Title	Proxy Re-Encryption based on Learning with Errors (Mathematical Foundation of Algorithms and Computer Science)
Author(s)	Xagawa, Keita; Tanaka, Keisuke
Citation	数理解析研究所講究録 (2010), 1691: 29-35
Issue Date	2010-06
URL	http://hdl.handle.net/2433/141576
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

数理解析研究所講究録
第1691巻 2010年 29-35

格子に基づく代理人再暗号方式

Proxy Re-Encryption based on Learning with Errors

草川 恵太
Keita Xagawa *

田中 圭介
Keisuke Tanaka *

Abstract— Proxy re-encryption enables a proxy to convert a ciphertext for some user to a ciphertext for another user, but a proxy cannot learn information of messages. All of the proxy re-encryption and identity-based proxy re-encryption schemes are based on the number-theoretic assumptions. This paper proposed proxy re-encryption schemes based on the learning with errors problem. They are first schemes based on combinatorial problems.

Keywords: proxy re-encryption, learning with errors, lattice problems.

1 Introduction

Suppose that Alice wants to forward a received encrypted e-mail to Bob in the public channel. She decrypts it by her secret key, encrypts the message with Bob's public key, and sends it to him. However, decryption and encryption are costly for her mobile phone in general. Therefore, she wants a mail server to forward her mail to Bob automatically. In this case, she does not trust the server, hence, she does not want to give her secret key to the server. The one of solutions is proxy re-encryption [3].

In a proxy re-encryption (PRE) scheme, the server is given a re-encryption key $rk_{A\leftrightarrow B}$ between Alice and Bob. The server, given a ciphertext ct_A for Alice, can convert it to a ciphertext ct_B for Bob by using the re-encryption key $rk_{A\leftrightarrow B}$ and without decrypting ct_A. In addition, proxy re-encryption ensures that even if the server knows $rk_{A\leftrightarrow B}$, it cannot learn the message of ct_A.

The study of proxy re-encryption is initiated by Blaze, Bleumer, and Strauss [3]. They formalize a proxy re-encryption and gave an example based on the ElGamal encryption scheme. There are several proxy re-encryption schemes [3, 2, 5, 10, 7, 1, 11] and identity-based proxy re-encryption schemes [12, 9, 6] in the literature. However, their underlying problems are the decisional Diffie-Hellman problem or its variants.

In this paper, we propose proxy re-encryption schemes based on other problems, the learning with errors and lattice problems. Our constructions are obtained by extending Regev's encryption scheme [14].

Ideas from the ElGamal-based PRE: We note that some lattice-based cryptosystems have similar structure on the DDH-based cryptosystems while inherent noises of lattice-based cryptosystems disturb the structure.

Consider the ElGamal encryption scheme over $\mathbb{G} = \langle g \rangle$ with order a large prime q. The key pair is $(x, y = g^x)$ for randomly chosen x. The ciphertext of $m \in \mathbb{G}$ under the encryption key y is $(g^k, m \cdot y^k)$ for randomly chosen k. Let $(x_A, y_A = g^{x_A})$ and $(x_B, y_B = g^{x_B})$ denote Alice's and Bob's key pair, respectively. Assume that the proxy has the re-encryption key $r_{A\leftrightarrow B} = x_A - x_B$ and has the ciphertext (c_1, c_2) to be converted. Then, the conversion is done by

$$\begin{aligned}(c_1', c_2') &= (c_1, c_2 \cdot c_1^{-r_{A\leftrightarrow B}}) \\ &= (g^k, m \cdot g^{kx_A} \cdot g^{k(x_B - x_A)}) = (g^k, m \cdot y_B^k).\end{aligned}$$

It can be shown that this proxy re-encryption scheme is based on the hardness of the DDH problem.

We here recall Regev's encryption scheme. The key pair is computed by $(\boldsymbol{s}, (\boldsymbol{A}, \boldsymbol{p} = \boldsymbol{s}^T\boldsymbol{A} + \boldsymbol{x}))$, where $\boldsymbol{s} \in \mathbb{Z}_q^n$, $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, $\boldsymbol{x} \in \mathbb{Z}_q^{1\times m}$ and the magnitudes of the elements of $\boldsymbol{x}$ are relatively smaller than $q/4m$, say the ℓ_1-norm of $\boldsymbol{x}$ is at most $q/4$. The encryption of the message $msg \in \{0,1\}$ under the encryption key $(\boldsymbol{A}, \boldsymbol{p})$ is $(\boldsymbol{u}, v) = (\boldsymbol{Ae}, \boldsymbol{pe} + msg\lfloor q/2 \rfloor)$, where $\boldsymbol{e} \leftarrow \{0,1\}^m$.

The decryption procedure is as follows: (1) compute $d = v - \boldsymbol{s}^T\boldsymbol{u}$ and (2) output 0 if the absolute value of d is at most $q/4$ and output 1 otherwise.

