PREs with HRA Security and Key Privacy
Based on Standard LWE Assumptions

Yang Wang\textsuperscript{1,2}\textsuperscript{[0000−0001−9274−8195]}, Yanmin Zhao\textsuperscript{3}, and Mingqiang Wang\textsuperscript{1,2}

\textsuperscript{1} School of Mathematics, Shandong University, Jinan, Shandong, 250100, P.R. China
\textsuperscript{2} Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, Shandong, 250100, P.R. China
\textsuperscript{3} Department of Computer Science, The University of Hong Kong, P.R. China
wyang1114@sdu.edu.cn
ymzhao@cs.hku.hk
wangmingqiang@sdu.edu.cn

\textbf{Abstract.} Proxy re-encryption (PRE) schemes, which nicely solve the problem of delegating decryption rights, enable a semi-trusted proxy to transform a ciphertext encrypted under one key into a ciphertext of the same message under another arbitrary key. For a long time, the semantic security of PREs is quite similar to that of public key encryption (PKE) schemes. Cohen first pointed out the insufficiency of the security under chosen-plaintext attacks (CPA) of PREs in PKC 2019, and proposed a \textit{strictly stronger} security notion, named security under honest re-encryption attacks (HRA), of PREs. Surprisingly, a few PREs satisfy the stronger HRA security and almost all of them are paring-based till now. To the best of our knowledge, we present the first detailed construction of HRA secure single-hop PREs based on standard LWE problems with \textit{comparably small and polynomially-bounded} parameters in this paper. Combing known reductions, the HRA security of our PREs could also be guaranteed by the worst-case basic lattice problems (e.g. SIVP\textsubscript{γ}). Meanwhile, our single-hop PRE schemes are also key-private, which means that the implicit identities of a re-encryption key will not be revealed even in the case of a proxy colluding with some corrupted users. Some discussions about key-privacy of multi-hop PREs are also proposed, which indicates that several constructions of multi-hop PREs do not satisfy their key-privacy definitions.

\textbf{Keywords:} Lattice-based Cryptography · Proxy Re-Encryption · Key Privacy · HRA Security · LWE

\section{Introduction}

A (public key, unidirectional) PRE scheme enables a semi-trusted proxy to transform ciphertexts under Alice’s public key into ciphertexts which are decryptable by Bob’s secret key with the help of a re-encryption key generated by Alice. Ever since it was initiated in [20] and formally constructed by Blaze,
BLEUMER AND STRAUSS IN [7], PRE HAS BECOME A POWERFUL CRYPTOGRAPHIC PRIMITIVE WITH VARIOUS APPLICATIONS, INCLUDING EMAIL FORWARDING AND PUBLISH/SUBSCRIBE SYSTEMS [5, 7, 8, 26], SECURING CLOUD STORAGE AND DISTRIBUTED FILE SYSTEMS WITH FINE-GRAINED ACCESS CONTROL [6, 18, 23, 29], INTERNET OF THINGS [1, 2], AND SO ON.

COMPARING WITH TRADITIONAL PKEs, THERE ARE TWO ADDITIONAL PROBABILISTIC POLYNOMIAL-TIME (PPT) ALGORITHMS OF PRE SCHEMES. ONE IS A RE-ENCRYPTION KEY GENERATION ALGORITHM, WHICH COULD BE USED TO GENERATE A RE-ENCRYPTION KEY FROM A USER ALICE TO ANY OTHER USER BOB WITH THE HELP OF ALICE’S SECRET KEY. THE OTHER IS A RE-ENCRYPTION ALGORITHM USED TO COVERT CIPHERTEXTS OF ALICE TO CIPHERTEXTS OF BOB. HOWEVER, THE SEMANTIC SECURITY (NAMELY, THE CPA SECURITY [3, 5, 6, 26] AND THE CCA SECURITY (SECURITY UNDER CHOSEN-CIPHERTEXT ATTACKS) [8, 12, 17]) OF PREs ARE VERY SIMILAR TO THOSE OF PKEs FOR A LONG TIME. THEORETICALLY, A THOUGHT-OUT GAME-BASED SECURITY DEFINITION OF PREs SHOULD TAKE ALL POSSIBLE SITUATIONS INTO ACCOUNT AFTER REMOVING ALL THE CASES IN WHICH AN ADVERSARY COULD WIN THE CORRESPONDING SECURITY GAME TRIVIALLY. THE DEFINITION OF CPA (AND CCA) SECURITY OF PREs IS OUT OF SUCH AN IDEAL RANGE, SINCE BOTH THE RE-ENCRYPTION ORACLE AND THE RE-ENCRYPTION KEY GENERATION ORACLE FROM AN HONEST USER (INCLUDING THE CHALLENGE USER) TO ANY CORRUPTED USER ARE FORBIDDEN IN THE CPA (AND CCA) SECURITY GAME. CONSTRAINTS ON QUERYING THE RE-ENCRYPTION ORACLE FROM THE CHALLENGE USER TO CORRUPTED USERS IS NOT SO REASONABLE. IN APPLICATIONS, AN ADVERSARY COULD, OF COURSE, GET SOME RE-ENCRYPTED CIPHERTEXTS FROM AN HONEST USER (SAY ALICE) TO SOME CORRUPTED USERS WITHOUT DRAWING ALICE’S ATTENTION. SO, WE MAY HOPE THAT THE SEMANTIC SECURITY OF OTHER CIPHERTEXTS OF ALICE STILL HOLDS UNDER THIS SITUATION. HOWEVER, THE CCA SECURITY OF PREs IS INADEQUATE EVEN FOR AN HONEST-BUT-CURIOS DELEGATEE (SAY BOB) [10]. AS OBSERVED IN [10], THE CPA SECURE PRE SCHEME PROPOSED IN [26] DOES NOT PREVENT BOB FROM LEARNING THE SECRET KEY OF A DELEGATOR (SAY ALICE) WITH NON-NEGATIVABLE PROBABILITY AFTER RECEIVING A SINGLE HONESTLY RE-ENCRYPTED CIPHERTEXT OF ALICE. BOB COULD DO THIS EVEN WITHOUT KNOWING THE CORRESPONDING RE-ENCRYPTION KEY. ESSENTIALLY, THE DEFINITION OF THE CPA SECURITY OF PREs CAPTURES LESS SECURITY GUARANTEES AGAINST THE DELEGATEE (BOB).


---

4 Such a PRE scheme is called non-interactive [6], i.e., no trusted third party or interaction is required in the re-encryption key generation process.
mainly focus on PREs based on lattice in this paper, and aiming to design PREs with HRA security.

1.1 Our Contributions and Technique Overview

In this paper, we present the first HRA secure single-hop PRE scheme with comparably small and polynomially-bounded parameters in the standard model.

Our single-hop PRE schemes also satisfy the key-privacy property [5], a notion stating that it is impossible for the proxy and a set of colluding users to derive either the sender or the receiver’s identities from a re-encryption key even when given the public keys and flexible interaction abilities within PREs. Both the key privacy and the HRA security of our schemes are based on standard LWE problems. Therefore, combing known reductions, we could show that our PRE schemes satisfy the HRA security and the key privacy for modulus $q = \tilde{O}(n^3)$, as long as the corresponding worst-case SIVP problem with $\gamma = \tilde{O}(n^{3.5})$ is hard.

Meanwhile, our single-hop PRE scheme could also be modified to a PRE+ scheme in which the re-encryption key of a ciphertext could be generated by its creator. Moreover, our single-hop PREs with polynomially-bounded parameters are also adaptive secure for some directed acyclic graph, since our single-hop constructions satisfy the requirements proposed in [13]. Some discussions about key-privacy of multi-hop PREs are also proposed, which indicates that several constructions of multi-hop PREs [3, 25] do not satisfy their key-privacy definitions.

Technique Overview: Roughly speaking, there are two types of techniques for generating re-encryption keys in lattice-based PREs. We take the dual cryptosystem [14] as an example. The form of ciphertexts of a user $i$ for a message $\mu \in \{0,1\}^m$ is $(c_{i,1} = B^T_i \cdot s + e_1, c_{i,2} = D^T_i \cdot s + e_2 + \lfloor \frac{s}{q} \rfloor \cdot \mu)$, where $B_i, D_i \in \mathbb{Z}^{n \times m}_u$ are the public key of $i$, $s \in \mathbb{Z}^m_\mu, e_1, e_2 \in \mathbb{Z}^m_\mu$ are some short vectors. The secret key of $i$ is either a short matrix $R_i$ satisfying $B_i \cdot R_i = D_i$ mod $q$ (the case 1), or a trapdoor $T_{B_i}$ of $B_i$ for pre-image sampling (the case 2). For the case 1, the re-encryption key from $i$ to a user $j$ could be designed as $rk_{i\rightarrow j} := (S^T_{i,j} B_j + E_{i,j,1} S^T_{i,j} D_j + E_{i,j,2} + R_i) \in \mathbb{Z}^{2m \times 2m}_q$, where $S_{i,j}, E_{i,j,1}, E_{i,j,2}$ are matrices with small elements. Then, the re-encryption of $(c_{i,1}, c_{i,2})$ is $(c_{j,1,1}, c_{j,2}) = (S^T_{i,j} B_j, S^T_{i,j} D_j) \cdot rk_{i\rightarrow j}$. Notice that $rk_{i\rightarrow j} \cdot (-R_i^T, I_{m \times m})^T \approx (-R_j^T, I_{m \times m})^T$, the user $j$ could decrypt $(c_{j,1,1}, c_{j,2})$ successfully. For the case 2, the re-encryption key could be designed as $rk_{i\rightarrow j} := (R_{i,j,1}; R_{i,j,2})$, where $R_{i,j,1}$ and $R_{i,j,2}$ are matrices with short elements which are sampled by using the trapdoor $T_{B_i}$ of $B_i$, and satisfy $B_i \cdot R_{i,j,1} = B_j, B_i \cdot R_{i,j,2} = D_j - D_i$. Then, the re-encryption of $(c_{i,1}, c_{i,2})$ is $(c_{j,1}, c_{j,2}) = (R_{i,j,1}^T \cdot c_{i,1}, R_{i,j,2}^T \cdot c_{i,1} + c_{i,2})$.

