Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A leakage-resilient certificateless public key encryption scheme with CCA2 security*

Yan-wei ZHOU[1], Bo YANG‡[1,2], Hao CHENG[1], Qing-long WANG[2]

[1]*School of Computer Science, Shaanxi Normal University, Xi'an 710119, China*

[2]*School of Information Engineering, Chang'an University, Xi'an 710064, China*

E-mail: zyw@snnu.edu.cn; byang@snnu.edu.cn; nicke_cheng@yahoo.com.cn; qlwang@chd.edu.cn

Received Dec. 21, 2016; Revision accepted Apr. 10, 2017; Crosschecked Apr. 8, 2018

Abstract: In recent years, much attention has been focused on designing provably secure cryptographic primitives in the presence of key leakage. Many constructions of leakage-resilient cryptographic primitives have been proposed. However, for any polynomial time adversary, most existing leakage-resilient cryptographic primitives cannot ensure that their outputs are random, and any polynomial time adversary can obtain a certain amount of leakage on the secret key from the corresponding output of a cryptographic primitive. In this study, to achieve better performance, a new construction of a chosen ciphertext attack 2 (CCA2) secure, leakage-resilient, and certificateless public-key encryption scheme is proposed, whose security is proved based on the hardness of the classic decisional Diffie-Hellman assumption. According to our analysis, our method can tolerate leakage attacks on the private key. This method also achieves better performance because polynomial time adversaries cannot achieve leakage on the private key from the corresponding ciphertext, and a key leakage ratio of 1/2 can be achieved. Because of these good features, our method may be significant in practical applications.

Key words: Certificateless public-key encryption; Leakage-resilience; Provable security; CCA2 security; Decisional Diffie-Hellman

https://doi.org/10.1631/FITEE.1601849 **CLC number:** TP309

1 Introduction

The identity-based encryption (IBE) scheme (Shamir, 1984) requires a trusted third-party private key generator (PKG) center to generate private keys for all identities. The complete private keys of all users are held by PKG. In other words, PKG can replace any user to encrypt a message or to decrypt a ciphertext. Therefore, in an IBE scheme, PKG can act with impunity. To alleviate this problem, Al-Riyami and Paterson (2003) proposed the certificateless public-key encryption (CL-PKE) scheme. In the CL-PKE scheme, PKG can generate only a partial private key that corresponds to a user's identity, and the master key is held secretly by PKG. Together with the secret value generated by the user, a complete private key can be constructed. PKG creates the private key of any user working together with other users. Therefore, in a CL-PKE scheme, the user requires both the receiver's identity and his/her public key to encrypt a message. To decrypt a ciphertext, a receiver requires a complete private key.

‡ Corresponding author

* Project supported by the National Key R&D Program of China (No. 2017YFB0802000), the National Natural Science Foundation of China (Nos. 61572303 and 61772326), the National Cryptography Development Fund During the 13th Five-Year Plan Period, China (No. MMJJ20170216), the Foundation of State Key Laboratory of Information Security, China (No. 2017-MS-03), and the Fundamental Research Funds for the Central Universities, China (No. GK201803064)

ORCID: Yan-wei ZHOU, http://orcid.org/0000-0002-7254-3579

1.1 Leakage resilience

Traditionally, cryptographic primitive security has been researched in an ideal setting, in which no polynomial time adversary could obtain valuable information about the internal secret stages (e.g., private key); i.e., in an ideal security model, the participant has a secret state which is assumed to be completely inaccessible to the adversary. However, in real life, some valuable information about the internal secret states can be captured by any polynomial time adversary through various leakage attacks (e.g., side-channel and cold-boot attacks). Therefore, the traditional cryptographic primitives cannot maintain their claimed security in real life if the adversary obtains leakage of a certain amount of internal secret state information, since their claimed security is proved in an ideal security model and the internal secret stage leakages are omitted.