Let $(\boldsymbol{s}_A, (\boldsymbol{A}_A, \boldsymbol{p}_A = \boldsymbol{s}_A^T\boldsymbol{A}_A + \boldsymbol{x}_A))$, and $(\boldsymbol{s}_B, (\boldsymbol{A}_B, \boldsymbol{p}_B = \boldsymbol{s}_B^T\boldsymbol{A}_B + \boldsymbol{x}_B))$ denote Alice's and Bob's key pair, respectively. Let $\boldsymbol{r}_{A\leftrightarrow B} = \boldsymbol{s}_A - \boldsymbol{s}_B$. Then, the conversion from $(\boldsymbol{u}, v_A)$ to $(\boldsymbol{u}, v_B)$ is done by $(\boldsymbol{u}, v_B) = (\boldsymbol{u}, v_A - \boldsymbol{r}_{A\leftrightarrow B}^T\boldsymbol{u})$,

* Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, W8-55, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan. {xagawa5,keisuke}@is.titech.ac.jp. This research was supported in part by NTT Information Sharing Platform Laboratories, JSPS Global COE program "Computationism as Foundation for the Sciences," and KAKENHI No.19-55201.

which is similar to that of the ElGamal-based proxy re-encryption scheme. The decryption by Bob works correctly since

$$d_B = v_B - \boldsymbol{s}_B^T\boldsymbol{u} = v_A - (\boldsymbol{s}_A - \boldsymbol{s}_B)^T\boldsymbol{u} - \boldsymbol{s}_B^T\boldsymbol{u} = v_A - \boldsymbol{s}_A^T\boldsymbol{u} = d_A.$$

The proof strategy for security is also similar to that of the ElGamal-based proxy re-encryption scheme.

2 Preliminaries

A security parameter is denoted by n. We use the standard O-notation. The function $f(n)$ is said to be negligible if $f(n) = n^{-\omega(1)}$. For a distribution χ, we often write $x \leftarrow \chi$ which indicates that we take a sample x from χ.

The leftover hash lemma often appears in the context of lattice-based cryptography. We summarize the arguments which appeared in many papers on lattice-based cryptography. See [14] for the proof.

Lemma 2.1 (The uniformity lemma for lattice-based hash functions). *Consider* $\mathcal{H} = \{h_{\boldsymbol{A}} : \{0,1\}^m \to \mathbb{Z}_q^{n+\ell} \mid \boldsymbol{A} \in \mathbb{Z}_q^{(n+\ell)\times m}\}$, *where* $h_{\boldsymbol{A}}(\boldsymbol{e}) = \boldsymbol{A}\boldsymbol{e}$. *Let* H *be the uniform distribution over* $\mathcal{H}$, *and* X *and* U *random variables distributed uniformly over* $\{0,1\}^m$ *and* $\mathbb{Z}_q^{n+\ell}$, *respectively. Applying the variant of the leftover hash lemma, we have*

$$\Pr_H[\Delta(H(X), U) \geq 2^{-\frac{1}{4}(m-(n+\ell)\log q)}] \leq 2^{-\frac{1}{4}(m-(n+\ell)\log q)}.$$

3 Proxy Re-Encryption

In this paper, we consider *bidirectional* and *multi-hop* proxy re-encryption. A PRE scheme is called bidirectional, if a proxy has a re-encryption key $rk_{i\leftrightarrow j}$, it can convert a ciphertext for the user i to a ciphertext for the user j, vice versa. A PRE scheme is said to be multi-hop, a proxy can re-encrypt a ciphertext for the user i into a ciphertext for the user j and it can re-encrypt that into one for the user k and so on.

3.1 Model of Proxy Re-Encryption

A PRE scheme PRE is a sextuplet of algorithms:

Setup(1^n): The setup algorithm, given the security parameter n, outputs parameters *param*.

Reg$(param, i)$: The registration algorithm, given the parameters *param* and a user identity i, outputs the pair of an encryption key and a decryption key (ek_i, dk_i).

ReKeyGen(dk_i, dk_j): The re-encryption key generation algorithm, given two decryption keys dk_i and dk_j, outputs a re-encryption key $rk_{i,j}$.

Enc$(param, ek_i, msg)$: The encryption algorithm, given the parameters *param*, the encryption key ek_i of the user i, and a message *msg*, outputs a ciphertext ct_i.

ReEnc$(rk_{i,j}, ct_i)$: The re-encryption algorithm, given the re-encryption key $rk_{i,j}$ between the users i and j, and a ciphertext ct_i for the user i, it outputs a ciphertext ct_j for the user j.

Dec(dk, ct): The decryption algorithm, given the decryption key dk and the ciphertext ct, outputs a plaintext *msg*.

Our definition of correctness is slightly weaker than the standard one [5]. We say a PRE scheme PRE is correct if an underlying public-key encryption scheme PKE = (Setup, Reg, Enc, Dec) is correct. Formally, it holds that if for any valid *msg*, there exists some negligible function negl(n) such that for any i

$$\Pr\left[msg \neq \widetilde{msg} : \begin{array}{l} param \leftarrow \mathsf{Setup}(1^n); \\ (ek_i, dk_i) \leftarrow \mathsf{Reg}(param, i); \\ ct \leftarrow \mathsf{Enc}(param, ek_i, msg); \\ \widetilde{msg} \leftarrow \mathsf{Dec}(dk_i, ct); \end{array}\right] \leq \mathsf{negl}(n).$$

Additionally, we say a PRE scheme PRE is multi-hop correct if for any valid *msg* and for any integer $k > 1$, one can correctly decrypt the ciphertext of *msg* converted k times into *msg*, that is,

$$\Pr\left[msg \neq \widetilde{msg} : \begin{array}{l} param \leftarrow \mathsf{Setup}(1^n); \\ (ek_i, dk_i) \leftarrow \mathsf{Reg}(param, i); \\ rk_{i\leftrightarrow i+1} \leftarrow \mathsf{ReKeyGen}(dk_i, dk_{i+1}); \\ ct_1 \leftarrow \mathsf{Enc}(param, ek_1, msg); \\ ct_{i+1} \leftarrow \mathsf{ReEnc}(rk_{i\leftrightarrow i+1}, ct_i); \\ \widetilde{msg} \leftarrow \mathsf{Dec}(dk_k, ct_k); \end{array}\right] \leq n^{-\omega(1}$$

where i runs from 1 to k.

