j + a\tilde{\mathbf{e}}_1^T + \boldsymbol{BD}(\mathbf{c}_{i_0})\mathbf{R}_3 - \mathbf{x}_0^T\mathbf{E}_i + \bar{\mathbf{x}}_1^T + \mu\lfloor\frac{q}{2}\rfloor = \bar{\mathbf{s}}^T\mathbf{U}_j + \bar{\mathbf{x}}_1^T + \mu\lfloor\frac{q}{2}\rfloor.$$

From the above, we can see that $\mathbf{s}'$, $\mathbf{x}_0'$ and $\mathbf{x}_1'$ have the same distribution as $\bar{\mathbf{s}}$, $\bar{\mathbf{x}}_0$and $\bar{\mathbf{x}}_1$, respectively. Therefore, it is not difficult for us to find that $\mathbf{c}'_{j_0 sim}$ and $\mathbf{c}'_{j_1 sim}$ are indistinguishable from $\mathbf{c}_{j_0}$ and $\mathbf{c}_{j_1}$, respectively. Meanwhile, the ciphertext $c_j = (\mathbf{c}'_{j_0 sim}, \mathbf{c}'_{j_1 sim})$ can be decrypted by sk_j.

Hence, the PRE scheme above has the property of re-encryption simulatability. □

Theorem 2. *The scheme I we proposed is selective HRA-secure under the hardness of LWE.*

Proof of Theorem 2. The selective CPA security of this scheme is easy to prove. Due to space constraints, we do not provide detailed proof of the CPA in this study.

According to Lemma 1, the PRE scheme we proposed above can be proved to be HRA-secure under the hardness of LWE. □

5. Construction of AB-CPRE with Re-Encryption Simulatability

AB-CPRE was initially introduced by Liang et al. in [14] at ESORICS'21. They provided a comprehensive definition of AB-CPRE, elaborated on its CPA model, meticulously constructed the corresponding scheme utilizing LWE, and presented rigorous proof of its selective CPA security. In this section, we use the construction idea in scheme I to improve the ***ReKeyGen*** and ***ReEnc*** of [14], enabling the modified scheme to have the property of re-encryption simulatability. The modified AB-CPRE scheme enhances the security from selective CPA to selective HRA. Our focuses are on the modified scheme with re-encryption simulatability and the implementation of the selective HRA security proof. The other four algorithms remain the same as those in [14].

5.1. HRA Security Model of AB-CPRE

We directly formalize the selective HRA security model of AB-CPRE for the first time, which is demonstrated below. Unlike Scheme I, in the following scheme, we use α and β to represent users's identities. (Since letters such as f and g represent strategies, it would be confusing to use letters i and j to represent identities again.) Note: The condition $f(\mathbf{x}) = 0$ indicates that the attributes encoded in the ***Enc*** algorithm satisfy the policy embedded in the ***ReKeyGen*** algorithm.

Definition 8 (Security Model for Selective HRA of AB-CPRE)**.** ***Honest Key Generation*** $\mathcal{O}_{Honest}$, ***Corrupted Key Generation*** $\mathcal{O}_{Corrupted}$ *and* ***Re-encryption Key Generation*** $\mathcal{O}_{ReKeyGen}$ *are the same as the security game for selective CPA [14]. Besides,* $\mathcal{O}_{Enc}$ *has been added. Additionally, the* $\mathcal{O}_{ReEnc}$ *and the* ***challenge phase*** *also differ from the CPA security model.*
Init*: $\mathcal{A}$ announces challenge user θ^* and the challenge attributes vector $\mathbf{x}^*$.*
Setup*: The challenger $\mathcal{C}$ runs* $\boldsymbol{Setup}(1^\lambda)$ *to generate pp and gives it to $\mathcal{A}$. Two sets, Γ_H (representing honest) and Γ_C (representing corrupted), are initially empty by $\mathcal{C}$. These operation above performs the same as the challenger $\mathcal{C}$ in the CPA security model of AB-CPRE. Besides, $\mathcal{C}$ initializes a counter* **numCt** *to 0, an empty key-value store* **Ct**, *and an empty set* **Derive**. *Besides, the set Γ_{rk} (representing re-encryption key) is initially empty.*
Query Phase 1*: In this Phase, $\mathcal{A}$ mainly makes four types of queries:*

- ***Honest Key Generation*** $\mathcal{O}_{Honest}$*: First, $\mathcal{C}$ obtains a key pair by running* $(pk_\alpha, sk_\alpha) \leftarrow \boldsymbol{KeyGen}(pp, \alpha)$ *after $\mathcal{A}$ sends the identity of a user α. Then, $\mathcal{C}$ give pk_α to $\mathcal{A}$. Finally, $\mathcal{C}$ inserts the identity α into the set Γ_H.*
- ***Corrupted Key Generation*** $\mathcal{O}_{Corrupted}$*: First, $\mathcal{C}$ obtains a key pair by running* $(pk_\alpha, sk_\alpha) \leftarrow \boldsymbol{KeyGen}(pp, \alpha)$ *after $\mathcal{A}$ sends the identity of a user α. Then, $\mathcal{C}$ give (pk_α, sk_α) to $\mathcal{A}$. Finally, $\mathcal{C}$ inserts the identity α into the set Γ_C.*

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

- ***Re-encryption Key Generation*** $\mathcal{O}_{ReKeyGen}$: *Given* α, β *and* f *by* $\mathcal{A}$, $\mathcal{C}$ *inputs* $\perp$ *if* $\alpha = \theta^*$, $\beta \in \Gamma_C$, *and satisfying* $f(\mathbf{x}^*) = 0$. *Otherwise,* $\mathcal{C}$ *returns the* $rk_{\alpha,f\to\beta}$ *by running the algorithm* $\boldsymbol{ReKeyGen}(sk_\alpha, pk_\beta, f) \to rk_{\alpha,f\to\beta}$, *inserts* $rk_{\alpha,f\to\beta}$ *into* Γ_{rk} *with the key-value* $(\alpha, \beta, f, rk_{\alpha,f\to\beta})$, *and outputs* $rk_{\alpha,f\to\beta}$.
- ***Encryption*** $\mathcal{O}_{Enc}$: *Given* pk_α, $\mu \in \{0,1\}^m$, $\mathbf{x} = \{x_i\}_{i\in[l]}$ *by* $\mathcal{A}$, $\mathcal{C}$ *obtains the ciphertext* $c_{\alpha,\mathbf{x}}$ *by running algorithm* $\boldsymbol{Enc}(pk_\alpha, \mu, \mathbf{x}) \to c_{\alpha,\mathbf{x}}$ *and increases* $\mathbf{numCt}$. *Then,* $\mathcal{C}$ *stores the value* $c_{\alpha,\mathbf{x}}$ *in* $\mathbf{Ct}$ *with key* $(\alpha, \mathbf{numCt})$ *and returns* $(\mathbf{numCt}, c_{\alpha,\mathbf{x}})$.
- ***Re-encryption*** $\mathcal{O}_{ReEnc}$: *Given* $c_{\alpha,\mathbf{x}}$, α, β, f *and* k *by* $\mathcal{A}$, *where* $k \leq \mathbf{numCt}$, $\mathcal{C}$ *returns* $\perp$ *if there is no value in* $\mathbf{Ct}$ *with key* (α, k) *or when* $f(\mathbf{x}) \neq 0$ *holds or when* $\alpha = \theta^*$, $\beta \in \Gamma_C$, $f(\mathbf{x}^*) = 0$, $k \in \mathbf{Derive}$. *Otherwise,* $\mathcal{C}$ *gets* $rk_{\alpha,f\to\beta}$ *by searching* Γ_{rk} *set or queries* $\mathcal{O}_{RekeyGen}$. *Then,* $\mathcal{C}$ *runs* $\boldsymbol{ReEnc}(rk_{\alpha,f\to\beta}, c_{\alpha,\mathbf{x}}) \to c_\beta$, *increases* $\mathbf{numCt}$, *and stores the value* c_β *in* $\mathbf{Ct}$ *with key* $(\beta, \mathbf{numCt})$. *Finally,* $\mathcal{C}$ *returns* $(\mathbf{numCt}, c_\beta)$.