The biggest difficulty to achieve HRA security for the above two types of constructions is how to answer re-encryption queries from the challenge user $i^*$.
to other corrupted users. Since in the security proof, the public key of $i^*$ is usually changed to be uniform and independent (in order to embed corresponding LWE instances). Therefore, the challenger could not answer re-encryption queries in both cases. To handle this dilemma, we combine together the above two types of constructions. Let’s take the single-hop construction as an example. Ciphertexts of a user $i$ in our constructions are of the form $c_{i,1} = B_i^T \cdot s + e_{i,1}$, $c_{i,2} = D_i^T \cdot s + e_{i,2} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mu$ and $c_{i,3} = A_i^T \cdot s + e_{i,3}$. Here, $A_{i,1} = A_i + [0] \cdot H \cdot G \cdot [H^T \cdot G]$ = $[\hat{A} \cdot A \cdot R_{i,1} + H \cdot G] \cdot A \cdot R_{i,2} + H^T \cdot G$ with $H'$ a random tag, the public key of $i$ is $(B_i, D_i, A_i)$ with $B_i \cdot R_i = D_i \mod q$, and the secret key of $i$ is $(R_{i,1}, R_{i,2})$ whose elements are all short. Notice that, $(c_{i,1}, c_{i,2})$ and $R_i$ are sufficient for decryption. To create a re-encryption key from $i$ to another user $j$, we could use $[\hat{A} \cdot A \cdot R_{i,1} + H \cdot G]$ to execute pre-image sampling for invertible $H$ [21], and generate $rk_{i \rightarrow j} = (R_{i,j,1}, R_{i,j,2})$ satisfying $[\hat{A} \cdot A \cdot R_{i,1} + H \cdot G] \cdot R_{i,j,1} = B_j - B_i$ and $[\hat{A} \cdot A \cdot R_{i,2} + H \cdot G] \cdot R_{i,j,2} = D_j - D_i$. Then, a re-encryption of $(c_{i,1,2}, c_{i,3})$ is $(c_{j,1,2}) = (R_{i,j,1} + c_{i,1} + c_{i,3} + c_{i,2}, c_{i,3})$, where $R_{i,j,1}^{(1)}$ and $R_{i,j,2}^{(2)}$ could be constructed from $R_{i,j,1}$ and $R_{i,j,2}$ easily. Notice that, though the forms of different level ciphertexts are different, they could be decrypted by using the same decryption algorithm. Meanwhile, for lattice-based PREs, it seems that different forms of ciphertexts are necessary. For more details, please refer to discussions above Definition 4 and Remark 5.

In the security proof, we will change $A_{i^*}$ of the challenge use $i^*$ to be the form $[\hat{A} \cdot A \cdot R_{i,1} | A \cdot R_{i,2} - H^* \cdot G]$. Here, $H^*$ is a random tag of the challenge ciphertext. In order to answer re-encryption queries of ciphertexts with tags $H \neq H^*$ from $i^*$ to corrupted users, the challenger could use trapdoor $R_{i^*,2}$ and $c_{i^*,3}$ (more precisely, the parts of $c_{i^*,3}$ corresponding to $[\hat{A} \cdot A \cdot R_{i^*,2} + (H - H^*) \cdot G]$) to solve corresponding LWE-like problems and recover $s$ (hence, $e_{i^*,1}, e_{i^*,2}, e_{i^*,3}$ and $\mu$) by using a LWE inversion algorithm [21]. Then, if we design the corresponding re-encryption algorithm carefully (please refer to the construction for more details), the challenger could generate a ciphertext that is distributed statistically close to a real re-encrypted ciphertext with the help of $e_{i^*,1}, e_{i^*,2}, e_{i^*,3}, s$ and $\mu$.

To achieve the key-privacy, a key observation is that we could replace $B_j$ and $D_j$ with random elements and re-randomize them by LWE problems. Since in our constructions, the re-encryption key generation algorithm does not use $A_j$. After replacing $B_j$ and $D_j$, the user $j$ still has the ability to generate re-encryption keys from himself to other users. That is to say, we could replace $B_j, D_j$ to be $S_{i,j} \cdot B_j + E_{i,j,1}$ and $S_{i,j} \cdot D_j + E_{i,j,2}$. Then, under corresponding LWE assumptions and by the properties of Gaussian distributions, the distribution of $R_{i,j,1}$ and $R_{i,j,2}$ is (computationally) closed to some discrete Gaussian distribution that contains no information about identities $i$ and $j$. Our constructions are not tag-based CCA secure [12], since in our constructions, the challenger has no ability to answer the decryption query of re-encrypted ciphertexts of the challenge user $i^*$.

\[ ^{6} \text{Notice that, we could only decrypt a fresh ciphertext by using this trapdoor in our design. A re-encrypted ciphertext does not contain information about matrix } A \text{ anymore.} \]
Organization: In Section 2, we will recall some notations and fundamental facts we need. Definitions of PRE schemes, detailed constructions and security analyses are put in Section 3.

2 Preliminaries

In this section, we introduce some notations and background results. Symbol \([n]\) denotes the set \(\{1, 2, \cdots, n\}\). When we write \(X \leftarrow \xi\), we mean that the random variable \(X\) follows the distribution \(\xi\). If \(S\) is a finite set, then \(|S|\) is its cardinality and \(U(S)\) is the uniform distribution over \(S\). Symbols \(\mathbb{Z}^+\) and \(\mathbb{R}^+\) stand for the sets of positive integers and positive reals. We use \(\log x\) to represent \(\log_2 x\) for \(x \in \mathbb{R}^+\). We also denote \([\cdot, \cdot]\) as the horizontal concatenation of vectors or matrices.

2.1 Lattices and Gaussian Distributions

A (full-rank) \(n\)-dimensional integer lattice \(\Lambda \subseteq \mathbb{Z}^n\) is a discrete additive group whose \(\mathbb{R}\)-span is \(\mathbb{R}^n\). The basis (which is not unique) of a lattice \(\Lambda\) is a linearly independent set of lattice vectors \(B := \{b_1, \cdots, b_n\}\), whose \(\mathbb{Z}\)-span is \(\Lambda\). In this case, we will denote it by \(\Lambda = \mathcal{L}(B)\). For a matrix \(A \in \mathbb{Z}^{n \times m}\) and any vector \(u \in \mathbb{Z}^n\) admitting an integral solution \(x \in \mathbb{Z}^m\) s.t. \(A \cdot x = u \mod q\), we define the “\(q\)-ary” integer lattices (or cosets) as \(\Lambda^q(A) = \{z \in \mathbb{Z}^m : A \cdot z = 0 \mod q\}\), \(\Lambda^q_s(A) = \{z \in \mathbb{Z}^m : A \cdot z = u \mod q\}\) and \(\Lambda(A^T) = \{z \in \mathbb{Z}^m : \exists s \in \mathbb{Z}^n \text{ s.t. } z = A \cdot s \mod q\}\).

The Gaussian distributions are defined as follows. For any \(s > 0, c \in H\), which is taken to be \(s = 1\) or \(c = 0\) when omitted, we define the (spherical) Gaussian function \(\rho_{s, c} : \mathbb{R}^n \to (0, 1]\) as \(\rho_{s, c}(x) = e^{-\pi \|x-c\|^2 / \sigma^2}\). By normalizing this function, we obtain the continuous Gaussian probability distribution \(D_{s, c}\) of parameter \(s\), whose density function is given by \(s^{-n} \cdot \rho_{s, c}(x)\). Applying a linear transformation given by a nonsingular matrix \(B\) to a spherical Gaussian with \(s = 1\) yields the Gaussian function

\[
\rho_B(x) := \rho(B^{-1} \cdot x) = e^{-\pi x^T \Sigma^{-1} x},
\]

where \(\Sigma = B \cdot B^T\) is a positive definite matrix (written \(\Sigma > 0\)) \(^7\). Since \(\rho_B\) is distributed only up to \(\Sigma\), we’ll denote it by \(\rho_{\sqrt{\Sigma}}\) in the following. Normalizing \(\rho_{\sqrt{\Sigma}}\) by its total measure \(\int_{\mathbb{R}^n} \rho_{\sqrt{\Sigma}}(x) dx = \sqrt{\det \Sigma}\) over \(\mathbb{R}^n\), we could obtain the probability distribution function of the continuous Gaussian distribution \(D_{\sqrt{\Sigma}}\). Functions are extended to sets in the usual way, e.g. \(\rho_{s, c}(A) = \sum_{x \in A} \rho_{s, c}(x)\). Recall that, for an \(n\)-dimensional lattice \(\Lambda\) and a positive real \(\varepsilon\), the smooth parameter \(\eta_{\varepsilon}(\Lambda)\) of \(\Lambda\) is defined as the smallest \(s\) such that \(\rho_{1/2}(A_\varepsilon \setminus \{0\}) \leq \varepsilon\) \(^8\).

For our applications, we need the following lemmata \([14, 16, 21, 22, 24, 27]\).

\(^7\) Recall that for every positive definite matrix \(\Sigma\), there exists a unique positive definite matrix \(\sqrt{\Sigma}\) such that \((\sqrt{\Sigma})^2 = \Sigma\).

\(^8\) Here, \(A_\varepsilon = \{x \in \mathbb{R}^n : \langle x, A \rangle < \varepsilon\}\).
Lemma 1. Let \( n, q \) be positive integers with \( q \) prime, and let \( m \geq 2n \log q \).

- For all but \( q^{-n} \) fraction of \( A \in \mathbb{Z}_q^{n \times m} \) and any \( \omega(\sqrt{\log m}) \) function, there is a negligible function \( \varepsilon(m) \), such that \( \eta_c(A^\perp) \leq \omega(\sqrt{\log m}) \).
- For any \( s \geq \omega(\sqrt{\log m}) \), the distribution of \( u = A \cdot e \mod q \) with \( e \leftarrow D_{\mathbb{Z}_q^m, s} \) is within statistical distance \( 2\varepsilon \) of \( U(\mathbb{Z}_q^n) \) for some negligible function \( \varepsilon = \varepsilon(m) \). Furthermore, fix \( u \in \mathbb{Z}_q^n \) and let \( t \in \mathbb{Z}_m \) be an arbitrary solution of \( A \cdot t = u \mod q \), then the conditional distribution of \( e \leftarrow D_{\mathbb{Z}_q^m, s} \) given \( A \cdot e = u \mod q \) is exactly \( D_{A^\perp(A), s} = t \) and \( D_{A^\perp(A), s} \).
- \( \eta_c(\mathbb{Z}^m) \leq \omega(\sqrt{\log m}) \) for some negligible \( \varepsilon = \varepsilon(m) \). For any \( s \geq \omega(\sqrt{\log m}) \), there is an efficient sampling algorithm to output samples with distribution that is statistically close to \( D_{\mathbb{Z}_q^m, s} \).

Lemma 2. The following facts hold:

1. Let \( A \) be an \( n \)-dimensional lattice and \( u \in \mathbb{R}^n \) be any vector. Assume \( r, s > 0 \) are two reals, and \( t = \sqrt{r^2 + s^2} \). If \( \frac{\eta_c}{t} \geq \eta_c(A) \) for some \( \varepsilon \leq \frac{1}{2} \), then the distribution \( Y \) obtained by sampling from \( D_{A + u, r} \) and then adding a noise vector taken from \( D_s \), is within statistical distance \( 4\varepsilon \) from \( D_t \).
2. Let \( \Sigma_1, \Sigma_2 > 0 \) be positive definite matrices with \( \Sigma = \Sigma_1 + \Sigma_2 > 0 \). Let \( \Lambda_1, \Lambda_2 \) be lattices such that \( \sqrt{\Sigma_1} \geq \eta_c(\Lambda_1) \) for some positive \( \varepsilon \leq \frac{1}{2} \). For arbitrary vectors \( c_1, c_2 \in \mathbb{R}^n \), consider the following probabilistic experiments:

   Samples \( x_2 \leftarrow D_{\Lambda_2 + c_2, \sqrt{\Sigma_2}} \), then set \( x_1 = x_2 + D_{\Lambda_1 + c_1, -x_2, \sqrt{\Sigma_1}} \).