3.2 IND-PRE-CPA Security

We describe the formal definition of CPA security of proxy re-encryption, denoted by IND-PRE-CPA. Consider the following experiment $\mathbf{Exp}^{\text{ind-pre-cpa}}_{\text{PRE},\mathcal{A}}(n)$ between the challenger C and the adversary $\mathcal{A}$.

Setup: The challenger takes a security parameter n. It sets $HU, CU \leftarrow \emptyset$, runs the algorithm Setup with 1^n, and obtains parameters *param*, where HU and CU denote the sets of honest users and corrupted users, respectively. It gives $\mathcal{A}$ the parameters *param*.

Challenge Phase: In this phase, the adversary issues queries to the following oracles in any order and many times except to the constraint in the oracle CHALLENGE.

- The oracle INIT receives an index i. If $i \in HU \cup CU$ then it returns $\perp$. Otherwise, it obtains $(ek_i, dk_i) \leftarrow \mathsf{Reg}(param, i)$, adds i to HU, and provides $\mathcal{A}$ with ek_i.
- The oracle CORR receives an index i. If $i \in HU \cup CU$ then it returns $\perp$. Otherwise, it generates $(ek_i, dk_k) \leftarrow \mathsf{Reg}(param; r_i)$, adds i to CU, and provides $\mathcal{A}$ with (ek_i, dk_i) and r_i.
- The oracle REKEY receives two indices $i, j \in HU \cup CU$. If $i, j \in HU$ or $i, j \in CU$ returns $rk_{i \leftrightarrow j} \leftarrow \mathsf{ReKeyGen}(dk_i, dk_j)$. Otherwise, the oracle returns $\perp$.
- The oracle REENC receives two indices $i, j \in HU \cup CU$ and a ciphertext ct. If $i, j \in HU$ or $i, j \in CU$, then it obtains $rk_{i \leftrightarrow j} \leftarrow \mathsf{ReKey}(dk_i, dk_j)$, obtains $\widetilde{ct} \leftarrow \mathsf{ReEnc}(param, rk_{i \leftrightarrow j}, ct)$, and provides $\mathcal{A}$ with the new ciphertext $\widetilde{ct}$. Otherwise, the oracle returns $\perp$.
- The oracle CHALLENGE can be queried only once. This oracle receives two plaintexts msg_0, msg_1 and a target user i^*. If i^* is not in HU then it provides $\perp$ with the challenger and C outputs 0 and halts. Otherwise, the oracle flips a coin $b \in \{0, 1\}$, sets the target ciphertext to be $ct^* \leftarrow \mathsf{Enc}(ek_{i^*}, msg_b)$, and sends ct^* to the adversary and b to the challenger.

Guessing Phase: Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. If $b' = b$, the challenger outputs 1, otherwise 0.

Definition 3.1 (IND-PRE-CPA security). Let PRE be a PRE scheme, $\mathcal{A}$ an adversary, and n a security parameter. We define the advantage of $\mathcal{A}$ as

$$\mathbf{Adv}^{\text{ind-pre-cpa}}_{\mathsf{PRE},\mathcal{A}}(n) = \left| 2 \Pr\left[\mathbf{Exp}^{\text{ind-pre-cpa}}_{\mathsf{PRE},\mathcal{A}}(n) = 1 \right] - 1 \right|.$$

We say that PRE is IND-PRE-CPA secure if $\mathbf{Adv}^{\text{ind-pre-cpa}}_{\mathsf{PRE},\mathcal{A}}(\cdot)$ is negligible for every polynomial-time adversary $\mathcal{A}$.

Since we only consider IND-PRE-CPA security, we prohibit the adversary to re-encrypt ciphertexts from an honest user to a corrupted user. This is because that this access can simulates a decryption oracle of the honest user.

4 Learning with Errors

The learning with errors (LWE) problem is a generalization of the learning parity noise (LPN) problem, proposed by Regev [14].

We recall the definitions of the distributions appearing the definition of the LWE problem and lattice-based cryptosystems. Later, we define two versions of the LWE problem.

The Gaussian distribution with mean 0 and variance σ^2, denoted by $N(0, \sigma^2)$, is defined by the density function $\frac{1}{\sigma\sqrt{2\pi}} \cdot \exp(-x^2/2\sigma^2)$ over $\mathbb{R}$. By the tail inequality, we have $\Pr[|x| \geq t\sigma] \leq \frac{1}{t} \cdot \exp(-t^2/2)$, where $x \leftarrow N(0, \sigma^2)$.