Challenge Phase*: After receiving* $(\theta^*, \mu_0, \mu_1, \mathbf{x}^*)$ *from* $\mathcal{A}$, $\mathcal{C}$ *selects* $b \xleftarrow{\$} \{0,1\}$ *and obtains the challenge ciphertext* $c_{\theta^*,\mathbf{x}^*}$ *by running the algorithm* $\boldsymbol{Enc}(pp, pk_{\theta^*}, \mu_b, \mathbf{x}^*)$. *Additionally,* $\mathcal{C}$ *increments* $\mathbf{numCt}$ *and adds it to the set* $\mathbf{Deriv}$. *The value* $c_{\theta^*,\mathbf{x}^*}$ *is stored in* $\mathbf{Ct}$ *with key* $(\theta^*, \mathbf{numCt})$, *and finally* $(\mathbf{numCt}, c_{\theta^*,\mathbf{x}^*})$ *is returned.*
Query Phase 2*: This phase is identical to* ***Query Phase 1****, with the exception of the* $\mathcal{O}_{ReEnc}$ *oracle.* $\mathcal{C}$ *outputs* $\perp$ *if* $\beta \in \Gamma_C$ *and* $k \in \mathbf{Derive}$.
Decision Phase*:* $\mathcal{A}$ *produces a decision* $b' \in \{0,1\}$. *Eventually,* $\mathcal{A}$ *is declared the winner of the game if and only if* $b' = b$.

5.2. Succinct Construction (Scheme II)

To facilitate our proofs, we elaborate on the entire scheme in this section. Notably, our primary innovations manifest in the last two algorithms. The initial four algorithms closely mirror those delineated in [14]. To avoid redundancy, we abstain from revisiting specific elements of previous knowledge integral to the scheme, such as the Gadget matrix $\mathbf{G}$, $\boldsymbol{ExtendLeft}$, $\boldsymbol{ExtendRight}$, $\boldsymbol{Eval}_{ct}$, $\boldsymbol{Eval}_{pk}$, $\boldsymbol{Eval}_{sim}$, as these have been extensively covered in reference reference [14].

- $\boldsymbol{Setup}(n) \to pp$: $\mathbf{B}_1, \cdots, \mathbf{B}_l \leftarrow \mathbb{Z}_q^{n\times m}$. Output $pp = (\mathbf{B}_1, \cdots, \mathbf{B}_l, \chi)$, where χ is an error sampling algorithm.
- $\boldsymbol{KeyGen}(pp, \alpha) \to (pk_\alpha, sk_\alpha)$: The following algorithms $(\mathbf{A}_\alpha, \mathbf{T}_{\mathbf{A}_\alpha}) \leftarrow \boldsymbol{TrapGen}(n, 1^m, q)$, and $\mathbf{R}_\alpha \leftarrow \boldsymbol{SamplePre}(\mathbf{A}_\alpha, \mathbf{T}_{\mathbf{A}_\alpha}, -\mathbf{D}_\alpha, \sigma)$. are executed. Then, this algorithm outputs $pk_\alpha = (\mathbf{A}_\alpha, \mathbf{D}_\alpha)$ and $sk_\alpha = (\mathbf{T}_{\mathbf{A}_\alpha}, \mathbf{R}_\alpha)$, where $\mathbf{D}_\alpha \leftarrow \mathbb{Z}_q^{n\times m}$.
- $\boldsymbol{Enc}(pp, pk_\alpha, \mu \in \{0,1\}^m, \mathbf{x} = \{x_i\}_{i\in[l]}) \to c_{\alpha,\mathbf{x}}$: Let $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_{in}, \mathbf{e}_{out} \in \chi^m$. Then $ct_{\alpha,\mathbf{x}} = (\mathbf{c}_{in}, \mathbf{c}_{out}) = (\mathbf{A}_\alpha^T\mathbf{s} + \mathbf{e}_{in}, \mathbf{D}_\alpha^T\mathbf{s} + \mathbf{e}_{out} + \mu\lfloor\frac{q}{2}\rfloor)$. If $\mathbf{x}$ is null, then set $cc = \emptyset$. Otherwise, $cc_{\alpha,\mathbf{x}} = (\{\mathbf{c}_i = (x_i\mathbf{G} + \mathbf{B}_i)^T\mathbf{s} + \mathbf{S}_i^T\mathbf{e}_{in}\}_{i\in[l]}) \in \mathbb{Z}_q^{lm}$, where $\mathbf{S}_i \leftarrow \{-1,1\}^{m\times m}$. Output $c_{\alpha,\mathbf{x}} = (ct_{\alpha,\mathbf{x}}, cc_{\alpha,\mathbf{x}})$.
- $\boldsymbol{Dec}(pp, sk_\alpha, c_{\alpha,\mathbf{x}}) \to \mu'$: Compute

$$\mu' = (\mathbf{c}_{in}^T\mathbf{c}_{out}^T)\begin{pmatrix}\mathbf{R}_\alpha\\ \mathbf{I}_{m\times m}\end{pmatrix}.$$