   We have that the marginal distribution of \( x_1 \) is within statistical distance \( 8\varepsilon \) of \( D_{\Lambda_1 + c_1, \sqrt{\Sigma_1}} \). Moreover, if \( x_2 \) is instead chosen from \( D_{\sqrt{\Sigma_2}} \), the marginal distribution of \( x_1 \) is also as above.

Lemma 3. Let \( q, m, m' \) be positive integers and \( r \) be a positive real satisfying \( r > \max\{\omega(\sqrt{\log m}), \omega(\sqrt{\log m'})\} \). Let \( b \in \mathbb{Z}_q^m \) be arbitrary and \( x \) chosen from \( D_{\mathbb{Z}_q^m, r} \). Then for any \( V \in \mathbb{Z}^{m \times m'} \) and positive real \( s > s_1(V) \), there exists a PPT algorithm \( \text{ReRand}(V, b + x, r, s) \) that outputs \( c \) such that \( c^T = b^T \cdot V + x' \), where \( x' \) is distributed statistically close to \( D_{\mathbb{Z}^{m', 2r s}} \).

Lemma 4. Let \( F \leftarrow D_{\mathbb{Z}^{m \times m}}, \) assume for convenience that \( m \geq n \). Then, with all but \( 2^{-m} \) probability it holds that \( s_1(F) \leq \gamma \cdot C \cdot \sqrt{m} \), where \( C \approx \frac{1}{\sqrt{2\pi}} \) is a universal constant.

Lemma 5. Let \( A \subseteq \mathbb{R}^n \) be any lattice, \( c \in \text{Span}_\mathbb{R}(A) \), and \( \sigma \geq \eta_c(A) \) for some \( \varepsilon \in (0, 1) \). We have \( \Pr[||D_{A + c, \sigma}|| \geq \sigma \cdot \sqrt{n}] \leq 2^{-n} \cdot \frac{1 + \varepsilon}{1 - \varepsilon} \). Moreover, if \( c = 0 \), then the bound holds for any \( \sigma > 0 \) with \( \varepsilon = 0 \).

Lemma 6. For any \( n \)-dimensional lattice \( A \), vector \( c \in \mathbb{R}^n \), positive \( \varepsilon > 0 \), \( \sigma \geq 2\eta_c(A) \) and \( c \in A \), we have \( D_{A, \sigma, c, e}(x) \leq \frac{1 + \varepsilon}{1 - \varepsilon} \cdot 2^{-n} \). In particular, for \( \varepsilon \leq \frac{1}{3} \), the min-entropy of \( D_{A, \sigma, c} \) is at least \( n - 1 \).

\(^9\) For any matrix \( A \), symbol \( s_1(A) \) represents its largest singular value.
For a modulus \( q \geq 2 \), we set \( k = \lceil \log q \rceil \), and let \( G \in \mathbb{Z}^{m \times nk} \) be the “gadget” matrix defined in \cite{ AUD}. The lattice \( A^\perp(G) \) has a basis \( B \) with \( \|B\|_{\text{GS}} \leq \sqrt{5}^{10} \).

**Lemma 7.** Let \( n \) be an integer, \( k = \lceil \log q \rceil \) with \( q \geq 2 \) a modulus, and \( m \geq 2nk \).

For any invertible matrix \( H \in \mathbb{Z}_q^{n \times n} \), \( A \leftarrow U(\mathbb{Z}_q^{n \times m}) \) and \( R \leftarrow D_{\mathbb{Z}_q^{m \times nk}, s} \) for some \( s = \omega(\log m) \geq \eta_e(\mathbb{Z}^m) \), we have:

- The matrix \( A = [A, A \cdot R - H \cdot G] \in \mathbb{Z}_q^{m + nk} \) is within statistical distance \( \text{negl}(n) \) of \( U(\mathbb{Z}_q^{n \times (m + nk)}) \).
- For \( b^T = s^T \cdot A + e^T \), where \( s \in \mathbb{Z}_q^n \) is arbitrary and either \( ||e|| < \frac{q}{C_T \sqrt{n \cdot \log q}} \)
  with \( C_T \) a universal constant or \( e \leftarrow D_{\mathbb{Z}_q^{m \times q}) \) with \( \frac{1}{\gamma} \geq \sqrt{n \cdot \log q} \cdot \omega(\sqrt{\log n}) \)
 there is a deterministic algorithm \text{Invert}(R, A, b) \ outputs \( s \) and \( e \).
- For any \( u \in \mathbb{Z}_q^n \) and large enough \( \gamma = O(\sqrt{n \cdot \log q}) \), there is a randomized algorithm \text{SampleD}(R, A; u; \gamma) \ samples from a distribution within \( \text{negl}(n) \) statistical distance of \( D_{A, \gamma, \omega(\sqrt{\log n})} \).

We note that for any fixed matrix \( A \), algorithm \text{SampleD} in Lemma 7 could output samples from a distribution within \( \text{negl}(n) \) statistical distance of \( D_{A, \gamma, \omega(\sqrt{\log n})} \) for any \( \gamma \geq \gamma \). Also, Lemmata 1 and 2 (together with the fact that \( \frac{1}{\sqrt{r}} \cdot \rho_s(A) \leq \rho_s(A + u) \leq \rho_s(A) \) for any \( \varepsilon \in (0, 1) \), \( s \geq \varepsilon(A) \) and any vector \( u \), which is implied by Lemmata 2.9 and 4.4 of \cite{BDFKST}) imply the following result: for \( m \geq 2n \log q \), \( A \leftarrow U(\mathbb{Z}_q^{n \times m}) \) and \( \gamma \geq \omega(\sqrt{\log m}) \), the distribution of \( e \) obtained by first choosing a vector \( u \leftarrow U(\mathbb{Z}_q^n) \), then choose \( e \leftarrow D_{A, \gamma, \gamma} \) is statistically closed to \( D_{\mathbb{Z}_q^{m \times \gamma}} \).

In our constructions, we need a large set \( \mathcal{G} \subset \mathbb{Z}_q^{n \times n} \) with the “unit differences” property: for any matrices \( X \neq Y \in \mathcal{G} \), the matrix \( X - Y \) is invertible in \( \mathbb{Z}_q^{n \times n} \). Such a set could be constructed efficiently, and for \( q = p^e \) with \( p \) a prime, we have \( |\mathcal{G}| = p^n \) \cite{AUD}.

### 2.2 The LWE Problems

Even since introduced by Regev \cite{R}, the LWE problems have become a fundamental building block of many cryptographic primitives over lattices. The formal definition of (normal form) decision LWE problems is proposed as follows.

**Definition 1.** For positive integers \( n, m, q \) and a noise distribution \( \chi \) over \( \mathbb{Z} \), the decision LWE problems with \( m \) samples, denoted by \( \text{LWE}_{n, q, m, \chi} \) is to distinguish the following two distributions:

\[
(A, A^T \cdot s + e) \text{ and } (A, u)
\]

where \( A \leftarrow U(\mathbb{Z}_q^{n \times m}) \), \( s \leftarrow \chi^n \), \( e \leftarrow \chi^m \), and \( u \leftarrow U(\mathbb{Z}_q^m) \) are sampled independently.

\(^{10}\) For a matrix \( R \in \mathbb{Z}_q^{n \times n} \), denote \( \|R\|_{\text{GS}} \) as the longest column of the Gram-Schmidt orthogonalization of \( R \).
There are quantum reductions from worst-case lattice problems to decision LWE problems for suitable $\chi$ [4, 27]. By using standard hybrid argument, it is possible to show that decision LWE problems with multiple secrets are also hard. Namely, it is hard to distinguish the following two distributions:

$$(A, A^T \cdot S + E) \text{ and } (A, U)$$

where $A \leftarrow U(Z_q^{n \times m})$, $S \leftarrow \chi^{n \times k}$, $E \leftarrow \chi^{m \times k}$ and $U \leftarrow U(Z_q^{m \times k})$ are sampled independently. We will denote the corresponding problem by Multi-LWE$_{n,q,m,\chi}$.

3 The PRE Schemes Based on LWE

In this section, we shall give formal definitions and security models of PRE schemes, present some simple discussions, and propose our PRE schemes based on standard LWE problems with different properties.

3.1 Definitions and Security Models

Let’s first recall the definitions of (single-hop) PRE schemes [5, 10].

**Definition 2.** Let $\lambda$ be the security parameter. A single-hop proxy re-encryption scheme contains the following six PPT algorithms.

- **Setup$(1^\lambda)$**: The key management center runs this algorithm with input $\lambda$, and generates the public parameters $pp$.
- **KeyGen$(pp)$**: To generate a public/secret key pair of a user $i$, the key management center runs this algorithm with input $pp$, and generates $(pk_i, sk_i)$.
- **Enc$(pp, pk_i, m)$**: The encryption algorithm, takes as input the public parameters $pp$, a message $m$ and the public key $pk_i$ of a user $i$, generates a ciphertext $(C_i; 1)$\textsuperscript{11} associated with $m$ and $i$.
- **ReKeyGen$(pp, sk_i, pk_j)$**: This algorithm, which is executed by a user $i$, takes as input the public parameters $pp$, the secret key $sk_i$ of the user $i$ and the public key $pk_j$ of a user $j$, outputs a re-encryption key $rk_{i\rightarrow j}$ which could be used to re-encrypt any level 1 ciphertext of the user $i$ to a corresponding level 2 ciphertext of the user $j$.
- **ReEnc$(pp, (C_i; 1), rk_{i\rightarrow j})$**: This algorithm is executed by the proxy, and takes as input the public parameters $pp$, a level 1 ciphertext $C_i$ of a user $i$, a re-encryption key $rk_{i\rightarrow j}$, and outputs a level 2 ciphertext $C_j$ of $j$.
- **Dec$(pp, (C_i; \kappa), sk_i)$**: For any $\kappa \in [2]$, this algorithm could decrypt any level-$\kappa$ ciphertext $C_i$ of a user $i$ to a message $m$ with the secret $sk_i$ and the public parameters $pp$.

\textsuperscript{11} Throughout this paper, we call such a ciphertext a fresh (level 1) ciphertext. In our constructions, every ciphertext $C_i$ contains its corresponding level information.
Notice that in our definition, one could decrypt his ciphertexts with his secret key by using the same decryption algorithm, regardless of the level of ciphertexts. So, our definitions and constructions satisfy the so-called proxy invisibility [23].

As usual, the correctness is required that for any \( pp \leftarrow \text{Setup}(1^\lambda) \), any key pair \((pk_{i_1, j}, sk_{i_1}) \leftarrow \text{KeyGen}(pp) \) with \( j \in [2] \), \((C_{i_1}; 1) = \text{Enc}(pp, pk_{i_1}, m) \) with any plaintext \( m \), any re-encryption key \( rk_{i_1, i_2} = \text{ReKeyGen}(pp, sk_{i_1}, pk_{i_2}) \), and \((C_{i_2}; 2) = \text{ReEnc}(pp, C_{i_1}, rk_{i_1, i_2}) \), we have

\[
\Pr[\text{Dec}(pp, C_{i_j}, sk_{i_j}) \neq m] \leq \text{negl}(\lambda)
\]

for all \( j \in [2] \).