Recently, many constructions of leakage-resilient cryptographic primitives have been proposed, such as leakage-resilient public-key encryption (LR-PKE) (Naor and Segev, 2012; Li et al., 2013; Liu et al., 2013), leakage-resilient identity-based encryption (Li et al., 2016), leakage-resilient certificateless signcryption (Zhou et al., 2016), leakage-resilient authenticated key exchange (Chen et al., 2016a,b), and leakage-resilient certificate-based encryption (Yu et al., 2016). The above constructions can maintain their claimed security even if the adversary obtains a certain amount of internal secret state leakage.

1.2 Bounded-leakage model

In the bounded-leakage model, arbitrary leakage on the private key can be captured by any polynomial time adversary, as long as the total bit leakage is bounded by a fixed leakage parameter. In the entire lifetime of the key, the leakage must be less than leakage parameter λ, which is a fixed parameter and subject to constraint $\lambda \leq |\mathrm{SK_{id}}|$.

In the leakage setting, the adversary's leakage attacks are modeled by obtaining access to a leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ (for details, see Section 2.7). In other words, in the proof of security, any polynomial time adversary can get leakage $f_i(\mathrm{SK_{id}})$ on private key $\mathrm{SK_{id}}$ from leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ with an efficient computable leakage function $f_i : \{0,1\}^* \to \{0,1\}^{\lambda_i}$ and an identity id as input.

1.3 Prior constructions and limitations

The first security model for the LR-PKE scheme was presented by Akavia et al. (2009). Later, based on existing conclusions (Cramer and Shoup, 2003; Akavia et al., 2009), the leakage-resilient chosen-plaintext attack (LR-CPA) and leakage-resilient chosen-ciphertext attack (LR-CCA) models were distinguished by Naor and Segev (2012). In addition, based on the Cramer-Shoup scheme (Cramer and Shoup, 2003), two LR-PKE schemes were constructed by Naor and Segev (2012).

Next, Liu et al. (2013), Li et al. (2013), and Qin et al. (2015) proposed three new methods to construct a CCA secure LR-PKE scheme. However, in these schemes, any polynomial time adversary can be leaked on the private key from the corresponding ciphertext, because some elements in ciphertext can be written as a function on the private key. In other words, for any polynomial time adversary, these schemes cannot ensure that all elements of ciphertext are random.

Furthermore, Xiong et al. (2013) proposed a leakage-resilient certificateless public-key encryption (LR-CL-PKE) scheme, whose security is proved based on the intractability assumptions of composite order bilinear groups. However, this scheme is only CCA1 secure, and cannot achieve CCA2 security.

Remark For presentation simplicity, we will call the CCA secure public-key encryption (PKE) schemes proposed by Naor and Segev (2012), Liu et al. (2013), Li et al. (2013), Qin et al. (2015), and Xiong et al. (2013) 'LR-PKE-NS', 'LR-PKE-Liu', 'LR-PKE-Li', 'LR-PKE-Qin', and 'LR-CL-PKE-Xiong', respectively.

1.4 Our contributions

In this study, we will focus on the construction of an efficient CCA2 secure CL-PKE scheme that can tolerate leakage attacks. Our proposal can achieve CCA2 security based on the hardness of the classical decisional Diffie-Hellman (DDH) assumption and provides better performance by preventing any polynomial time adversary from achieving leakage on the private key from the corresponding ciphertext, resisting against bounded leakage attacks, etc.

Overall, our proposal is a new mathod for constructing a more efficient LR-CL-PKE scheme without sacrificing CCA2 security.

2 Preliminaries

2.1 Notations

Let $k \in \mathbb{N}$ denote the security parameter, $[n]$ the set $\{1, 2, \cdots, n\}$, $s \leftarrow_R S$ the operation of picking an element s uniformly at random from S, and $y \leftarrow \mathcal{A}(x)$ the operation of running algorithm $\mathcal{A}$ with x as input and assigning y as the result. If an algorithm $\mathcal{A}$ is randomized and the computation of $\mathcal{A}$ for any input terminates in at most polynomial steps, then $\mathcal{A}$ is a probabilistic polynomial-time (PPT) algorithm.