  For $j \in [m]$, set $\mu_j = 1$ if $\mu'_j - \lfloor\frac{q}{2}\rfloor < q/4$, otherwise set $\mu_j = 0$. Finally, output $\mu \in \{0,1\}^m$.
- $\boldsymbol{ReKeyGen}(pp, sk_\alpha, pk_\beta, f) \to rk_{\alpha,f\to\beta}$: The inputs of this algorithm are $pp = (\mathbf{B}_1, \cdots, \mathbf{B}_l, \chi)$, $sk_\alpha = (\mathbf{T}_{\mathbf{A}_\alpha}, \mathbf{R}_\alpha)$, $pk_\beta = (\mathbf{A}_\beta, \mathbf{D}_\beta)$ and a policy f. Then this algorithm executes the following steps one by one.

$$\mathbf{B}_f = \boldsymbol{Eval}_{pk}(f, \{\mathbf{B}_i\}_{i\in[l]}),$$

$$\mathbf{T}_{(\mathbf{A}_\alpha|\mathbf{B}_f)} \leftarrow \boldsymbol{ExtendRight}(\mathbf{A}_\alpha, \mathbf{T}_{\mathbf{A}_\alpha}, \mathbf{B}_f),$$

$$\mathbf{R}_{\alpha,f} \leftarrow \boldsymbol{SamplePre}((\mathbf{A}_\alpha|\mathbf{B}_f), \mathbf{T}_{(\mathbf{A}_\alpha|\mathbf{B}_f)}, -\mathbf{D}_\alpha, \sigma).$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

Let

$$\mathbf{g}^T = \mathbf{r}_1^T(\mathbf{A}_\beta|\mathbf{D}_\beta) + (\tilde{\mathbf{e}}_0^T|\tilde{\mathbf{e}}_1^T) \in \mathbb{Z}_q^{1\times 2m},$$

and

$$\mathbf{Q} = \begin{pmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \boldsymbol{P2}(\mathbf{R}_{\alpha,f}) \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{pmatrix} \in \mathbb{Z}_q^{(2k+1)m\times 2m},$$

where $\mathbf{E}_1 \xleftarrow{\$} \chi^{2km\times n}$, $\mathbf{E}_2, \mathbf{E}_3 \xleftarrow{\$} \chi^{2km\times m}$, vectors $\mathbf{r}_1 \xleftarrow{\$} \mathbb{Z}_q^n$, $\tilde{\mathbf{e}}_0 \xleftarrow{\$} \chi^m$, $\tilde{\mathbf{e}}_1 \xleftarrow{\$} \chi^m$.

Finally, this algorithm outputs $rk_{i\to j} = \{\mathbf{g}^T, \mathbf{Q}\}$.

– ***ReEnc***$(pp, rk_{\alpha,f\to\beta}, c_\alpha) \to c_\beta$: The inputs of this algorithm are $pp = (\mathbf{B}_1, \cdots, \mathbf{B}_l, \chi)$, $rk_{i\to j} = \{\mathbf{g}^T, \mathbf{Q}\}$ and $c_\alpha = (ct_\alpha, cc_\alpha)$. If $f(\mathbf{x}) \neq 0$ (represents that the attributes embedded in the ***Enc*** algorithm do not satisfy the policy embedded in the ***ReKeyGen*** algorithm) or $cc_\alpha = \emptyset$, output $\perp$. Otherwise, let $ct_\alpha = (\mathbf{c}_{in}, \mathbf{c}_{out})$, $cc_\alpha = (\{\mathbf{c}_i\}_{i\in[l]}) \in \mathbb{Z}_q^{lm}$. This algorithm performs the following steps in sequence:

$$\mathbf{c}_f \leftarrow \boldsymbol{Eval}_{ct}(f, \{(x_i, \mathbf{B}_i, \mathbf{c}_i)\}_{i\in[l]}),$$

$$ct_\beta^T = a\mathbf{g}^T + \boldsymbol{BD}(\mathbf{c}_f'^T)|\mathbf{c}_{out}^T) \cdot \mathbf{Q}$$

where $\mathbf{c}_f' = [\mathbf{c}_{in}|\mathbf{c}_f]$. Finally, this algorithm outputs $c_\beta = (ct_\beta, cc_\beta = \emptyset)$.

5.3. Security Proof

It is not difficult to see that such two minor modifications do not affect the selective CPA security in [14]. Since the original scheme is selective CPA secure, the revised one still satisfies selective CPA secure. Below we will mainly focus on its re-encryption simulatability and HRA security proof.

Theorem 3. *The Scheme II is re-encryption simulatable.*

Proof of Theorem 3. We have

$$\begin{aligned} ct_\beta^T &= a\mathbf{g}^T + \boldsymbol{BD}(\mathbf{c}_f'^T)|\mathbf{c}_{out}^T) \cdot \mathbf{Q} \\ &= a(\mathbf{r}_1^T(\mathbf{A}_\beta|\mathbf{D}_\beta) + (\tilde{\mathbf{e}}_0^T|\tilde{\mathbf{e}}_1^T)) + (\boldsymbol{BD}(\mathbf{c}_f'^T)|\mathbf{c}_{out}^T) \begin{pmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \boldsymbol{P2}(\mathbf{R}_{\alpha,f}) \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{pmatrix}. \end{aligned}$$

Let $ct_\beta^T = (\mathbf{c}_{in}^T, \mathbf{c}_{out}^T)$. Through a series of calculations, we get

$$\begin{cases} \mathbf{c}_{in}^T = (a\mathbf{r}_1^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{E}_1)\mathbf{A}_\beta + a\tilde{\mathbf{e}}_0^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{E}_2, \\ \mathbf{c}_{out}^T = (a\mathbf{r}_1^T + \mathbf{BD}(\mathbf{c}_f'^T)\mathbf{E}_1)\mathbf{D}_\beta + a\tilde{\mathbf{e}}_1^T + \mathbf{BD}(\mathbf{c}_f'^T)\mathbf{E}_3 + \mathbf{e}_{out}^T - [\mathbf{e}_{in}|\mathbf{e}_f]^T\mathbf{R}_{\alpha,f} + \mu\lfloor\frac{q}{2}\rfloor. \end{cases} \tag{3}$$

Let

$$\bar{\mathbf{s}}^T = a\mathbf{r}_1^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{E}_1 \in \mathbb{Z}_q^{1\times n},$$

$$\bar{\mathbf{e}}_{in}^T = a\tilde{\mathbf{e}}_0^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{E}_2 \in \chi^{1\times m},$$

$$\bar{\mathbf{e}}_{out}^T = a\tilde{\mathbf{e}}_1^T + \mathbf{BD}(\mathbf{c}_f'^T)\mathbf{E}_3 + \mathbf{e}_{out}^T - [\mathbf{e}_{in}|\mathbf{e}_f]^T\mathbf{R}_{\alpha,f} \in \chi^{1\times m}.$$