The HRA security, proposed in [10], is strictly stronger than the CPA security [3, 5]. In the definition of the HRA security, re-encryptions even from the challenge identity \( i^* \) to corrupted users are also allowed, as long as the queried ciphertext does not equal to the challenge ciphertext (or some derivations of it). In our constructions, every (fresh) ciphertext is labeled by some randomly chosen tag as [12]. The formal definition of the HRA security of (our tag-based) single-hop PRE schemes could be modified as follows.

**Definition 3.** Consider the following interactions between a PPT adversary \( \mathcal{A} \) and a challenger \( \mathcal{C} \):

- **Phase 1:**
  1. The challenger \( \mathcal{C} \) generates the public parameters \( pp \) and sends them to an adversary \( \mathcal{A} \). It also maintains a counter \( \text{numCt} \) which is initialized to be 0, two databases \( D_K, D_C \) which are initialized to be empty, and a value \( \text{Value} \) which is also initialized to be 0.
  2. The adversary \( \mathcal{A} \) first decides the sizes of two sets \( U_H \) and \( U_C \), and could adaptively query the following two oracles.
     - **Corrupted key generation oracle,** in which \( \mathcal{C} \) would return \((pk, sk) = \text{KeyGen}(pp) \) to \( \mathcal{A} \) and record \((pk, sk)\), together with corresponding user index, to the set \( U_C \).
     - **Uncorrupted key generation oracle,** in which \( \mathcal{C} \) would first generate \((pk, sk) = \text{KeyGen}(pp) \), then return \( pk \) to \( \mathcal{A} \) and record \((pk, sk)\), together with corresponding user index, to the set \( U_H \).

- **Phase 2:** In this phase, \( \mathcal{A} \) could adaptively query the following oracles. We require that the queried users’ indexes must have been appeared in Phase 1, and user indexes \( i \neq j \) if exist.
  1. **Encryption oracle:** On input an index \( i \) and a message \( m \), the challenger \( \mathcal{C} \) computes \( C_i = \text{Enc}(pp, pk_i, m) \), increments \( \text{numCt} \), then adds the item \((\bot, i; \bot, \text{ct}_i := (C_i; H_{Ci}; 1); \text{numCt}) \) \( \text{numCt} \) to \( D_C \) and returns it to the adversary \( \mathcal{A} \).

---

\( H_{Ci} \) is a random tag which could be regarded as a part of \( C_i \). We emphasize its role by giving it explicitly in this definition. In our constructions, the tag of different ciphertexts could be the same (which do not affect securities).
2. Re-encryption key generation oracle: Upon receiving indexes \((i, j)\), \(C\) generates a re-encryption key \(rk_{i\rightarrow j} = \text{ReKeyGen}(pp, sk_i, pk_j)\), records the tuple \((i, j, rk_{i\rightarrow j})\) to the database \(\mathcal{D}_K\), and returns \(rk_{i\rightarrow j}\) to \(A\). If this oracle is queried after the challenge oracle, we require \(i \neq i^*\) when \(j \in U_C\).

3. Re-encryption oracle: \(A\) submits query \((i, j, k)\) for some \(k \leq \text{numCt}\), the challenger \(C\) will first retrieve \(\mathcal{D}_C\) to recover an item \((\bot, i; \bot, ct_i; k)\). If no such item exists, output \(\bot\). If \(k = \text{Value}\) and \(j \in U_C\), output \(\bot\). Otherwise, generate a re-encryption key \(rk_{i\rightarrow j} = \text{ReKeyGen}(pp, sk_i, pk_j)\) and parse \(ct_i = (C_i; H_{C_i}; 1)\). Then, the challenger will compute \(C_j = \text{ReEnc}(pp, C_i, rk_{i\rightarrow j})\), increment \(\text{numCt}\), record \((i, j; ct_i, ct_j := (C_j; \bot; 2); \text{numCt})\) to \(\mathcal{D}_C\), and return it to \(A\).

4. Challenge oracle: This oracle could only be queried once. After \(A\) submits a challenge \((i^*, m^*_0, m^*_1)\) with \(i^* \in U_H\) and \(m^*_0 \neq m^*_1\), \(C\) will first judge whether the adversary has queried a re-encryption key from \(i^*\) to some corrupted users by using \(\mathcal{D}_K\). If such re-encryption keys exist, return \(\bot\). Otherwise, he chooses a random bit \(b \leftarrow U(\{0, 1\})\), and computes \(C_b = \text{Enc}(pp, pk_{i^*}, m^*_b)\). Then, \(C\) increments \(\text{numCt}\), records the corresponding item \((\bot, i^*; \bot, ct^* := (C_b; H^{ct^*}; 1); \text{numCt})\) to \(\mathcal{D}_C\) and returns it to \(A\).

   - Phase 3: \(A\) outputs a bit \(b^*\), and wins if and only if \(b^* = b\).

We say a single-hop PRE scheme is HRA secure, if for any PPT adversary \(A\), its advantage \(\Pr[A \text{ wins}] := |\Pr[b^* = b] - \frac{1}{2}|\) in the above game is negligible.

Remark 1. In the above definition (as well as that proposed in \([10]\)), only indistinguishability of fresh ciphertexts is considered. It’s possible to modify the encryption algorithm of Definition 2 to be level-based as \([12]\). The corresponding encryption algorithm is proposed as following:

\[-\text{Enc}(pp, pk_i, m, \kappa) : \text{The encryption algorithm, takes as input the public parameters } pp, \text{ a message } m, \text{ the public key } pk_i \text{ of a user } i \text{ and a level } \kappa \in [2], \text{ generates a ciphertext } (C_i; \kappa) \text{ associated with } m \text{ and the user } i.\]

We use Definition 2 mainly because we only consider the HRA security proposed in \([10]\). One could also define the indistinguishability of level 2 ciphertexts, either by generating corresponding ciphertexts via encryption algorithms, or by generating corresponding ciphertexts via re-encryption algorithms. For some constructions (e.g. those are source hiding \([13]\), or re-encryption simulatable \([10]\) \[^{13}\]), these definitions are equivalent. Most importantly, for challenge users all belonged to \(U_H\), the (fresh ciphertexts, \(\kappa = 1\)) HRA security defined in Definition 2 implies the HRA security of level-2 ciphertexts (defined via re-encryption algorithms) \[^{15}\].

\[^{13}\] Here, \(\bot\) could also be changed to a randomly selected tag \(H_{C_i}\).

\[^{14}\] Source hiding requires corresponding ciphertexts to be \textit{computationally} indistinguishable, while re-encryption simulatable requires the distributions of corresponding ciphertexts to be \textit{statistically} close to each other.

\[^{15}\] For those constructions with high level (\(\kappa \geq 2\)) ciphertexts having different forms or distributions (when generating from encryption algorithms or re-encryption algorithms), things seem to be more complicated.
Remark 2. Note that there is no constraint on the claimed public key. This requirement is not enough in general. Our definition of the HRA security is different from that proposed in [10]. More precisely, we allow the adversary to query re-encryption keys from non-challenge honest users to corrupted users as [12]. As explained in [10], we need to use the encryption oracle to track derivations of the challenge ciphertext. Another important reason which has not been mentioned in [10] is that for a single-hop PRE scheme, the correctness only guarantees that we could decrypt successfully for both fresh and re-encrypted ciphertexts. However, for a ciphertext which is re-encrypted by 2 times, we do not know the probability that we could decrypt it successfully. If some of the re-encryption processes are malformed, this probability may be very high, especially in some lattice-based PRE schemes [3, 12, 17].

Most importantly, the encryptions of the challenge user $i^*$ must be recorded in the definition the HRA security, since adding a small error to the challenge ciphertext may do not affect its decryption in latticed-based constructions. Similar concerns also exist in multi-hop lattice-based PRE schemes. Definitions given in [12] did not use the encryption oracle, and only required the queried ciphertext to be well-formed in the sense that it was an encryption of a message under the claimed public key. This requirement is not enough in general.

The key privacy of a single-hop PRE scheme is defined as follows [3, 5].

**Definition 4.** Consider the following interactions between a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, in which Phase 1 is almost the same as that in Definition 3 with the differences that the challenger does not need to maintain $\mathcal{D}_C$ and $\text{numCt}$. 

- **Phase 2:** In this phase, $\mathcal{A}$ could adaptively query the following oracle. We require that the queried users’ indexes must have been appeared in Phase 1.
  1. Re-encryption key generation oracle: Upon receiving indexes $(i, j)$ with $i \neq j$, the challenger $\mathcal{C}$ will generate $r_{k_{i \rightarrow j}} = \text{ReKeyGen}(pp, sk_i, pk_j)$, record $(i, j, r_{k_{i \rightarrow j}})$ to $\mathcal{D}_K$ and return $r_{k_{i \rightarrow j}}$ to $\mathcal{A}$. If this oracle is queried after the Challenge oracle and $(i, j) = (i^*, j^*)$, we require the generated re-encryption key does not equal to $r_{k^*}$.
  2. Re-encryption oracle: $\mathcal{A}$ submits query $(i, j, ct_j)$ with $i \neq j$, the challenger $\mathcal{C}$ will first create $r_{k_{i \rightarrow j}} = \text{ReKeyGen}(pp, sk_i, pk_j)$, and return $ct_j = \text{ReEnc}(pp, ct_i, r_{k_{i \rightarrow j}})$ to $\mathcal{A}$.
  3. Challenge oracle: This oracle could only be accessed once. On input $(i^*, j^*)$, the challenger $\mathcal{C}$ creates $r_{k_{i^* \rightarrow j^*}} = \text{ReKeyGen}(pp, sk_{i^*}, pk_{j^*})$ until $(i^*, j^*, r_{k_{i^* \rightarrow j^*}}) \notin \mathcal{D}_K$, if no such key exists, return $\bot$. Then, he samples a random $b \leftarrow U\{0, 1\}$, and returns $r_{k^*} := r_{k_{i^* \rightarrow j^*}}$ to $\mathcal{A}$ if $b = 1$, or returns a random key $r_{k^*}$ (which may obey to some distributions with large enough entropy) in the key space to $\mathcal{A}$ if $b = 0$. The constraints are $i^* \neq j^*$ and $j^* \in \Gamma_H$.

- **Phase 3:** $\mathcal{A}$ outputs a bit $b'$, and wins if and only if $b' = b$.

We say a single-hop PRE scheme is key-private, if for any PPT adversary $\mathcal{A}$, its advantage $\Pr[\mathcal{A} \text{ wins}] = |\Pr[b' = b] - \frac{1}{2}|$ in the above game is negligible.

Remark 2. Note that there is no constraint on $i^*$ as in [3], which means that honest delegates are enough to provide key privacy. This definition seems stronger,
since key-privacy definition proposed in [5] requires \( i^* \in U_H \). However, in many applications, if user \( i^* \) is corrupted, there may be no point to discuss the privacy of re-encryption key \( rk_{i^*,i^*} \), since the re-encryption key from \( i^* \) to \( j^* \) is usually produced by the user \( i^* \).