For $\alpha \in (0, 1)$, Ψ_α denotes the folded Gaussian distribution over $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ [14], obtained by (1) take a sample x from $N(0, \alpha^2/2\pi)$ and (2) output $x \bmod 1$. We have $\Pr_{x \leftarrow \Psi_\alpha}[|x| \geq t] \leq \frac{\alpha}{\sqrt{2\pi}t} \cdot \exp(-\pi t^2/\alpha^2)$ by simple calculations. Often, we set t a constant and $\alpha = 1/\omega(\sqrt{\log n})$ to ensure that the right hand side is negligible in n.

For any probability distribution ϕ over $\mathbb{T}$ and an integer $q \in \mathbb{N}$, $\bar{\phi}$ denotes the discretization of ϕ over $\mathbb{Z}_q$; the distribution is defined by the following procedure: (1) take a sample $x \leftarrow \phi$ and (2) output $\lfloor qx \rceil \bmod q$.

For $\boldsymbol{s} \in \mathbb{Z}_q^n$ and a distribution χ over $\mathbb{Z}_q$, let $A_{\boldsymbol{s},\chi}$ be a distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ defined as follows: (1) take samples $\boldsymbol{a} \leftarrow \mathbb{Z}_q^n$ and $x \leftarrow \chi$ and (2) output $(\boldsymbol{a}, \boldsymbol{s}^T\boldsymbol{a} + x)$.

For simplifying expressions, we define $A_{S,\chi}$ for a matrix $S \in \mathbb{Z}_q^{n \times l}$ as follows: (1) take samples $\boldsymbol{a} \leftarrow \mathbb{Z}_q^n$ and $\boldsymbol{x} \leftarrow \chi^l$ and (2) output $(\boldsymbol{a}, S^T\boldsymbol{a} + \boldsymbol{x})$.

The (search) LWE problem with respect to q and χ, denoted by $\mathrm{sLWE}(q, \chi)$, is finding $\boldsymbol{s} \in \mathbb{Z}_q^n$ given oracle access to $A_{\boldsymbol{s},\chi}$.

For an integer $q = q(n)$ and a distribution χ over $\mathbb{Z}_q$, the (decision) learning with errors problem $\mathrm{dLWE}(q, \chi)$ is distinguishing the oracle $A_{\boldsymbol{s},\chi}$ from the oracle $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ for a uniformly random $\boldsymbol{s} \in \mathbb{Z}_q^n$.

Note that an adversary $\mathcal{A}$ distinguishing $A_{\boldsymbol{s},\chi}$ and $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ with advantage ϵ implies an adversary distinguishing $A_{S,\chi}$ and $U(\mathbb{Z}_q^n \times \mathbb{Z}_q^l)$ for $S \leftarrow \mathbb{Z}_q^l$ with advantage ϵ/l. The proof is simply obtained by the hybrid lemma [13].

5 Proxy Re-Encryption Schemes

We employ the variant by Peikert, Vaikuntanathan, and Waters [13] of Regev's public-key encryption scheme [14]. The main algorithms are the same as those in the PVW scheme. We add to it a re-encryption key generation algorithm and a re-encryption algorithm appeared in Section 1.

5.1 Our First Construction

Our PRE scheme LWEPRE is defined as follows:

Setup(1^n): Given a security parameter n, it outputs $\perp$ as *param*.

Reg$(\perp, i)$: It generates $A_i \leftarrow \mathbb{Z}_q^{n\times m}$, $S_i \leftarrow \mathbb{Z}_q^{n\times l}$, and $X_i \leftarrow \chi^{l\times m}$, and computes $P_i = S_i^T A_i + X_i \in \mathbb{Z}_q^{l\times m}$. It outputs $ek_i = (A_i, P_i)$ and $dk_i = S_i$.

ReKeyGen$(dk_i = S_i, dk_j = S_j)$: It outputs $R_{i\mapsto j} = S_i - S_j \in \mathbb{Z}_q^{n\times l}$.

Enc$(ek = (A, P), w)$: The message space is $\mathbb{Z}_p^l$. It, given w, computes $t = t(w) \in \mathbb{Z}_q^l$, where $t(w) = \lfloor wq/p \rceil \in \mathbb{Z}_q$ and chooses a vector $e \leftarrow \{0,1\}^m \subset \mathbb{Z}_q^m$ uniformly at random. It outputs a pair $(u, v) = (Ae, Pe + t) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^l$ as a ciphertext.

ReEnc$(rk_{i\mapsto j} = R_{i\mapsto j}, (u, v_i))$: It computes $v_j = v_i - R_{i\mapsto j}^T u$ and outputs (u, v_j).

Dec$(dk = S, (u, v))$: It computes $d = v - S^T u \in \mathbb{Z}_q^l$ and outputs the plaintext $w \in \mathbb{Z}_p^l$ such that $d - t(v) \in \mathbb{Z}_q^l$ is closest to $\mathbf{0}$.

The parameters setting for correctness appeared in [13].

Theorem 5.1 (Correctness [13]). *Let $\chi = \bar{\Psi}_\alpha$. Let $q \geq 4pm$, let $\alpha \leq 1/(p\sqrt{m}\cdot g(n))$ for any $g(n) = \omega(\sqrt{\log n})$. Then, the above scheme is correct.*

The multi-hop correctness is easily derived by the correctness.