Equation (3) can be simplified to Equation (4) as follows:

$$\begin{cases} \mathbf{c}_{in}^T = \bar{\mathbf{s}}^T\mathbf{A}_\beta + \bar{\mathbf{e}}_{in}^T, \\ \mathbf{c}_{out}^T = \bar{\mathbf{s}}^T\mathbf{D}_\beta + \bar{\mathbf{e}}_{out}^T + \mu\lfloor\frac{q}{2}\rfloor. \end{cases} \tag{4}$$

According to the Definition 7, our ***Sim.ReEnc*** algorithm is as follows.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

– $\boldsymbol{Sim.ReEnc}(pp, pk_\beta, sk_\beta, c_{\alpha,\mathbf{x}}, \beta, f, \mathbf{x})$: On input $pp = (\mathbf{B}_1, \cdots, \mathbf{B}_l, \chi)$, $pk_\beta = (\mathbf{A}_\beta, \mathbf{D}_\beta)$, $sk_\beta = (\mathbf{T}_{\mathbf{A}_\beta}, \mathbf{R}_\beta)$, $c_{\alpha,\mathbf{x}} = (ct_{\alpha,\mathbf{x}}, cc_{\alpha,\mathbf{x}})$ where

$$ct_{\alpha,\mathbf{x}} = (\mathbf{A}_\alpha^T\mathbf{s} + \mathbf{e}_{in}, \mathbf{D}_\alpha^T\mathbf{s} + \mathbf{e}_{out} + \mu\lfloor\frac{q}{2}\rfloor)$$

and

$$cc_{\alpha,\mathbf{x}} = (\left\{\mathbf{c}_i = (x_i\mathbf{G} + \mathbf{B}_i)^T\mathbf{s} + \mathbf{S}_i^T\mathbf{e}_{in}\right\}_{i\in[l]}) \in \mathbb{Z}_q^{lm}.$$

Sample $\mathbf{s}'^T$ from $\mathbb{Z}_q^{1\times n}$ uniformly at random. Choose $\mathbf{e}_{in}'^T \xleftarrow{\$} \chi^{1\times m}$ and $\mathbf{e}_{out}'^T \xleftarrow{\$} \chi^{1\times m}$ uniformly at random. Then, this algorithm outputs ciphertext $ct_\beta = (\mathbf{c}_{in}^{sim}, \mathbf{c}_{out}^{sim})$ which can be decrypted by sk_β, where

$$\mathbf{c}_{in}^{sim} = \mathbf{s}'^T\mathbf{A}_\beta + \mathbf{e}_{in}'^T,$$

$$\mathbf{c}_{out}^{sim} = \mathbf{s}'^T\mathbf{D}_\beta + \mathbf{e}_{out}'^T + \mu\lfloor\frac{q}{2}\rfloor.$$

Therefore, this scheme has the property of re-encryption simulatability. □

Then, the concrete selective HRA security proof of AB-CPRE scheme is shown below.

Theorem 4. *The Scheme II is selective HRA-secure under the hardness of LWE.*

Proof of Theorem 4. First, we define three simulation algorithms, which are $\boldsymbol{Sim.ReKeyGen}_1$, $\boldsymbol{Sim.ReKeyGen}_2$ and $\boldsymbol{Sim.ReEnc}$. The algorithm $\boldsymbol{Sim.ReEnc}$ is presented in Theorem 3, and below we present two simulation algorithms, $\boldsymbol{Sim.ReKeyGen}_1$ and $\boldsymbol{Sim.ReKeyGen}_2$, respectively.

– $\boldsymbol{Sim.ReKeyGen}_1(pp, \alpha, \beta, f) \to rk_{\alpha,f\to\beta}$: If $\alpha \in \Gamma_H$, $\beta \in \Gamma_H$, let

$$\mathbf{Q} = \begin{pmatrix} \mathbf{X}_1 & \mathbf{X}_2 \\ \mathbf{0}_{m\times(l+1)m} & \mathbf{I}_{m\times m} \end{pmatrix},$$

$$\mathbf{g}^T = \mathbf{r}_1^T(\mathbf{A}_\beta|\mathbf{D}_\beta) + (\tilde{\mathbf{e}}_0^T|\tilde{\mathbf{e}}_1^T),$$

where $\mathbf{X}_1 \xleftarrow{\$} \chi^{2mk\times m}$, $\mathbf{X}_2 \xleftarrow{\$} \chi^{2mk\times m}$, $\mathbf{r}_1 \xleftarrow{\$} \mathbb{Z}_q^n$, $\tilde{\mathbf{e}}_0 \xleftarrow{\$} \chi^m$, $\tilde{\mathbf{e}}_1 \xleftarrow{\$} \chi^m$ are randomly chosen matrices or vectors. Outputs $rk_{i\to j} = \{\mathbf{g}^T, \mathbf{Q}\}$.

– $\boldsymbol{Sim.ReKeyGen}_2(pp, \alpha, \beta, f) \to rk_{\alpha,f\to\beta}$: If the adversary inputs α, β, f, where $\alpha = \theta^*$, $\beta \in \Gamma_C$ and $f(\mathbf{x}^*) \neq 0$, the algorithm does the following:

Firstly, sample $\mathbf{S}_i \leftarrow \{-1, 1\}^{m\times m}$ and run $\mathbf{S}_f^* \leftarrow \boldsymbol{Eval}_{sim}(f, \{x_i^*, \mathbf{S}_i^*\}_{i\in[l]}, \mathbf{A}_{\theta^*})$, satisfying $\mathbf{A}_{\theta^*}\mathbf{S}_f^* - f(x_i^*)\mathbf{G} = \mathbf{B}_f$.

Secondly, obtain a trapdoor $\mathbf{T}_{(\mathbf{A}_{\theta^*}|\mathbf{B}_f)}$ by running $\boldsymbol{ExtendLeft}(\mathbf{A}_{\theta^*}, f(x^*)\mathbf{G}, \mathbf{T}_\mathbf{G}, \mathbf{S}_f^*)$ algorithm.