Remark 3. In the key privacy definition proposed in [5], the re-encryption key \( rk_{i,j} \) generated for each pair of users \( i, j \) is unique. Under this constraint, the re-encryption algorithm can’t be deterministic. However, in Definition 4, even many different re-encryption keys from challenger user \( i^* \) to \( j^* \) could be exposed to the adversary, as long as the re-encryption key sampled in the challenge oracle never appears before. Our sing-hop constructions satisfy this stronger key-privacy definition, however schemes constructed in [5] do not remain key-private if multiple keys per pair are released.

The key-privacy defined in Definition 4 is very strong in the sense that we do not constrain the ability of an adversary to query re-encryption key generation oracle and re-encryption oracle from challenge user \( i^* \) and \( j^* \) to corrupted users. Next, we give a simple impossible result, which shows that the key privacy defined in Definition 4 could not be applied to multi-hop PRE schemes directly. This result also means that the multi-hop PRE schemes proposed in [3, 25] do not satisfy their key-privacy definitions. We also notice that definitions proposed in [19] do not suffer this problem, since in their security definitions, the re-encryption key of each pair of users is the same, and re-encryption oracles from a user \( i \) to a user \( j \) should be queried after the corresponding query of the re-encryption key generation oracle. Meanwhile, the challenge ciphertext could not be re-encrypted any more.

**Lemma 8.** The key privacy defined in Definition 4 could not be applied to multi-hop PRE scheme which satisfies correctness and CPA security.

**Proof.** We give an attack for the multi-hop case. Consider the case both \( i^* \) and \( j^* \) in the challenge phase belonging to \( U_H \). Adversary \( A \) could choose an arbitrary message \( m \), encrypt it under user \( i^* \)’s public key and get \( C_{i^*} \). Notice that \( A \) could do this both before and after the challenge phase. Assume \( rk_{i^*,j^*} \) is the re-encryption key obtained from \( C_{i^*} \). \( A \) then compute \( C_{j^*} = \text{ReEnc}(pp, C_{i^*}, rk_{i^*,j^*}) \), and query the re-encryption oracle with input \( j^*, k, C_{j^*} \) to get corresponding ciphertext \( C_k \). Here, \( k \) is an arbitrary user in \( U_C \). Finally, \( A \) computes \( m' = \text{Dec}(pp, C_k, sk_k) \) and judges whether \( m' = m \).

If \( rk_{i^*,j^*} \) is a valid re-encryption key from user \( i^* \) to \( j^* \), then \( m' = m \) with overwhelming probability by the correctness of multi-hop PRE schemes. While if \( rk_{i^*,j^*} \) is a random key, then the probability that \( m' = m \) is negligible since the PRE scheme is CPA secure.

Remark 4. The key-privacy definition proposed in [3] adds an additional constraint in the challenge phase that there should be no chain of re-encryption keys from \( j^* \) to any \( k \in U_C \). However, attack showed in Lemma 8 means that this constraint is not enough, i.e. the key privacy definition of multi-hop PRE re-encryption schemes proposed in [3] is not well-defined.
To define the key-privacy of multi-hop PRE schemes, a key point is to prohibit the adversary from winning the distinguishing game via attack proposed in Lemma 8 trivially. So, we also need to record the ciphertext generated in the security game as Definition 3. The details are omitted.

### 3.2 Single-Hop Key-Private PRE Schemes with HRA Security

In this subsection, we propose our single-hop PRE schemes, and use the following parameters.

- The modulus $q$ is a prime, $\mathcal{S} \subseteq \mathbb{Z}_{q}^{n \times n}$ is a set with the unit differences property [21].
- Integers $k = \lceil \log q \rceil$, $m = m + 2nk$.
- $s = \omega(\sqrt{\log m}) \geq \eta_{e}(\mathbb{Z}^{m})$ is some fixed function, which is used for sampling Gaussian distributions over $\mathbb{Z}^{m}$.
- $\beta = C \cdot s \cdot \sqrt{m} + 1$ (which is used in the ReRand algorithm of Lemma 3) with $C$ the global constant appeared in Lemma 4; $\gamma = O(\sqrt{n \cdot \log q} \cdot \omega(\log m))$ (which is used for pre-image sampling) with the $O$ function appeared in Lemma 7.
- $\chi = D_{\mathbb{Z}, \alpha}$ is the error distribution of corresponding (single-secret/multi-secrets) LWE problems with $\alpha \geq 2\sqrt{m} \geq 2 \cdot \omega(\sqrt{\log m})$ (which is used for ensuring worst-case to average-case reductions of LWE problems).

In the following constructions, $G \in \mathbb{Z}_{n \times nk}$ is the gadget matrix [21]. To ensure correctness of decryptions, we require

$$q \geq 10 \cdot \omega(\log m) \cdot \sqrt{m} \cdot (1 + \sqrt{n} \cdot \alpha + \sqrt{\frac{\alpha}{m}} \cdot \beta \cdot \gamma) \cdot \alpha$$

in general. If $\alpha \leq \beta \cdot \gamma$ and $\omega(\sqrt{\log m}) \geq 1 + \sqrt{\frac{\alpha}{m}} \cdot \sqrt{m} \cdot (\sqrt{\alpha} + \sqrt{n} \cdot \alpha \cdot \beta)$ is sufficient. Notice that $k = O(\log q) = O(1)$, if we set $m = 2nk$, then $q = O(n^{2} \cdot \alpha^{2})$ is sufficient. To ensure worst-case to average-case reductions of LWE problems, we set $\alpha = \alpha' \cdot q$ (according to our definition of LWE problems), and have a (quantum) reduction from worst-case SIVP$_{\gamma}$ problem to LWE problem with $\gamma = O(n^{2.5})$, as long as $\alpha' \cdot q \geq 2\sqrt{m}$ [27]. So, a possible parameter setting is $\alpha' = O(n^{-2.5})$ and $q = O(n^{3})$. Meanwhile, we could conclude that our PRE schemes satisfy HRA security and key privacy assuming the worst-case SIVP$_{O(n^{3})}$ problem is hard (by Theorems 1 and 2).

Our single-hop key-private PRE scheme with HRA security is proposed as follows.

- **Setup**$(1^{\lambda})$: This algorithm uses large enough integer $n$ (e.g. $n \geq 512$) and parameters satisfying the above relations. It also samples $\hat{A} \leftrightarrow U(\mathbb{Z}_{q}^{n \times m})$, $H_{1} \leftrightarrow U(\mathcal{S})$ and outputs the public parameters

$$pp = (n, q, m, \chi, D_{\mathbb{Z}, \alpha}, h, G; \hat{A}; H_{1}).$$

$^{16}$ Note that $D_{\mathbb{Z}, \alpha}^{m} = D_{\mathbb{Z}^{m}, \alpha}$.
\(-\textbf{KeyGen}(pp)\) : To generate a public/secret key pair of a user \(i\), this algorithm, implemented by the key management center, samples matrices \(B_i \leftarrow U(\mathbb{Z}_q^{n \times m})\), \(R_i \leftarrow D_{\mathbb{Z}_q^m}^{m \times m}\) and \(R_{i,1}, R_{i,2} \leftarrow D_{\mathbb{Z}_q}^{m \times nk}\). Then, it computes \(A_i = [A_i \cdot R_{i,1}, A_i \cdot R_{i,2}] \in \mathbb{Z}_q^{n \times m}\), \(D_i = B_i \cdot R_i \in \mathbb{Z}_q^{n \times m}\) and returns
\[
(pk_i, sk_i) = ((A_i, B_i, D_i), (R_{i,1}, R_{i,1}, R_{i,2})).
\]

\(-\textbf{Enc}(pp, pk_i, \mu)\) : To encrypt a message \(\mu \in \{0, 1\}^m\) under the public key \(pk_i\) of a user \(i\), this algorithm first samples \(s \leftarrow \mathbb{Z}^n\) and errors \(e_{i,1}, e_{i,2} \leftarrow \mathbb{Z}^m\). It also samples error \(e_{i,3} \leftarrow D_{\mathbb{Z}_q^m \times 2^{\beta \cdot 0}}\) and \(H_{e_i} \leftarrow U(\mathbb{S})\). Then, it sets the level matrix
\[
A_{i,1} = A_i + \lfloor 0 \rceil \underbrace{H_i \cdot G}_{H_{e_i} \cdot G} \cdot \underbrace{A_i \cdot R_{i,1} + H_i \cdot G \cdot A_i \cdot R_{i,2}}_{H_{e_i} \cdot G} \in \mathbb{Z}_q^{n \times m},
\]
and computes \(c_{i,1} = B_i^T \cdot s + e_{i,1} \mod q\), \(c_{i,2} = D_i^T \cdot s + e_{i,2} + \lfloor \frac{q}{2} \rceil \cdot \underbrace{\mu}_{\mathbb{Z}^m}\) \(\mod q\), \(c_{i,3} = A_i^T \cdot s + e_{i,3} \mod q\). Finally, it outputs \(ct_i = (c_{i,1}, c_{i,2}, c_{i,3}; H_{e_i}, 1) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^m \times \mathbb{Z}_q^m \times \mathbb{Z}^n \times \mathbb{Z}^n \times \mathbb{Z}^n\).

\(-\textbf{ReKeyGen}(pp, sk_i, pk_j)\) : The user \(i\) samples matrices \(X_{1,1} \in \mathbb{Z}_q^{m \times m}, X_{1,2}, X_{1,3} \in \mathbb{Z}_q^{m \times nk}, X_{2,1} \in \mathbb{Z}_q^{nk \times m}, X_{2,2}, X_{2,3} \in \mathbb{Z}_q^{nk \times nk}\) by using algorithm \textbf{SampleD} (of Lemma 7) with \(R_{i,1}\) and \(\gamma\), such that
\[
A_{i,1} \cdot \begin{pmatrix} X_{1,1} & X_{1,2} & X_{1,3} \\ X_{2,1} & X_{2,2} & X_{2,3} \end{pmatrix} \in \mathbb{Z}_q^{m \times m},
\]
where \(S_{i,j} \leftarrow \mathbb{Z}^n \quad \text{and} \quad E_{i,j,1} \leftarrow \mathbb{Z}^m\). Similarly, he also samples matrices \(X'_{1,1} \in \mathbb{Z}_q^{m \times m}, X'_{1,2}, X'_{1,3} \in \mathbb{Z}_q^{m \times nk}, X'_{2,1} \in \mathbb{Z}_q^{nk \times m}, X'_{2,2}, X'_{2,3} \in \mathbb{Z}_q^{nk \times nk}\) by using algorithm \textbf{SampleD} with \(R_{i,1}\) and \(\gamma\), such that
\[
A_{i,1} \cdot \begin{pmatrix} X'_{1,1} & X'_{1,2} & X'_{1,3} \\ X'_{2,1} & X'_{2,2} & X'_{2,3} \end{pmatrix} \in \mathbb{Z}_q^{m \times m},
\]
where \(E_{i,j,2} \leftarrow \mathbb{Z}^n\). In the end, he returns the corresponding re-encryption key
\[
r_{i \rightarrow j} = \{X'_{i,j'; i' \in \{1, 2\}, j' \in \{1, 2, 3\}}\}.
\]