Theorem 5.2 (Multi-hop correctness). *Let q, α, and g be as in the above. Then, the above scheme is multi-hop correct.*

Proof. Consider the users $1, \ldots, k$. Suppose that (u, v_1) is the valid ciphertext under the encryption key (A_1, P_1) of the user 1 and the re-encryption procedure is performed from 1 to k through $2, \ldots, k-1$. By the re-encryption procedures, we have that

$$v_k = v_1 - \sum_{i=1}^{k-1} R_{i\mapsto i+1}^T u = v_1 - \sum_{i=1}^{k-1} (S_i - S_{i+1})^T u = v_1 - (S_1 - S_k)^T u,$$

where S_i denotes the decryption key of the user i. In the decryption procedure by the user k, d_k is computed as follows:

$$d_k = v_k - S_k^T u = v_1 - (S_1 - S_k)^T u - S_k^T u = v_1 - S_1^T u.$$

So, we have that $d_k = d_1$. Therefore, the multi-hop correctness follows from Theorem 5.1 straightforwardly. □

The security of the scheme is based on the dLWE assumption.

Theorem 5.3 (Security). *Let $m \geq 5(n + l)\log q$. The above scheme is IND-PRE-CPA secure if* dLWE(q, χ) *is hard on average.*

Proof. It follows by combining the claims below. □

Sequence of games: We define the sequence of the games and bound the distance between the games.

Game_0: The original IND-PRE-CPA game. First, the challenger feeds $\perp$ to the adversary. The challenger simulates the oracles in the challenge phase. If the oracle CHALLENGE receives (i^*, w_0, w_1), it flips a coin $b \in \{0,1\}$ and returns the target ciphertext $(u^*, v^*) = (A_{i^*} e^*, P_{i^*} e^* + t(w_b))$, where $e^* \leftarrow \{0,1\}^m$. Finally, the adversary outputs a guess b'. If $b = b'$, then the challenger outputs 1, otherwise 0.

Game_1: We modify the above game, by changing the generation methods of keys. At the beginning of the challenge phase, the challenger first generates re-encryption keys $R_{1\mapsto j} \leftarrow \mathbb{Z}_q^{n\times l}$ for $j = 2, \ldots, Q$. The other re-encryption key $R_{i\mapsto j}$ is computed by $R_{i\mapsto j} = R_{1\mapsto l} - R_{1\mapsto i}$. Next it chooses $S_1 \leftarrow \mathbb{Z}_q^{n\times l}$, $A_1 \leftarrow \mathbb{Z}_q^{n\times m}$, and $X_1 \leftarrow \chi^{l\times m}$, and computes $P_1 = S_1^T A_1 + X_1$. If INIT is called with an input i, the challenger chooses $A_i \leftarrow \mathbb{Z}_q^{n\times m}$, and $X_i \leftarrow \chi^{l\times m}$, and computes $P_i = S_1^T A_i - R_{1\mapsto i}^T A_i + X_i$. If REKEY is called with $i, j \in HU$, then it returns $R_{i\mapsto j}$. If REENC is called with $i, j, (u, c)$, then it uses the re-encryption key $R_{i\mapsto j}$ to re-encrypt the ciphertext. The other conditions are the same as in the original game, Game_0.

Game_2: We replace the generation method of keys. The challenger queries to the oracle $A_{S,\chi}$ and obtains Qm samples $(\bar{A}, \bar{P}) \in \mathbb{Z}_q^{n\times Qm} \times \mathbb{Z}_q^{l\times Qm}$. Then, it chops into $(\bar{A}_i, \bar{P}_i) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{l\times m}$ for $i = 1, \ldots, Q$. It sets $(A_1, P_1) = (\bar{A}_1, \bar{P}_1)$ and $(A_i, P_i) = (\bar{A}_i, \bar{P}_i - R_{1\mapsto i}^T \bar{A}_i)$. The other conditions are the same as in the previous game, Game_1.

Game_3: We replace the oracle $A_{S,\chi}$ with $U(\mathbb{Z}_q^n \times \mathbb{Z}_q^l)$. Hence, the challenger obtains Qm samples (A, P) from $U(\mathbb{Z}_q^n \times \mathbb{Z}_q^l)$ at first. Now, P is chosen uniformly at random.

Let S_i denote the event that the adversary wins, i.e., $b' = b$ in the game Game_i. We denote by $\mathbf{Adv}_{\mathsf{LWEPRE},\mathcal{A}}^{\text{ind-pre-cpa}}(n)$ the advantage of the adversary $\mathcal{A}$ in the IND-PRE-CPA game with the security parameter n. By definition, we have that $\mathbf{Adv}_{\mathsf{LWEPRE},\mathcal{A}}^{\text{ind-pre-cpa}}(n) = |2\Pr[S_0] - 1| = |\Pr[S_0] - \Pr[S_1]|$.

Claim 5.4. Game$_0$ *and* Game$_1$ *are identical.*

Proof. Recall that $R_{i\leftrightarrow j} = S_i - S_j$ by the definition. Hence, we have that $R_{i\leftrightarrow j} = R_{1\leftrightarrow j} - R_{1\leftrightarrow i}$ in Game$_0$. This calculation corresponds to the computation of $R_{i\leftrightarrow j}$ in Game$_1$.

Additionally, in Game$_1$ we have $S_i = S_1 - R_{1\leftrightarrow i}$ imaginary, since $P_i = (S_1 - R_{1\leftrightarrow i})^T A_i + X_i$. Therefore, two games are identical. □

Claim 5.5. Game$_1$ *and* Game$_2$ *are identical.*

Proof. In Game$_1$, we have that $P_i = S_i^T A_i + X_i - R_{1\leftrightarrow i}^T A_i$.