Thirdly, sample $\mathbf{R}_{\theta^*,f} \leftarrow \boldsymbol{SamplePre}([\mathbf{A}_{\theta^*}|\mathbf{B}_f], T_{(\mathbf{A}_{\theta^*}|\mathbf{B}_f)}, -\mathbf{D}_{\theta^*}, \sigma)$. Let

$$\mathbf{g}^T = \mathbf{r}_1^T(\mathbf{A}_\beta|\mathbf{D}_\beta) + (\tilde{\mathbf{e}}_0^T|\tilde{\mathbf{e}}_1^T),$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{E}_1\mathbf{A}_\beta + \mathbf{E}_2 & \mathbf{E}_1\mathbf{D}_\beta + \mathbf{E}_3 + \boldsymbol{P2}(\mathbf{R}_{\theta^*,f}) \\ \mathbf{0}_{m\times m} & \mathbf{I}_{m\times m} \end{pmatrix},$$

where $\mathbf{E}_1 \xleftarrow{\$} \chi^{2km\times n}$, $\mathbf{E}_2, \mathbf{E}_3 \xleftarrow{\$} \chi^{2km\times m}$, $\mathbf{r}_1 \xleftarrow{\$} \mathbb{Z}_q^n$, $\tilde{\mathbf{e}}_0 \xleftarrow{\$} \chi^m$, $\tilde{\mathbf{e}}_1 \xleftarrow{\$} \chi^m$. Outputs $rk_{i\to j} = \{\mathbf{g}^T, \mathbf{Q}\}$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

It should be noted that $\mathcal{A}$ announces both the challenge user θ^* and the challenge attribute vector $\mathbf{x}^*$ in the Init phase. The security proof for selective HRA can be presented as a sequence of games, as shown below.

Game 0. This game is identical to Definition 8.

Game 1. Based on **Game 0**, this game is mainly modified $\mathcal{O}_{RekeyGen}$ and $\mathcal{O}_{ReEnc}$.

- **Re-encryption Key Generation** $\mathcal{O}_{ReKeyGen}$: If $\mathcal{A}$ inputs α, β, f and the key pairs for α and β were generated in $\mathcal{O}_{Honest}$ or $\mathcal{O}_{Corrupted}$. The oracle does the following:

  When $\alpha \in \Gamma_H$, $\beta \in \Gamma_H$, $\mathcal{C}$ returns $rk_{\alpha,f\to\beta}$ by running $\boldsymbol{Sim.ReKeyGen}_1$ algorithm, and inserts $rk_{\alpha,f\to\beta}$ into Γ_{rk} with the key-value $(\alpha, \beta, f, rk_{\alpha,f\to\beta})$;

  When $\beta \in \Gamma_H$, $\beta \in \Gamma_C$,

  1) $\alpha = \theta^*$, $f(\mathbf{x}^*) = 0$, $\mathcal{C}$ outputs $\perp$;

  2) $\alpha = \theta^*$, $f(\mathbf{x}^*) \neq 0$, $\mathcal{C}$ returns $rk_{\alpha,f\to\beta}$ by running $\boldsymbol{Sim.ReKeyGen}_2$ algorithm. Then, $\mathcal{C}$ inserts $rk_{\alpha,f\to\beta}$ into Γ_{rk} with the key-value $(\alpha, \beta, f, rk_{\alpha,f\to\beta})$;

  3) $\alpha \neq \theta^*$, $\mathcal{C}$ returns $rk_{\alpha,f\to\beta}$ by running $\boldsymbol{ReKeyGen}$ algorithm. Then, $\mathcal{C}$ inserts $rk_{\alpha,f\to\beta}$ into Γ_{rk} with the key-value $(\alpha, \beta, f, rk_{\alpha,f\to\beta})$.

  when $\alpha \in \Gamma_C$ and $\beta \in \Gamma_H$, $\alpha \in \Gamma_C$ and $\beta \in \Gamma_C$, $\mathcal{C}$ returns $rk_{\alpha,f\to\beta}$ by running $\boldsymbol{ReKeyGen}$, and inserts $rk_{\alpha,f\to\beta}$ into Γ_{rk} with the key-value $(\alpha, \beta, f, rk_{\alpha,f\to\beta})$.

  Finally, the challenger $\mathcal{C}$ outputs $rk_{\alpha,f\to\beta} = \{\mathbf{g}^T, \mathbf{Q}\}$ to the adversary $\mathcal{A}$.
- **Re-encryption** $\mathcal{O}_{ReEnc}$: Given $c_{\alpha,\mathbf{x}}$, α, β, f and k, where $k \leq$ **numCt**. If there is no value in **Ct** with key (α, k) or when $f(\mathbf{x}) \neq 0$ holds, return $\perp$. Otherwise, $\mathcal{C}$ gets $rk_{\alpha,f\to\beta}$ by searching Γ_{rk} set or queries $\mathcal{O}_{RekeyGen}$. Then, when $\beta \in \Gamma_H$, $\beta \in \Gamma_C$,

  1) $\alpha = \theta^*$, $f(\mathbf{x}^*) = 0$, and $k \in$ **Derive**, outputs $\perp$. If $f(\mathbf{x}^*) = 0$, $\beta \in \Gamma_C$ and $k \notin$ **Derive**, return the ct_β by running $\boldsymbol{Sim.ReEnc}$ algorithm.$\mathcal{C}$ outputs $\perp$;

  2) $\alpha = \theta^*$, $f(\mathbf{x}^*) \neq 0$, $\mathcal{C}$ returns ct_β by running $\boldsymbol{ReEnc}$ algorithm.

  3) $\alpha \neq \theta^*$, $\mathcal{C}$ returns ct_β by running $\boldsymbol{ReEnc}$ algorithm.

  when $\alpha \in \Gamma_H$ and $\beta \in \Gamma_H$, $\alpha \in \Gamma_C$ and $\beta \in \Gamma_H$, $\alpha \in \Gamma_C$ and $\beta \in \Gamma_C$, $\mathcal{C}$ returns ct_β by running $\boldsymbol{ReEnc}$ where $rk_{\alpha,f\to\beta} = \{\mathbf{g}^T, \mathbf{Q}\}$ were obtained by $\mathcal{C}$. Then $\mathcal{C}$ inserts $rk_{\alpha,f\to\beta}$ into Γ_{rk} with the key-value $(\alpha, \beta, f, rk_{\alpha,f\to\beta})$.

Game 2. The game being described here is the same as **Game 1**, with the only difference being the method used to generate $\mathbf{B}_1, \cdots, \mathbf{B}_l$. Let $\mathbf{B}_i = \mathbf{A}_{\theta^*}\mathbf{S}_i^* - x_i^*\mathbf{G}$ where the random matrices $\mathbf{S}_1^*, \cdots, \mathbf{S}_l^* \in \{+1, -1\}^{m\times m}$ be chosen randomly at **Setup phase**. The matrices $\{\mathbf{S}_i^*\}_{i\in[l]}$ should be kept secret, while pp can be disclosed and consists of $pp = (\mathbf{B}_1, \cdots, \mathbf{B}_l, \chi)$.

By using Definition 6, we can demonstrate that **Game 2** is statistically identical to **Game 1**. Consequently, from the perspective of the adversary, all the matrices $\mathbf{A}_{\theta^*}\mathbf{S}_i^*$ are statistically close to a uniform distribution, which implies that the $\mathbf{B}_i$ (defined as $\mathbf{B}_i = \mathbf{A}_{\theta^*}\mathbf{S}_i^* - x_1^*\mathbf{G}$ are also close to a uniform distribution. Consequently, it can be concluded that **Game 2** and **Game 1** are statistically indistinguishable.