\(-\textbf{ReEnc}(pp, ct_i, r_{i \rightarrow j})\) : The proxy parses \(ct_i = (c_{i,1}, c_{i,2}, c_{i,3}; h(\mu), 1)\). If \(ct_i\) does not follow to this form, return \(\perp\). Otherwise, it sets
\[
R_{i,j}^{(1)} = \begin{pmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,3} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,3} \end{pmatrix}^T, R_{i,j}^{(2)} = \begin{pmatrix} X'_{1,1} & X'_{1,2} & \cdots & X'_{1,3} \\ X'_{2,1} & X'_{2,2} & \cdots & X'_{2,3} \end{pmatrix}^T.
\]

\[17\] In fact, \(S_{i,j} \cdot B_j + E_{i,j,1}\) is sufficient.
He first computes
\[ c'_{j,1} := R_{i,j}^{(1)} \cdot c_{i,3} + c_{i,1} + R_{i,j}^{(1)} \cdot e'_1 \mod q \]
and
\[ c'_{j,2} := R_{i,j}^{(2)} \cdot c_{i,3} + c_{i,2} + R_{i,j}^{(2)} \cdot e'_2 \mod q. \]

Here, \( e'_k = ((e_{k'})^T, (e_{k'})^T, 0_{q_k}^T)^T \in \mathbb{R}^m \) with \( e_{k',1} \leftrightarrow D_m^k \) and \( e_{k',2} \leftrightarrow D_{qk}^k \) for \( k' \in \{1, 2\} \). Then, he computes \( c_{j,1} = c'_{j,1} + e''_1 \mod q \) and \( c_{j,2} = c'_{j,2} + e''_2 \mod q. \) Here, \( e''_1 \leftrightarrow D_{\mathbb{Z}^m - e'_{j,1}, r} \) with
\[ r_{k'} = \sqrt{(\bar{m} + nk) \cdot \gamma^2 - ||R_{i,j,k'}||^2} \cdot (4/3^2 + 1) \cdot \alpha^2 + \omega(\sqrt{\log m}) \]
for \( k' \in [m] \), where \( \{R_{i,j,k'}\} \)'s are the row vectors of \( R_{i,j}^{(1)} \). Similarly, \( e''_2 \leftrightarrow D_{\mathbb{Z}^m - e'_{j,2}, r} \) with
\[ r_{k'} = \sqrt{(\bar{m} + nk) \cdot \gamma^2 - ||R_{i,j,k'}||^2} \cdot (4/3^2 + 1) \cdot \alpha^2 + \omega(\sqrt{\log m}) \]
for \( k' \in [m] \), where \( \{R_{i,j,k'}\} \)'s are the row vectors of \( R_{i,j}^{(2)} \). Finally, he returns
\[ ct_j = (e_{j,1}, e_{j,2}, c_{i,3}) \leftrightarrow D \]

- \( \text{Dec}(pp, ct_i, sk_i) : \) To decrypt a ciphertext of user \( i \), we first parse \( ct_i = (c_{i,1}, c_{i,2}, c_{i,3}, H_{e_i} \text{ or } \perp, l) \) with \( l \in \{1, 2\} \), compute \( \mu' = -R_i^T \cdot c_{i,1} + c_{i,2} \mod q \), and return \( \mu \) with \( \mu_i = \lfloor \frac{\mu_i}{2} \rfloor \) for \( i \in [m] \).

To show the correctness, we first notice that for a fresh (i.e level 1) ciphertext \( (c_{i,1}, c_{i,2}, c_{i,3}, H_{e_i}, 1) \) of a user \( i \), we have
\[ -R_i^T \cdot c_{i,1} + c_{i,2} \mod q = -R_i^T \cdot e_{i,1} + e_{i,2} + \lfloor \frac{q}{2} \rfloor \cdot \mu \mod q. \]

Since \( e_{i,1}, e_{i,2} \leftrightarrow \chi^m \), the absolute value of each of the coefficients of \(-R_i^T \cdot e_{i,1} + e_{i,2}\) is bounded by \( N_1 := (\sqrt{m} + 1) \cdot \omega(\sqrt{\log m}) \). Then, for any \( j \in [m] \), we get \( \mu_j = \text{err}_1 + \lfloor \frac{q}{2} \rfloor \cdot \mu_j \) for \( \text{err}_1 \in (-N_1, N_1) \) with overwhelming probability. Therefore, our choice of \( q \) ensures that for any \( j \in [m] \), \( \mu_j = \lfloor \frac{q}{2} \rfloor \cdot (\text{err}_1 + \lfloor \frac{q}{2} \rfloor \cdot \mu_j) \), since \( ||\text{err}_1||_{\infty} \leq \frac{q}{2} \).

For a re-encrypted (i.e level 2) ciphertext \( (c_{j,1}, c_{j,2}, \perp, 2) \), we first analyze the distributions of \( c'_{j,1} \) and \( c'_{j,2} \) in the re-encryption algorithm. Notice that
\[ R_{i,j}^{(1)} \cdot c_{i,3} + c_{i,1} + R_{i,j}^{(1)} \cdot e'_1 \mod q \]
\[ = R_{i,j}^{(1)} \cdot e_{i,3} + R_{i,j}^{(1)} \cdot e'_1 + e_{i,1} + B_j^T \cdot S_{i,j}^T \cdot s + E_{i,j,1}^T \cdot s \mod q, \]
by Lemma 2 and the property of Gaussian distribution, we get \( R_{i,j}^{(1)} \cdot e_{i,3} + R_{i,j}^{(1)} \cdot e'_1 \leftrightarrow D_s \) with \( s_k' = ||R_{i,j,k'}|| \cdot \sqrt{4/3^2 + 1} \cdot \alpha \) for \( k' \in [m] \). Therefore, by using
\[ \text{We could also put a random tag here. But this tag is irrelevant to ciphertexts.} \]
Lemma 2 again, we get that the distribution of $R^{(1)}_{i,j} \cdot (e_{i,3} + e'_{j}) + e''_j$ is statistically close to $D_{\mathbb{Z}^n \cdot \sqrt{(m+nk) \cdot \gamma^2 \cdot (4 \beta^2 + 1) \cdot 2^\alpha + \omega(\log m)}}$ by noticing that $\mathbb{Z}^n + c'_{j,1} = \mathbb{Z}^n + R^{(1)}_{i,j} \cdot e_{i,3} + R^{(1)}_{i,j} \cdot c'_1$. Therefore, we have

$$c'_{j,1} + e''_j = R^{(1)}_{i,j} \cdot e_{i,3} + R^{(1)}_{i,j} \cdot e'_j + e''_j + e_{i,1} + B_j^T \cdot S_i^T \cdot s + E^T_{i,j,1} \cdot s \mod q$$

for some $e''_j \leftarrow D_{\mathbb{Z}^n \cdot \sqrt{(m+nk) \cdot \gamma^2 \cdot (4 \beta^2 + 1) \cdot 2^\alpha + \omega(\log m)}}$. Similarly,

$$c'_{j,2} + e''_j \approx e''_j + e_{i,1} + D_j^T \cdot S_i^T \cdot s + E^T_{i,j,2} \cdot s + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mu \mod q$$

for some $e''_j \leftarrow D_{\mathbb{Z}^n \cdot \sqrt{(m+nk) \cdot \gamma^2 \cdot (4 \beta^2 + 1) \cdot 2^\alpha + \omega(\log m)}}$. So, we could deduce that

$$-R_j \cdot e_{j,1} + c_{j,2} \mod q = \left\lfloor \frac{q}{2} \right\rfloor \cdot \mu + err_2,$$

where

$$err_2 = -R_j \cdot (E^T_{i,j,1} \cdot s + e_{i,1} + e''_j) + E^T_{i,j,2} \cdot s + e_{i,2} + e''_j \in \mathbb{Z}^n.$$

By Lemma 5, it is easy to bound

$$||err_2||_{\infty} \leq 2 \cdot \omega(\log m) \cdot \sqrt{m} \cdot (1 + \sqrt{n} \cdot \alpha + \sqrt{5} \cdot \sqrt{m} + nk \cdot \beta \cdot \gamma) \cdot \alpha$$

with overwhelming probability. For our choice of $q$, we also $||err||_{\infty} \leq \frac{q}{2}$ and we could recover $\mu$ successfully. Overall, we could deduce the following lemma.

**Lemma 9.** For $q \geq 10 \cdot \omega(\log m) \cdot \sqrt{m} \cdot (1 + \sqrt{n} \cdot \alpha + \sqrt{5} \cdot \sqrt{m} + nk \cdot \beta \cdot \gamma) \cdot \alpha$, our single-hop PRE scheme satisfies correctness.

Next, let’s show the key-privacy and HRA security of our single-hop PRE scheme. The key point of our design to achieve key-privacy is that for the challenge identity $j^*$, the re-encryption key generation algorithm does not use $A_{j^*,1}$ which offers the ability to generate re-encryption keys from $j^*$ to other users. Therefore, it’s enough to change $B_{j^*}$ and $D_{j^*}$ to be uniform, and maintain $A_{j^*}$ unchanged.

**Theorem 1.** Under the Multi-LWE$_{n,q,2m,\alpha}$ assumption for $q \geq 10 \cdot \omega(\log m) \cdot \sqrt{m} \cdot (1 + \sqrt{n} \cdot \alpha + \sqrt{5} \cdot \sqrt{m} + nk \cdot \beta \cdot \gamma) \cdot \alpha$, our single-hop PRE scheme is key private in the standard model.

**Proof.** Assume that there is an adversary $\mathcal{A}$ who could break the key-privacy of our single-hop PRE scheme with advantage $\delta = |\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}|$. We consider the following sequence of games. Assume the sizes of $U_H$ and $U_C$ are $N_1 = \text{poly}(n)$ and $N_2 = \text{poly}(n)$. Notice that by Lemma 6, for any users $i$ and $j$, after we sample $Q = \text{poly}(n)$ re-encryption keys $rk_{i \to j}^{(k)}$ with $k' \in [Q]$, the new sampled re-encryption key satisfies $rk_{i \to j}^{(k)} \neq rk_{i \to j}^{(k')}$ for any $k' \in [Q]$ with overwhelming probability.
- **Game 0**: This is the original key-privacy game. Notice that, in the challenge query, when $b = 0$, the re-encryption key

$$rk_{i^* \rightarrow j^*} = (\{X_{i^* \cdot j^*}; X_{i^* \cdot j^*}'\}_{i^* \in \{1, 2\}, j^* \in \{1, 2, 3\}})$$

of challenge users $i^*$ and $j^*$ are sampled as $(R_{i^* \cdot j^*}^{(1)})^T, (R_{i^* \cdot j^*}^{(2)})^T \leftrightarrow D_{\mathbb{Z}^{m+nk, \gamma}}^{m}$, where $R_{i^* \cdot j^*}^{(1)}, R_{i^* \cdot j^*}^{(2)}$ are defined as those in the ReEnc algorithm.