In Game$_2$, we have that $P_i = \bar{P}_i - R_{1\leftrightarrow i}^T A_i$. Since the samples from A_{S,Ψ_α} is $(\bar{A}, \bar{P} = S^T\bar{A} + \bar{X})$, we conclude that two games are identical. □

Claim 5.6. Game$_2$ *and* Game$_3$ *are computationally indistinguishable if* dLWE(q,χ) *is hard on average.*

Proof. Notice that in both games, the challenger does not know the secret keys of the honest users. Hence, if Game$_2$ and Game$_3$ differs computationally, one can distinguish $A_{S,\chi}$ from $U(\mathbb{Z}_q^n \times \mathbb{Z}_q^l)$. □

Claim 5.7. *In* Game$_3$*, no adversary can obtain the information b if* $m \geq 5(n+l)\log q$*. Formally, we have that*

$$\left|\Pr[S_3] - \frac{1}{2}\right| \leq \mathrm{negl}(n).$$

Proof. By the parameter setting, we can apply the left-over hash lemma to the target ciphertext and this concludes the proof. □

5.2 Extension

We next consider a variant of LWEPRE, denoted by LWEPRE2. In this variant, users share A as the public parameter as users share the group $(\mathbb{G}, q, g)$ in the ElGamal encryption scheme.

Setup(n): Given input the security parameter n, it outputs a random matrix $A \in \mathbb{Z}_q^{n\times m}$ as *param*.

Reg(A, i): It generates $S_i \leftarrow \mathbb{Z}_q^{n\times l}$, and $X_i \leftarrow \bar{\Psi}_\alpha^{l\times m}$, and computes $P_i = S_i^T A + X_i \in \mathbb{Z}_q^{l\times m}$. It outputs $ek_i = P_i$ and $dk_i = S_i$.

ReKeyGen, Enc, ReEnc, Dec: They are the same as in LWEPRE.

The correctness and the multi-hop correctness of LWEPRE2 follow from these of LWEPRE. In order to show the security, we need a lemma on the Gaussian below.

Key Lemma: The following lemma states that the discretized folded Gaussian with variance $\alpha^2/2\pi$ statistically hides the discretized folded Gaussian with variance $\delta^2\alpha^2/2\pi$, when δ is negligible. The similar lemma appears in [14, 8]. Additionally, the lemmas are used to construct a key-leakage resilient secret-key encryption scheme [8] and a key-dependent-message secure public-key encryption scheme [4].

Binding two following claims, our lemma is obtained.

Lemma 5.8. *Let* $q = q(n)$ *be super-polynomial integer function of* n *and* $\alpha = \alpha(n) > 0$ *and* $\delta \in (0,1)$ *reals. If* δ *is* $n^{-\omega(1)}$*, then the statistical distance between* $\bar{\Psi}_\alpha$ *and* $\bar{\Psi}_\alpha + \bar{\Psi}_{\delta\alpha}$ *is at most* $n^{-\omega(1)}$*.*

A similar claim already appeared in [14, Claim 2.2], the statistical distance between Ψ_α and $\Psi_{(1+\delta)\alpha} = \Psi_\alpha + \Psi_{\delta\alpha}$ is at most 9δ for any $\delta \in [0,1)$, whose distributions are not discretized.

Proof. Let $\mu = \delta q\alpha t$ be a natural number. Then, from Claim 5.9, we have that $\Pr[|\bar{X}| \geq \mu]$ is at most $\frac{1}{\sqrt{2\pi}t}\exp(-\pi t^2)$. For $\mu \leq \mu'$, we have that the statistical distance between $\bar{\Psi}_\alpha$ and $\bar{\Psi}_\alpha + \mu'$ is at most $(\mu+2)/q\alpha$. Hence, the statistical distance between $\bar{\Psi}_\alpha$ and $\bar{\Psi}_\alpha + \bar{\Psi}_{\delta\alpha}$ is at most $\frac{1}{\sqrt{2\pi}t}\exp(-\pi t^2) + 2\delta t$. By setting $t = \omega(\sqrt{\log n}) \in \mathrm{poly}(n)$ and $\delta t = n^{-\omega(1)}$, we have that the upperbound is $n^{-\omega(1)}$. □

For example, we set $q(n) = n^{2\log n}$, $\alpha = 1/n^2$, $\delta = n^{-\log n}$, $t = \log n$. Then, $q\cdot\delta\alpha = n^{\Theta(\log n)}$ is super-polynomial in n and $\delta t = n^{-\Theta(\log n)}$ is negligible in n.

Claim 5.9. *Let* $\bar{X}$ *be a random variable according to* $\bar{\Psi}_{\delta\alpha}$*. Then,*

$$\Pr[|\bar{X}| \leq \mu] \geq 1 - B(q,\alpha,\delta,\mu),$$

where

$$B(q,\alpha,\delta,\mu) = \frac{\delta q\alpha}{(\mu+1/2)\sqrt{2\pi}}\cdot\exp\left(-\frac{\pi(\mu+1/2)^2}{\delta^2 q^2\alpha^2}\right).$$

In particular, if $\mu = \delta q\alpha\cdot\omega(\sqrt{\log n})$*,* $\Pr[|\bar{X}| \geq \mu]$ *is negligible in* n*.*

Proof. Let $B_\delta = \frac{\delta q\alpha}{(\mu+1/2)\sqrt{2\pi}}\exp(-\pi(\mu+1/2)^2/\delta^2 q^2\alpha^2)$. In order to prove the claim, it is sufficient to show that, for $X \sim \Psi_{\delta\alpha}$, $\Pr[|X| \geq (\mu+1/2)/q] \leq B(q,\alpha,\delta,\mu)$. Hence, we show that, for $X \sim N(0,(\delta\alpha)^2/2\pi)$, $\Pr[|X| \geq (\mu+1/2)/q] \leq B(q,\alpha,\delta,\mu)$.