Game 3. Compared to **Game 2**, we change how $\mathbf{A}_{\theta^*}$ is produced where a uniformly random matrix $\mathbf{A}_{\theta^*} \in \mathbb{Z}_q^{n\times m}$. The construction of $\mathbf{B}_1, \cdots, \mathbf{B}_l$ remains as the same as in **Game 2**, where $\mathbf{B}_i = \mathbf{A}_{\theta^*}\mathbf{S}_i^* - x_i^*\mathbf{G}$.

By using Definition 3, we can demonstrate that **Game 3** is statistically identical to Game **Game 2**.

Game 4. The game being described here is the same as **Game 3**, with the only difference being the method used to generate $\mathbf{c}^* = (\mathbf{c}_{in}, \mathbf{c}_{out})$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

Reduction from DLWE: Assume that $\mathcal{A}$ confers a non-negligible advantage in differentiating between **Game 4** and **Game 3**. We build $\mathcal{B}$, a DLWE solver, using $\mathcal{A}$.

– **DLWE instance**: $\mathcal{B}$ begins by obtaining a DLWE challenge consisting of two random matrices $\mathbf{A}_{\theta^*}, \mathbf{D}_{\theta^*} \in \mathbb{Z}_q^{n\times m}$, and two vectors $\mathbf{c}_{in}, \mathbf{c}_{out} \in \mathbb{Z}_q^m$. Here, $\mathbf{c}_{in}$, $\mathbf{c}_{out}$ are either random or

$$\mathbf{c}_{in} = \mathbf{A}_{\theta^*}^T \mathbf{s} + \mathbf{e}_{in},$$

$$\mathbf{c}_{out} = \mathbf{D}_{\theta^*}^T \mathbf{s} + \mathbf{e}_{out},$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_{in}, \mathbf{e}_{out} \xleftarrow{\$} \chi^m$.

Init: $\mathcal{A}$ announces challenge user θ^* and the challenge attributes vector $\mathbf{x}^*$.
Setup: The same as the **Game 3**.
Query Phase 1: The same as the **Game 3** and $(\mathbf{A}_{\theta^*}, \mathbf{D}_{\theta^*})$ be the public key for user θ^*.
Challenge Phase: After $\mathcal{A}$ sends two messages $\mu_0, \mu_1 \in \{0,1\}^m$ to $\mathcal{B}$, $\mathcal{B}$ first randomly selects a bit b from $\{0,1\}$, then computes $\mathbf{c}_{in}^* = \mathbf{c}_{in}$ and $\mathbf{c}_{out}^* = c_{out} + \mu_b \lfloor \frac{q}{2} \rfloor$. $\mathcal{B}$ sends $ct^* = (\mathbf{c}_{in}^*, \mathbf{c}_{out}^*)$ to $\mathcal{A}$. Additionally, $\mathcal{B}$ increments and add **numCt** to the set **Derive**. Finally, $\mathcal{B}$ stores ct^* to the set **Ct** with key $(x^*, \mathbf{numCt})$.
Query Phase 2: The same as the Query Phase 1 of the **Game 3**.
Decision Phase: $\mathcal{A}$ guesses if it interacts with a **Game 4** or **Game 3** challenger. Then $\mathcal{B}$ will output $\mathcal{A}$'s guess as an answer to the DLWE challenge.

As mentioned earlier, if $\mathcal{A}$ exhibits a non-negligible advantage in distinguishing **Game 4** from **Game 3**, then $\mathcal{B}$ similarly possesses a non-negligible advantage in solving the DLWE problem. We establish the security of our AB-CPRE scheme with re-encryption simulatability against selective HRA in the standard model under the LWE assumption. □

6. Construction of AB-PRE with Re-Encryption Simulatability

In this section, we use the construction idea in scheme I to obtain the first lattice-based AB-PRE scheme with the property of re-encryption simulatability by modifying the ***ReKeyGen*** and ***ReEnc*** algorithms of the AB-PRE scheme proposed in [16]. Let's primarily focus on how the modified scheme achieves re-encryption simulatability and how the selective HRA security proof is established. It's worth noting that the definition of AB-PRE, along with its selective CPA and selective HRA security models, can be found in [16].

6.1. Succinct Construction (Scheme III)

In this section, we only provide the two algorithms that have been modified, while the other four algorithms remain the same as those in [16].

– ***ReKeyGen***$(pp, sk_f, f, g) \to rk_{f\to g}$: Given $pp = \{\mathbf{A}_0, \mathbf{A}_1, \cdots, \mathbf{A}_l, \mathbf{U}, \mathbf{G}\}$ and $sk_f = \mathbf{R}_f \in \mathbb{Z}_q^{2m\times m}$. Select an attribute set $\mathbf{y} = (y_1, \cdots, y_l)$ such that $g(\mathbf{y}) = 0$. $\mathbf{R}_1 \xleftarrow{\$} \mathbb{Z}_q^{2mk\times n}$, $\mathbf{R}_2 \xleftarrow{\$} \chi^{2mk\times(l+1)m}$ and $\mathbf{R}_3 \xleftarrow{\$} \chi^{2mk\times m}$. Vectors $\mathbf{r}_1 \xleftarrow{\$} \mathbb{Z}_q^n$, $\tilde{\mathbf{e}}_0 \xleftarrow{\$} \chi^{(l+1)m}$, $\tilde{\mathbf{e}}_1 \xleftarrow{\$} \chi^m$. Then, matrices $\mathbf{H}_\mathbf{y}$, $\mathbf{Z}$, and $\mathbf{g}^T$ are defined as

$$\mathbf{H}_\mathbf{y} = [\mathbf{A}_0 | y_1\mathbf{G} + \mathbf{A}_1 | \cdots | y_l\mathbf{G} + \mathbf{A}_l] \in \mathbb{Z}_q^{n\times(l+1)m},$$

$$\mathbf{Z} = \begin{pmatrix} \mathbf{R}_1\mathbf{H}_\mathbf{y} + \mathbf{R}_2 & \mathbf{R}_1\mathbf{U} + \mathbf{R}_3 - P\mathbf{2}(\mathbf{R}_f) \\ \mathbf{0}_{m\times(l+1)m} & \mathbf{I}_{m\times m} \end{pmatrix} \in \mathbb{Z}_q^{(2k+1)m\times(l+2)m},$$

$$\mathbf{g}^T = \mathbf{r}_1^T(\mathbf{H}_\mathbf{y}|\mathbf{U}) + (\tilde{\mathbf{e}}_0^T|\tilde{\mathbf{e}}_1^T) \in \mathbb{Z}_q^{1\times(l+2)m},$$

respectively. Output $rk_{i\to j} = \{\mathbf{g}^T, \mathbf{Z}\}$ along with the attribute vector $\mathbf{y}$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 November 2023 doi:10.20944/preprints202311.1769.v1