- **Game 1**: This game is almost identical to Game 0, with the differences that the challenger $\mathcal{C}$ will select a random user $j^* \in U_H$ after the adversary decided the sizes of $U_H$ and $U_C$ in Phase 1, and if the users $(i^*, j^*)$ queried in the challenge oracle query satisfy $j^* \neq j^*$, $\mathcal{C}$ aborts.

- **Game 2**: This game is almost identical to Game 1, with the differences that after the challenger $\mathcal{C}$ decides $j^*$, he chooses $B_{j^*}, D_{j^*} \leftarrow U(\mathbb{Z}_q^{nxm})$.

- **Game 3**: This game is almost identical to Game 2, with the differences that in the challenge oracle query, the syndromes in the generation of re-encryption key $rk_{i^* \rightarrow j^*}$ when $b = 1$ are changed to uniform elements.

- **Game 4**: This game is almost identical to Game 4, with the differences that in the challenge oracle query, $\mathcal{C}$ outputs

$$rk_{i^* \rightarrow j^*} = (\{X_{i^* \cdot j^*}; X_{i^* \cdot j^*}'\}_{i^* \in \{1, 2\}, j^* \in \{1, 2, 3\}})$$

with $(R_{i^* \cdot j^*}^{(1)})^T, (R_{i^* \cdot j^*}^{(2)})^T \leftrightarrow D_{\mathbb{Z}^{m+nk, \gamma}}^{m}$, regardless of the value of $b$.

If no abort occurs, Game 0 and Game 1 are identical. So, the advantage of $\mathcal{A}$ in Game 1. In the following, assume that no abort occurs. We also notice that in the following games, we do not change the generation of $A_{j^*}$ of the challenge user $j^*$, so both re-encryption key generation queries and re-encryption queries from $j^*$ to any user $k \in U_H \cup U_C$ could be answered correctly. We will use $\text{Adv}_i(\mathcal{A})$ to denote the advantage of $\mathcal{A}$ in Game $i$ for $i \in \{1, 2, 3, 4\}$.

Recall that we choose $m \geq 2nk$ for $k = \lceil \log q \rceil$, so $m = m + 2nk \geq 4n \lceil \log q \rceil$ and with overwhelming probability of $B_{j^*} \leftrightarrow U(\mathbb{Z}_q^{nxm})$, the distribution of $(B_{j^*}, B_{j^*} \cdot R_{j^*})$ with $R_{j^*} \leftrightarrow \mathbb{Z}_q^{nxm}$ is within statistical distance $\text{negl}(n)$ of $U(\mathbb{Z}_q^{nxm} \times \mathbb{Z}_q^{nxm})$ due to Lemma 1. So, we have $|\text{Adv}_1(\mathcal{A}) - \text{Adv}_2(\mathcal{A})| \leq \text{negl}(n)$.

In Game 2, the re-encryption key $rk_{i^* \rightarrow j^*} = (\{X_{i^* \cdot j^*}; X_{i^* \cdot j^*}'\}_{i^* \in \{1, 2\}, j^* \in \{1, 2, 3\}})$ is generated satisfying

$$A_{i^* \cdot 1} \cdot \begin{pmatrix} X_{1,1} X_{1,2} X_{1,3} \\ X_{2,1} X_{2,2} X_{2,3} \end{pmatrix} = S_{i^* \cdot j^*, -B_{j^*} + E_{i^* \cdot j^*, -1} - B_{i^*} \mod q}$$

and

$$A_{i^* \cdot 1} \cdot \begin{pmatrix} X_{1,1} X_{1,2} X_{1,3} \\ X_{2,1} X_{2,2} X_{2,3} \end{pmatrix} = S_{i^* \cdot j^*, -D_{j^*} + E_{i^* \cdot j^*, -2} - D_{i^*} \mod q}$$

for some $S_{i^* \cdot j^*} \leftrightarrow \mathbb{Z}_q^{nxn}$ and $E_{i^* \cdot j^*, 1}, E_{i^* \cdot j^*, 2} \leftrightarrow \mathbb{Z}_q^{nxm}$ when $b = 1$. Here, $A_{i^* \cdot 1} = [\hat{A} \cdot R_{i^* \cdot 1} + H_1 \cdot \mathbf{G}]$. Therefore, under the Multi-LWE$_{n, q, 2m, D_{\mathbb{Z}_q}}^{nxn}$ assumption, if we change $S_{i^* \cdot j^*, -B_{j^*} + E_{i^* \cdot j^*, -1} - B_{i^*} \mod q}$ and $S_{i^* \cdot j^*, -D_{j^*} + E_{i^* \cdot j^*, -2} - D_{i^*} \mod q}$, to be $U_1, U_2 \leftrightarrow U(\mathbb{Z}_q^n)$, the adversary could not distinguish the re-encryption
keys generated from the above two cases. Hence, we get $|\text{Adv}_2(\mathfrak{A}) - \text{Adv}_3(\mathfrak{A})| \leq \text{Adv}_3(\text{Multi-LWE}_{n,q,2m,DZ,\alpha})$.

Discussions after Lemma 7 shows that the distribution of the re-encryption key $rk_{i^*} \rightarrow j^*$ in Game 4 is statistically closed to that in Game 3 when $b = 1$. So, $|\text{Adv}_3(\mathfrak{A}) - \text{Adv}_4(\mathfrak{A})| \leq \text{negl}(n)$. Moreover, in Game 4, the distribution of re-encryption $rk_{i^*} \rightarrow j^*$ is identical, regardless of the value of $b$. So, the advantage of $\mathfrak{A}$ in Game 4 is $\frac{1}{2}$. Overall, we deduce that

$$|\frac{\delta}{N_1} - \frac{1}{2}| \leq \text{Adv}_4(\text{Multi-LWE}_{n,q,2m,DZ,\alpha}) + \text{negl}(n),$$

and conclude the desired result.

It is a little more complicated to show the HRA security. The biggest obstacle is how to answer re-encryption queries from $i^*$ to other users (including those corrupted) after we change the generation of $A_{i^*}$. To do so, we add some “back door” (i.e. the $[\bar{A} \cdot R_{i^*} + h(\mu) \cdot G]$-part of $A_{i^*}$) to decrypt-then-encrypt corresponding ciphertexts.

**Theorem 2.** Our single-hop PRE scheme satisfies HRA security under the Multi-LWE_{n,q,2m,DZ,\alpha} assumption and the LWE_{n,q,2m+\bar{m},DZ,\alpha} assumption in the standard model.

**Proof.** In the following, we shall show that any PPT adversary could not distinguish the ciphertext of an arbitrary message (the case $b = 1$) and a random element chosen from the ciphertext space (the case $b = 0$). This implies our desired result. Assume that there is an adversary $\mathfrak{A}$ who could break the HRA security of our single-hop PRE scheme with advantage $\delta = |\text{Pr}[\mathfrak{A} \text{ wins}] - \frac{1}{2}|$. We consider the following sequence of games. Assume the sizes of $U_H$ and $U_C$ are $N_1 = \text{poly}(n)$ and $N_2 = \text{poly}(n)$.

- **Game 0:** This is the original HRA security game.
- **Game 1:** This game is almost identical to Game 0, with the differences that the challenger $\mathcal{C}$ will select a random user $i^* \in U_H$ after the adversary decided the sizes of $U_H$ and $U_C$ in Phase 1, and if the challenge users $(i^*, j^*)$ queried in challenge oracle query satisfies $i^* \neq \bar{i}^*$, $\mathcal{C}$ aborts.

We will also use $\text{Adv}_1(\mathfrak{A})$ to denote the advantage of adversary $\mathfrak{A}$ in Game $i$ for $i \in [8]$, then we have $\text{Adv}_1(\mathfrak{A}) = \frac{\delta}{N_1}$. Assume that in the following games no abort occurs.

- **Game 2:** This game is almost identical to Game 1, with the differences that after the challenger $\mathcal{C}$ decides $\bar{i}^*$, he changes all the $\{B_i, D_i\}$’s of honest users $i \in U_H$ to be uniformly random elements of $\mathbb{Z}_q^{n \times m}$.

We note that the $\{A_i\}$’s of every user $i \in U_H$ is unchanged now, so re-encryption key generation queries, hence re-encryption queries, from user $i \in U_H$ to any user $j \in U_H \cup U_C$ could be answered correctly. By Lemma 1, we have $|\text{Adv}_1(\mathfrak{A}) - \text{Adv}_2(\mathfrak{A})| \leq N_1 \cdot \text{negl}(n)$. 

– **Game 3**: This game is almost identical to **Game 2**, with the differences that when generating re-encryption keys $rk_{i \rightarrow j}$ with $j \in U_H$, we changes the syndromes to be uniformly random elements as in **Game 3** of the proof of Theorem 1.

Since there are at most $N_1 - 1$ such queries, under Multi-LWE$_{n,q,2m,D_{\mathbb{Z}_n}}$ assumption, we get $|\text{Adv}_2(\mathfrak{A}) - \text{Adv}_3(\mathfrak{A})| \leq (N_1 - 1) \cdot \text{Adv}_2(\text{Multi-LWE}_{n,q,2m,D_{\mathbb{Z}_n}})$.

– **Game 4**: This game is almost identical to **Game 3**, with the differences that when generating re-encryption keys $rk_{i \rightarrow j}$ with $j \in U_H$, we just generate the re-encryption key

$$rk_{i \rightarrow j} = \{(X_{i',j'};X'_{i',j'}) | i' \in \{1,2\}, j' \in \{1,2,3\}\}$$

by sampling $(R_{i,j}^{(1)})^T, (R_{i,j}^{(2)})^T \leftrightarrow D_{\mathbb{Z}_m+nk,\gamma}$, where $R_{i,j}^{(1)}, R_{i,j}^{(2)}$ are defined as those in the ReEnc algorithm.

Similar to discussion in Theorem 1, we have $|\text{Adv}_3(\mathfrak{A}) - \text{Adv}_4(\mathfrak{A})| \leq (N_1 - 1) \cdot \text{negl}(n)$. **Notice that, from now on, we do not need to use the trapdoor embedded in $A_{i*}$ to generate re-encryption keys from $i^*$ to honest users.**

– **Game 5**: This game is almost identical to **Game 4**, with the differences that after the challenger $\mathcal{C}$ decides $i^*$, he chooses a $H^* \leftarrow U(\mathbb{S})$ which will be used as the tag of the encryption of the challenge message $\mu^*$, and set $A_{i^*} = [A; A \cdot R_{i^*,1} - H_1 \cdot G; A \cdot R_{i^*,2} - H^* \cdot G]$. Notice that, the public key of $i^*$ is $(A_{i^*}, B_{i^*}, D_{i^*})$ with $B_{i^*}, D_{i^*} \leftarrow U(\mathbb{Z}_{q^m}^{\times})$ now. Meanwhile, the re-encryption queries from $i^*$ to $j \in U_C$ will be replaced as follows.