Applying the tail bound for the Gaussian that $\Pr[|X| \geq t\sigma] \leq \frac{1}{t}\cdot\exp(-t^2/2)$ for $X \sim N(0,\sigma^2)$, we have that

$$\Pr[|X| \geq (\mu+1/2)/q] \leq \frac{\delta q\alpha}{(\mu+1/2)\sqrt{2\pi}}\cdot\exp\left(-\frac{\pi(\mu+1/2)^2}{\delta^2 q^2\alpha^2}\right).$$

This completes the proof. □

Claim 5.10. *For any $\alpha > 0$, any $q \in \mathbb{N}$, and any $\mu \in \mathbb{N}$, the statistical distance between $\bar{\Psi}_\alpha$ and $\bar{\Psi}_\alpha + \mu$ is at most $(\mu + 2)/q\alpha$.*

Proof. Let us consider a statistical distance Δ_μ between $dN_q(\alpha^2/2\pi)$ and $dN_q(\alpha^2/2\pi)+\mu$, where $dN_q(\sigma^2)$ is the following distribution; samples X from $N(0, \sigma^2)$ and returns $\lfloor qX \rceil$. Since $\Delta_\mu \geq \Delta(\bar{\Psi}_\alpha, \bar{\Psi}_\alpha + \mu)$, we bound this distance by $(\mu + 2)/q\alpha$.

It is obvious that $\Delta_\mu \geq \Delta_{\mu'}$ if $\mu \geq \mu'$. Hence, we assume that μ is even and show that $\Delta_\mu \leq (\mu + 1)/q\alpha$. Now, since μ is even, the probability that $\mu/2$ is the sample from $dN_q(\alpha^2/2\pi)$ equals to the probability that from $dN_q(\alpha^2/2\pi) + \mu$. Therefore, we have that

$$
\begin{aligned}
\Delta_\mu &\leq 2 \sum_{k<\mu/2} \Pr_{X\sim dN_q(\alpha^2/2\pi)}[X = k] - \Pr_{X\sim dN_q(\alpha^2/2\pi)+\mu}[X = k] \\
&= 2 \sum_{k<\mu/2} \int_{k-1/2}^{k+1/2} \frac{1}{q\alpha}\rho_{q\alpha}(x)\,dx - \int_{k-1/2}^{k+1/2} \frac{1}{q\alpha}\rho_{q\alpha}(x-\mu)\,dx \\
&= 2\left(\int_{-\infty}^{\mu/2+1/2} \frac{1}{q\alpha}\rho_{q\alpha}(x)\,dx - \int_{-\infty}^{\mu/2+1/2} \frac{1}{q\alpha}\rho_{q\alpha}(x-\mu)\,dx\right) \\
&= \Pr_{X\sim N(0,q^2\alpha^2/2\pi)}[X \leq \mu/2 + 1/2] \\
&\quad - \Pr_{X\sim N(0,q^2\alpha^2/2\pi)}[X \leq -\mu/2 + 1/2] \\
&\leq \Pr_{X\sim N(0,q^2\alpha^2/2\pi)}[|X| \leq \mu/2 + 1/2] \\
&= \int_{-(\mu+1)/2}^{(\mu+1)/2} \frac{1}{q\alpha} \exp\left(-\pi\frac{x^2}{q^2\alpha^2}\right) dx \\
&\leq \int_{-(\mu+1)/2}^{(\mu+1)/2} \frac{1}{q\alpha}\,dx = \frac{\mu+1}{q\alpha}.
\end{aligned}
$$

□

Proof of Security: We define the sequence of the games and bound the distance between the games.

Game_0: The original IND-PRE-CPA game. First, the challenger feeds $A \leftarrow \mathbb{Z}_q^{n\times m}$ to the adversary $\mathcal{A}$. The challenger simulates the oracles in the challenge phase. If the oracle CHALLENGE receives (i^*, w_0, w_1), it flips a coin $b \in \{0,1\}$ and returns the target ciphertext $(u^*, v^*) = (Ae^*, P_{i^*}e^* + t(w_b))$, where $e^* \leftarrow \{0,1\}^m$. Finally, the adversary outputs a guess b'. If $b = b'$, then the challenger outputs 1, otherwise 0.