– ***ReEnc***$(pp, rk_{f\to g}, c_{\mathbf{x}}, \mathbf{x}) \to c_{\mathbf{y}}$: Given $pp = \{\mathbf{A}_0, \mathbf{A}_1, \cdots, \mathbf{A}_l, \mathbf{U}, \mathbf{G}\}$, $rk_{i\to j} = \{\mathbf{g}^T, \mathbf{Z}\}$, $c_{\mathbf{x}} = (\mathbf{c}_{in}, \mathbf{c}_1, \cdots, \mathbf{c}_l, \mathbf{c}_{out})$, and $\mathbf{x} = (x_1, \cdots, x_l)$. If $f(\mathbf{x}) \neq 0$, output $\perp$. Otherwise, this algorithm computes

$$\mathbf{c}_f \leftarrow \boldsymbol{Eval}_{ct}(\{x_i, \mathbf{A}_i, \mathbf{c}_i\}_{i=1}^{l}, f).$$

Let $\mathbf{c}'_f = [\mathbf{c}_{in}|\mathbf{c}_f] \in \mathbb{Z}_q^{2m}$. Sample a small random number $a \in \chi$. Compute

$$c_{\mathbf{y}}^T = a\mathbf{g}^T + (\boldsymbol{BD}(\mathbf{c}_f'^T)|\mathbf{c}_{out}^T) \cdot \mathbf{Z}.$$

Output $c_{\mathbf{y}}$ along with the attribute vector $\mathbf{y}$.

6.2. Security Proof

It is not difficult to see that such two minor modifications do not affect the selective CPA security in [16]. Since the original scheme is selective HRA secure, the revised one still satisfies selective HRA secure. Below we will mainly focus on its re-encryption simulatability.The proof of its HRA security using re-encryption simulatability is similar to the proof of Scheme III, which we will not expand in detail in this paper.

Theorem 5. *The Scheme III is re-encryption simulatable.*

Proof of Theorem 5. First, let's review the ***ReEnc*** algorithm in the modified scheme.

$$c_{\mathbf{y}}^T = a\mathbf{g}^T + (\boldsymbol{BD}(\mathbf{c}_f'^T)|\mathbf{c}_{out}^T) \cdot \mathbf{Z}.$$

Let $c_{\mathbf{y}}^T = (c_{\mathbf{y}0}^T, c_{\mathbf{y}1}^T)$. Through a series of calculations, we get

$$\begin{cases} c_{\mathbf{y}0}^T = a\mathbf{r}_1^T\mathbf{H}_{\mathbf{y}} + a\tilde{\mathbf{e}}_0^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{R}_1\mathbf{H}_{\mathbf{y}} + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{R}_2, \\ c_{\mathbf{y}1}^T = a\mathbf{r}_1^T\mathbf{U} + a\tilde{\mathbf{e}}_1^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{R}_1\mathbf{U} + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{R}_3 - \mathbf{c}_f'^T\mathbf{R}_f + \mathbf{c}_{out}^T. \end{cases} \tag{5}$$

Let

$$\bar{\mathbf{s}}^T = a\mathbf{r}_1^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{R}_1 \in \mathbb{Z}_q^{1\times n},$$

$$\bar{\mathbf{e}}^T = a\tilde{\mathbf{e}}_0^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{R}_2 \in \chi^{1\times(l+1)m},$$

$$\bar{\mathbf{e}}_{out}^T = a\tilde{\mathbf{e}}_1^T + \boldsymbol{BD}(\mathbf{c}_f'^T)\mathbf{R}_3 + \mathbf{e}_{out}^T - [\mathbf{e}_{in}|\mathbf{e}_f]^T\mathbf{R}_f \in \chi^{1\times m}.$$

Equation (5) can be simplified to Equation (6) as follows:

$$\begin{cases} c_{\mathbf{y}0}^T = \bar{\mathbf{s}}^T\mathbf{H}_{\mathbf{y}} + \bar{\mathbf{e}}^T, \\ c_{\mathbf{y}1}^T = \bar{\mathbf{s}}^T\mathbf{U} + \bar{\mathbf{e}}_{out}^T + \mu\lfloor\frac{q}{2}\rfloor. \end{cases} \tag{6}$$

According to Definition 7, we can propose the ***Sim.ReEnc*** algorithm as follows.

– ***Sim.ReEnc***$(pp, c_{\mathbf{x}}, f, g, \mathbf{x}, \mathbf{R}_g, \mu)$: Uniformly sample $\mathbf{s}'$ from $\mathbb{Z}_q^n$ uniformly at random. Choose $\mathbf{e}'$ and $\mathbf{e}'_{out}$ uniformly at random from $\chi^{(l+1)m}$ and χ^m, respectively. Then, this algorithm outputs

$$c_{\mathbf{y}0} = \mathbf{H}_{\mathbf{y}}^T\mathbf{s}' + \mathbf{e}',$$

$$c_{\mathbf{y}1} = \mathbf{U}^T\mathbf{s}' + \mathbf{e}'_{out} + \mu\lfloor\frac{q}{2}\rfloor.$$

Output $c_{\mathbf{y}} = (c_{\mathbf{y}0}, c_{\mathbf{y}1})$ along with the attribute vector $\mathbf{y}$ and note that $\mathbf{R}_g$ can correctly decrypt this ciphertext.

From above, we know that the scheme III has re-encryption simulatability property. □

The security proof of the selective HRA given in [16] is unusually complex. That method relies heavily on the construction of the scheme itself and is not scalable. In this study, we ensure that the scheme III is HRA-secure under the hardness of LWE by primarily utilizing re-encryption simulatability property. The detailed process of the proof is similar to the proof of the scheme II, so I will not go into detail in this work.

7. Conclusions

In this work, we first proposed a generic method to elevate the security of PRE schemes based on key switching technique from CPA security to HRA security and simplify the HRA proof. A concise lattice-based PRE scheme was constructed to better illustrate our generic method, which has the re-encryption simulatability property. Utilizing this method, we made some improvements to AB-PRE and AB-CPRE of ESORICS'21 to make them have the re-encryption simulatability property. Simultaneously, we show how to improve the AB-CPRE scheme [14] by enhancing its security from selective CPA to selective HRA and the AB-PRE scheme [16] by simplifying the security proof. Moreover, our method can be extended to other PRE schemes or its variants schemes based on key switching technique, and is not limited to the above two examples.