- When receiving query $(i^*, j \in U_C, (c_1, c_2, c_3; H_e, 1))$, $\mathcal{C}$ will abort if $H_e = H^*$. Otherwise, we have $A_{i^*,1} = [A; A \cdot R_{i^*,1} + (h(\mu) - H^*) \cdot G]$. By Lemma 7, the challenge $\mathcal{C}$ could recover $s$ used in the encryption algorithm from $c_{i,3}$ and $[A; A \cdot R_{i^*,2} + (H_e - H^*) \cdot G]$ with trapdoor $R_{i^*,2}$, since $\frac{q}{\sigma} > \sqrt{n \cdot \log q \cdot \omega(\sqrt{\log n})}$. With the recovered $s$, $\mathcal{C}$ could recover errors $e_{i,1}, e_{i,2}, e_{i,3}$, as well as $\mu$, by our choices of parameters. Therefore, the challenger $\mathcal{C}$ could sample $\mathcal{S}_{i^*,j} \leftarrow \chi^{n \times n}, \mathcal{E}_{i^*,j,1}, \mathcal{E}_{i^*,j,2} \leftarrow \chi^{n \times m}$ and $e_{i^*,1}^m, e_{i^*,2}^m \leftarrow D_{\mathbb{Z}_{2^{\beta + 1} \cdot (4\beta^2 + 1) \gamma^2 / \omega(\sqrt{\log m})}}$. Then, he returns $(c_{i,1}, c_{i,2}, \perp; \mu, 2)$ to $\mathfrak{A}$, where

$$c_{i,1} = e_{i,1}^m + e_{i,1} + B_{i^*,j} T \cdot S_{i^*,j} ^T \cdot s + E_{i^*,j,1} T \cdot s \mod q$$

and

$$c_{i,2} = e_{i,2}^m + e_{i,2} + D_{i^*,j} T \cdot S_{i^*,j} ^T \cdot s + E_{i^*,j,2} T \cdot s \mod q.$$

Assume that there is at most $Q = \text{poly}(n)$ queries of the encryption oracle, then the probability that $\mathcal{C}$ aborts in the above modified re-encryption query oracle is less than $\frac{Q}{\text{negl}(n)}$. Since in our security game, encryptions are guaranteed to be honest-generated. As long as no abort occurs, the outputted re-encryptions are distributed statistically close to a real re-encrypted ciphertext according to our analysis in the correctness (above Lemma 9). So, we have $|\text{Adv}_4(\mathfrak{A}) - \text{Adv}_5(\mathfrak{A})| \leq \text{negl}(n)$. 

Title Suppressed Due to Excessive Length
Game 6: This game is almost identical to Game 5, with the differences that we change the generation of the challenge ciphertext when \( b = 1 \) as follows.

- To encrypt the a message \( \mu^* \), \( \mathcal{C} \) first samples \( s \leftarrow \chi^n \), \( e_{i,1}, e_{i,2} \leftarrow \chi^m \) and \( e_{i,3} \leftarrow \chi^m \). Then, he computes \( c_{i,1} = B_i^T \cdot s + e_{i,1} \mod q \), \( c_{i,2} = D_i^T \cdot s + e_{i,2} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mu^* \mod q \) and \( c^* = A_i^T \cdot s + e_{i,3} \mod q \). Finally, he computes

\[
c_{i,3} = \text{ReRand}([I_{m \times m}|R_i^x|D_i^2], c^*, \alpha, \beta)
\]

by using Lemma 3, and returns \( (c_{i,1}, c_{i,2}, c_{i,3}; H^*, 1) \) to \( \mathfrak{A} \).

Recall that the challenge ciphertext is of the form \( c_{i,1} = B_i^T \cdot s + e_{i,1} \mod q \), \( c_{i,2} = D_i^T \cdot s + e_{i,2} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mu^* \mod q \), and

\[
c_{i,3} = [A][A \cdot R_i^x, 1]A \cdot R_i^x, 2]T \cdot s + e_{i,3} \mod q
\]

for the case \( b = 1 \) in Game 5. Here, \( s \leftarrow \chi^n \), \( e_{i,1}, e_{i,2} \leftarrow \chi^m \) and \( e_{i,3} \leftarrow D_{2m, 2, \beta, 0} \). According to our choice of parameters and Lemma 4, we have \( \beta \geq 1 + \log \frac{1}{\varepsilon} (|I_{m \times m}|R_i^x, 2|) \geq 1 + \log \frac{1}{\varepsilon} (|I_{m \times m}|R_i^x, 2|) \) with overwhelming probability. So, Lemma 3 implies that the distribution of \( c_{i,3} \) in Game 5 is statistically closed to the distribution of \( c_{i,3} \) generated in the above modified encryption algorithm in Game 6. Therefore, we have \( \| \text{Adv}_5(\mathcal{A}) - \text{Adv}_6(\mathcal{A}) \| \leq \text{negl}(n) \).

Game 7: This game is almost identical to Game 6, with the differences that we change the generation of challenge ciphertext when \( b = 1 \) as follows.

To encrypt the challenged message \( \mu^* \), \( \mathcal{C} \) sets \( c_{i,1} = u_1 \leftarrow U(Z_q^n) \), \( c_{i,2} = u_2 + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mu^* \) with \( u_2 \leftarrow U(Z_q^n) \) and \( c^* = u_3 + e_{i,3} \) with \( u_3 \leftarrow U(Z_q^n) \) and \( e_{i,3} \leftarrow \chi^m \). He computes \( c_{i,3} = \text{ReRand}([I_{m \times m}|R_i^x, 1]R_i^x, 2, 2], c^*, \alpha, \beta) \) and returns \( (c_{i,1}, c_{i,2}, c_{i,3}; H^*, 1) \) to \( \mathfrak{A} \) in the end.

The main difference between Game 7 and Game 6 is the distributions of \( c_{i,1}, c_{i,2} \) and \( c^* \), that differ by some LWE samples. Hence, it is easy to verify that \( \| \text{Adv}_6(\mathcal{A}) - \text{Adv}_7(\mathcal{A}) \| \leq \text{Adv}_\mathcal{A}(\text{LWE}_{n, q, 2m + \tilde{m}, D_{\mathcal{E}, 0}}) \).

Game 8: This game is almost identical to Game 7, with the differences that we change the generation of challenge ciphertext when \( b = 1 \) as follows.

To encrypt the challenged message \( \mu^* \), \( \mathcal{C} \) sets \( c_{i,1}, c_{i,2}, c_{i,3} \leftarrow U(Z_q^n) \). Notice that we have that \( c_{i,3} = [I_{m \times m}|R_i^x, 1]R_i^x, 2]^T \cdot u_3 + e' \) for some \( e' \leftarrow D_{2m, 2, \beta, 0} \) in Game 7 by Lemma 3. Meanwhile, the entire view of \( \hat{A}, u; A \cdot R_i^x, k' \cdot u^T \cdot R_i^x, k \) \( \approx (\hat{A}, u; U', (u')^T) \approx (A, u; A \cdot R_i^x, k' \cdot (u')^T) \) for \( k' \in [2] \), \( U' \leftarrow U(Z_q^n) \), \( u \leftarrow U(Z_q^n) \) and \( u' \leftarrow U(Z_q^n) \) by Lemma 1. Therefore, we have \( \| \text{Adv}_7(\mathcal{A}) - \text{Adv}_8(\mathcal{A}) \| \leq \text{negl}(n) \).

Finally, notice that we have \( \| \text{Adv}_8(\mathcal{A}) \| = \frac{1}{2} \). We could deduce that

\[
\frac{1}{N_1} - \frac{1}{2} \leq \text{negl}(n) + (N_1 - 1) \cdot \text{Adv}_\mathcal{A}(\text{Multi-LWE}_{n, q, 2m, D_{\mathcal{E}, 0}}) + \text{Adv}_\mathcal{A}(\text{LWE}_{n, q, 2m + \tilde{m}, D_{\mathcal{E}, 0}})
\]

The proof is finished.
Remark 5. We note that for our constructions of single-hop PRE schemes, the encryption oracle only needs to record encryptions of the challenge user $i^*$ in the security game. I.e. we only need to make sure that ciphertexts of the user $i^*$ are not malformed. This is because by our constructions, the adversary could not re-encrypt a level 2 ciphertext of an honest user to any other corrupted users, since the form of level 2 ciphertext is different from those of fresh ciphertexts.

Remark 6. Since our constructions satisfy the requirements ((weakly) key privacy, source hiding, indistinguishability) proposed in [13], our single-hop PRE schemes are also adaptive secure for some directed acyclic graphs with polynomially bounded parameters by Theorem 6 of [13].

Remark 7. As [28], our single-hop PRE scheme could be easily modified to a PRE+ scheme, in which the re-encryption keys of a ciphertext of any message could be generated by its encrypter. For LWE-based constructions, the intuition is quite easy. Any encrypter of a message $m$ knows the ephemeral secrets $s$ and $e$'s. So, the corresponding re-encryption key generation algorithm and the re-encryption algorithm could be designed as follows.

- \textbf{ReKeyGen}(pp, pk_i, pk_j, ct_i; eph): Sample $e_{i,j,1}, e_{i,j,2} \leftarrow \chi^m$, then return $rk_{i\rightarrow j}(ct_i) = ((B_j - B_i) \cdot s + e_{i,j,1}; (D_j - D_i) \cdot s + e_{i,j,2})$.
- \textbf{ReEnc}(pp, ct_i, rk_{i\rightarrow j}(ct_i)): Parse $rk_{i\rightarrow j}(ct_i) = (rk_1; rk_2)$, then return $c_{j,1} = c_{i,1} + rk_1$ and $c_{j,2} = c_{i,2} + rk_1$.

Here, $eph$ represents the ephemeral secret $s$.

Recall that as discussed in Remark 1, our single-hop PRE schemes also satisfies the second-level security of ciphertexts [12]. In the above security proof, the challenger has no ability to answer re-encryption queries of non-challenge ciphertexts with tag $H^*$ from $i^*$ to corrupted users. By our definition of security, ciphertexts of $i^*$ are generated in the encryption oracle. So, this case will happen with negligible-probability. While in [12], this case is avoided by adding an additional not-derivative requirement [12, Definition 3.2]. Meanwhile, we note that in the secure proof of [12, Theorem 4.2], the distributions of re-encryption keys between $(i^*, j)$ with $j \in U_H$ and $(i, j)$ with $i \in U_H \setminus \{i^*\}$ and $j \in U_H \cup U_C$ are different from each other (due to the quality of trapdoors used for sampling pre-images), which may result in that the norms of re-encryption keys of the later case would be larger than the desired (Gaussian) bound as in the real schemes. This means that the statistical distance of Hybrid $H_0$ and Hybrid $H_1$ in the security proof of [12] is not negligible, so the adversary may output $\perp$ indicating that he detects the scheme is different from the real scheme. The somewhat trivial extensions of our single-hop constructions to multi-hop cases suffer similar problem.

Acknowledgement: The authors would like to express their gratitude to Lisha Yao for pointing out a flaw in the early version of this paper. The authors are supported by the National Key Research and Development Program of China (Grant No. 2018YFA0704702), and the National Natural Science Foundation of China (Grant No. 61672019).
References