Game_1: We modify the above game, by changing the generation methods of keys. At the beginning of the challenge phase, the challenger first generates re-encryption keys $R_{1\leftrightarrow j} \leftarrow \mathbb{Z}_q^{n\times l}$ for $j = 2, \ldots, Q$. The other re-encryption key $R_{i\leftrightarrow j}$ is computed by $R_{i\leftrightarrow j} = R_{1\leftrightarrow j} - R_{1\leftrightarrow i}$. Next it chooses $S_1 \leftarrow \mathbb{Z}_q^{n\times l}$ and $X_1 \leftarrow \chi^{l\times m}$, and computes $P_1 = S_1^T A + X_1$. If INIT is called with an input i, the challenger chooses and $X_i \leftarrow \chi^{l\times m}$, and computes $P_i = S_1^T A - R_{1\leftrightarrow i}^T A + X_i$. If REKEY is called with $i, j \in HU$, then it returns $R_{i\leftrightarrow j}$. If REENC is called with $i, j, (u, c)$, then it uses the re-encryption key $R_{i\leftrightarrow j}$ to re-encrypt the ciphertext. The other conditions are the same as in the original game, Game_0.

$\mathsf{Game}_{1.5}$: We change the generation method of the noises. We replace $X_1, \ldots, X_Q \leftarrow \bar{\Psi}_\alpha^{l\times m}$ with $X + X_1, \ldots, X + X_Q$, where $X \leftarrow \bar{\Psi}_{\delta\alpha}^{l\times m}$. Hence, the key of the user i is $P_i = S_1^T A - R_{1\leftrightarrow i}^T A + X + X_i$.

Game_2: We replace the key of the user 1. The challenger queries to the oracle $A_{S,\bar{\Psi}_{\delta\alpha}}$ and obtains m samples $(\bar{A}, \bar{P} = S^T\bar{A} + \bar{X}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{l\times m}$. It computes $P_i = \bar{P} - R_{1\leftrightarrow i}^T\bar{A} + X_i$, where $X_i \leftarrow \bar{\Psi}_\alpha^{l\times m}$ for $i = 1, \ldots, k$. The other conditions are the same as in the previous game, $\mathsf{Game}_{1.5}$.

Game_3: We replace the oracle $A_{S,\bar{\Psi}_{\delta\alpha}}$ with $U(\mathbb{Z}_q^n \times \mathbb{Z}_q^l)$. Then, the challenger obtains m samples (A, P) from $U(\mathbb{Z}_q^n \times \mathbb{Z}_q^l)$.

The main strategy of the security proof is similar to that in the previous one. We note that Game_1 and $\mathsf{Game}_{1.5}$ is statistically identical if the parameter settings satisfy the conditions in Lemma 5.8. The other games are statistically or computationally identical as in the previous proofs. We omit the details due to limit of the paper.

References

[1] ATENIESE, G., BENSON, K., AND HOHENBERGER, S. Key-private proxy re-encryption. In *CT-RSA 2009* (2009), M. Fischlin, Ed., vol. 5473 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 279–294. The full version is available at `http://eprint.iacr.org/2008/463`.

[2] ATENIESE, G., FU, K., GREEN, M., AND HOHENBERGER, S. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC) 9*, 1 (February 2006), 1–30.

[3] BLAZE, M., BLEUMER, G., AND STRAUSS, M. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT '98* (1998), K. Nyberg,

Ed., vol. 1403 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 127–144.

[4] Brakerski, Z., Goldwasser, S., and Kalai, Y. Circular-secure encryption beyond Affine functions. Cryptology ePrint Archive, Report 2009/485, 2009.

[5] Canetti, R., and Hohenberger, S. Chosen-ciphertext secure proxy re-encryption. In *CCS 2007* (2007), P. Ning, S. De Capitani di Vimercati, and P. F. Syverson, Eds., ACM, pp. 185–194.

[6] Chu, C.-K., and Tzeng, W.-G. Identity-based proxy re-encryption without random oracles. In *ISC 2007* (2007), J. A. Garay, A. K. Lenstra, M. Mambo, and R. Peralta, Eds., vol. 4779 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 189–202.

[7] Deng, R. H., Weng, J., Liu, S., and Chen, K. Chosen-ciphertext secure proxy re-encryption without pairings. In *CANS 2008* (2008), M. K. Franklin, L. C. K. Hui, and D. S. Wong, Eds., vol. 5339 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 1–17.

[8] Goldwasser, S., Kalai, Y., Peikert, C., and Vaikuntanathan, V. Robustness of the learning with errors assumption. In *ICS 2010* (Beijing, China, 2010), A. C.-C. Yao, Ed., Tsinghua University Press, pp. 230–240.

[9] Green, M., and Ateniese, G. Identity-based proxy re-encryption. In *ACNS 2007* (2007), J. Katz and M. Yung, Eds., vol. 4521 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 288–306.

[10] Libert, B., and Vergnaud, D. Unidirectional chosen-ciphertext secure proxy re-encryption. In *PKC 2008* (2008), R. Cramer, Ed., vol. 4939 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 360–379.

[11] Matsuda, T., Nishimaki, R., and Tanaka, K. CCA proxy re-encryption without bilinear maps in the standard model (extended abstract). SCIS 2010, 2010. To appear in *PKC 2010*.

[12] Matsuo, T. Proxy re-encryption systems for identity-based encryption. In *Pairing 2007* (2007), T. Takagi, T. Okamoto, E. Okamoto, and T. Okamoto, Eds., vol. 4575 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 247–267.

[13] Peikert, C., Vaikuntanathan, V., and Waters, B. A framework for efficient and composable oblivious transfer. In *CRYPTO 2008* (2008), D. Wagner, Ed., vol. 5157 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 554–571.

[14] Regev, O. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM 56*, 6 (2009), Article 34. Preliminary version in *STOC 2005*, 2005.