Author Contributions: Conceptualization, all authors; methodology, B.W., L.H., L.Y.; validation, all authors; formal analysis, L.H., F.X., J.W.; writing—original draft preparation, B.W., L.H. L.Y.; supervision, J.W.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Plan of China (Grant No. 2020YFB1005600), Major Program of Guangdong Basic and Applied Research Project (Grant No. 2019B030302008), National Natural Science Foundation of China (Grant No. 61825203), Guangdong Provincial Science and Technology Project (Grant Nos. 2017B010111005, 2021A0505030033), National Joint Engineering Research Center of Network Security Detection and Protection Technology and Guangdong Key Laboratory of Data Security and Privacy Preserving. This work is also supported by Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation ("Climbing Program" Special Funds) (Grant No. pdjh2021a0050). We all thank the reviewers for their valuable comments and suggestions which improve the content and presentation of this work a lot.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PRE	Proxy Re-encryption
CPA	Chosen-Plaintext Attacks
HRA	Honest Re-encryption Attacks
FHE	Fully Homomorphic Encryption
AB-PRE	Attribute-based Proxy Re-encryption
AB-CPRE	Attribute-based Conditional Proxy Re-encryption
LWE	Learning With Errors
DLWE	Decisional Learning With Errors
DBDH	Decisional Bilinear Diffie-Hellman assumption

References

1. Blaze, M.; Bleumer, G.; Strauss, M. Divertible Protocols and Atomic Proxy Cryptography. In Proceedings of the EUROCRYPT 1998; Nyberg, K., Ed. Springer, 1998, Vol. 1403, *LNCS*, pp. 127–144.
2. Ge, C.; Susilo, W.; Wang, J.; Fang, L. Identity-based Conditional Proxy Re-encryption with Fine Grain Policy. *Comput. Stand. Interfaces* **2017**, *52*, 1–9.
3. Deng, H.; Qin, Z.; Wu, Q.; Guan, Z.; Zhou, Y. Flexible Attribute-based Proxy Re-encryption for Efficient Data Sharing. *Inf. Sci.* **2020**, *511*, 94–113.

4. Shao, J.; Cao, Z. Multi-use Unidirectional Identity-based Proxy Re-encryption from Hierarchical Identity-based Encryption. *Inf. Sci.* **2012**, *206*, 83–95.
5. Qin, Z.; Xiong, H.; Wu, S.; Batamuliza, J. A Survey of Proxy Re-encryption for Secure Data Sharing in Cloud Computing. *IEEE Transactions on Services Computing* **2016**.
6. Su, M.; Zhou, B.; Fu, A.; Yu, Y.; Zhang, G. PRTA: A Proxy Re-encryption based Trusted Authorization scheme for Nodes on CloudIoT. *Inf. Sci.* **2020**, *527*, 533–547.
7. Zhuang, E.S.; Fan, C.I. Multi-Keyword Searchable Identity-Based Proxy Re-Encryption from Lattices. *Mathematics* **2023**, *11*, 3830.
8. Agyekum, K.O.O.; Xia, Q.; Sifah, E.B.; Cobblah, C.N.A.; Xia, H.; Gao, J. A Proxy Re-Encryption Approach to Secure Data Sharing in the Internet of Things Based on Blockchain. *IEEE Syst. J.* **2022**, *16*, 1685–1696.
9. Manzoor, A.; Braeken, A.; Kanhere, S.S.; Ylianttila, M.; Liyanage, M. Proxy Re-encryption Enabled Secure and Anonymous IoT Data Sharing Platform based on Blockchain. *J. Netw. Comput. Appl.* **2021**, *176*, 102917.
10. Xiao, Y.; Xu, L.; Chen, Z.; Zhang, C.; Zhu, L. A Blockchain-Based Data Sharing System with Enhanced Auditability. *Mathematics* **2022**, *10*, 4494.
11. Cohen, A. What About Bob? The Inadequacy of CPA Security for Proxy Reencryption. In Proceedings of the PKC 2019; Lin, D.; Sako, K., Eds. Springer, 2019, Vol. 11443, *LNCS*, pp. 287–316.
12. Ateniese, G.; Fu, K.; Green, M.; Hohenberger, S. Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. *ACM Trans. Inf. Syst. Secur.* **2006**, *9*, 1–30.
13. Gentry, C. A fully homomorphic encryption scheme. PhD thesis, Stanford University, USA, 2009.
14. Liang, X.; Weng, J.; Yang, A.; Yao, L.; Jiang, Z.; Wu, Z. Attribute-Based Conditional Proxy Re-encryption in the Standard Model Under LWE. In Proceedings of the ESORICS 2021; Bertino, E.; Shulman, H.; Waidner, M., Eds. Springer, 2021, Vol. 12973, *LNCS*, pp. 147–168.
15. Luo, F.; Al-Kuwari, S.M.; Wang, F.; Chen, K. Attribute-based Proxy Re-encryption from Standard Lattices. *Theor. Comput. Sci.* **2021**, *865*, 52–62.
16. Susilo, W.; Dutta, P.; Duong, D.H.; Roy, P.S. Lattice-Based HRA-secure Attribute-Based Proxy Re-Encryption in Standard Model. In Proceedings of the ESORICS 2021; Bertino, E.; Shulman, H.; Waidner, M., Eds. Springer, 2021, Vol. 12973, *LNCS*, pp. 169–191.
17. Agrawal, S.; Boneh, D.; Boyen, X. Efficient Lattice (H)IBE in the Standard Model. In Proceedings of the EUROCRYPT 2010; Gilbert, H., Ed. Springer, 2010, Vol. 6110, *LNCS*, pp. 553–572.
18. Micciancio, D.; Peikert, C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In Proceedings of the EUROCRYPT 2012; Pointcheval, D.; Johansson, T., Eds. Springer, 2012, Vol. 7237, *LNCS*, pp. 700–718.
19. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for Hard Lattices and New Cryptographic Constructions. In Proceedings of the Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008; Dwork, C., Ed. ACM, 2008, pp. 197–206.
20. Brakerski, Z.; Gentry, C.; Vaikuntanathan, V. (Leveled) fully homomorphic encryption without bootstrapping. In Proceedings of the Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012; Goldwasser, S., Ed. ACM, 2012, pp. 309–325.
21. Agrawal, S.; Boneh, D.; Boyen, X. Efficient Lattice (H)IBE in the Standard Model. In Proceedings of the EUROCRYPT 2010; Gilbert, H., Ed. Springer, 2010, Vol. 6110, *LNCS*, pp. 553–572.
22. Ateniese, G.; Benson, K.; Hohenberger, S. Key-Private Proxy Re-encryption. In Proceedings of the CT-RSA 2009; Fischlin, M., Ed. Springer, 2009, Vol. 5473, *LNCS*, pp. 279–294.