2.2 Target collision resistant

Definition 1 (Cramer and Shoup, 2003) Let $\mathcal{H}_{\mathcal{I}} = \{H_i : \mathcal{X} \to \mathcal{Y}\}_{i \in \mathcal{I}}$ be a set of one-way hash functions. For any PPT adversary $\mathcal{A}$, if probability $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{TCR}}(k) = \Pr[H_i(x) = H_i(x') \wedge x \neq x']$ is negligible, where $x' \leftarrow \mathcal{A}(x, H_i)$, $i \leftarrow_R \mathcal{I}$, and $x \leftarrow_R \mathcal{X}$, then $\mathcal{H}_{\mathcal{I}}$ is target collision resistant.

2.3 Computational assumptions

Let $\mathbb{G} = (q, G, g)$ be generated by a PPT algorithm $\mathcal{G}(1^k)$ with security parameter k, where G is a group of order big prime q, and g is a generator of G.

Definition 2 (DDH) (Cramer and Shoup, 2003) Given two four-tuple sets (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c) for some unknown exponents $a, b, c \leftarrow_R Z_q^*$ for any PPT adversary $\mathcal{A}$, advantage $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DDH}}(k) = |\Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1] - \Pr[\mathcal{A}(g, g^a, g^b, g^c) = 1]|$ is negligible.

Definition 3 (Discrete logarithm, DL) (Cramer and Shoup, 2003) Given a two-tuple set (g, g^d) for an unknown exponent $d \leftarrow_R Z_q^*$ for any PPT adversary $\mathcal{A}$, advantage $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DDH}}(k) = \Pr[\mathcal{A}(g, g^d) = d]$ is negligible.

2.4 Random extractor

The statistical distance of A and B is defined as $\mathrm{SD}(A, B) = \frac{1}{2}\sum_{\omega \in \Omega} |\Pr[A = \omega] - \Pr[B = \omega]|$, where A and B are two random variables over a finite domain Ω. For any random variables A and C, let $H_\infty(A) = -\log(\max_a \Pr[A = a])$ denote the minimum entropy of A, and $\widetilde{H}_\infty(A|C) = \log(E_c \max_a \Pr[A = a|C = c]) = -\log(E_{c \leftarrow C}[2^{-H_\infty(A|C=c)}])$ denote the average minimum entropy of A conditioned on another variable C. In other words, for any PPT adversary $\mathcal{A}$, we can obtain

$$\begin{aligned}\Pr(\mathcal{A}(C) = A) &= E_c[\Pr(\mathcal{A}(C) = A)] \\ &\leq E_c[2^{-H_\infty(A|C=c)}] \\ &= 2^{-\widetilde{H}_\infty(A|C)}.\end{aligned}$$

Definition 4 (Random extractor) (Dodis et al., 2008) Efficient computable function $\mathrm{Ext} : \{0,1\}^{l_n} \cdot \{0,1\}^{l_t} \to \{0,1\}^{l_m}$ is an average-case (x, y)-strong extractor. For any random variables A and C (such that $A \in \{0,1\}^{l_n}$ and $\widetilde{H}_\infty(A|C) \geq x$), we have $\mathrm{SD}((\mathrm{Ext}(A, S), S, C), (U_m, S, C)) \leq y$, where $S \leftarrow_R \{0,1\}^{l_t}$ and $U_m \leftarrow_R \{0,1\}^{l_m}$.

Lemma 1 (Dodis et al., 2008) For any random variables A, B, and C, if B has at most 2^l possible values, then we can obtain

$$\widetilde{H}_\infty(A|(B, C)) \geq \widetilde{H}_\infty(A|C) - l.$$

2.5 Leftover hash lemma

If for all distinct $x_1 \neq x_2 \leftarrow_R \mathcal{X}$, we have $\Pr[H_i(x_1) = H_i(x_2)] \leq 1/|\mathcal{Y}|$, then $H_i : \mathcal{X} \to \mathcal{Y}$ is a universal hash function.

Example 1 (Liu et al., 2013) Let G be a cyclic group of prime order q, $H_{k_1, k_2, \cdots, k_L}(g_0, g_1, \cdots, g_L) = g_0 g_1^{k_1} \cdots g_L^{k_L}$ be a universal hash function, where $g_0 \in G$, and $\forall i \in [L], k_i \in Z_q^*$ and $g_i \in G$.

Lemma 2 (Dodis et al., 2008) If $\mathcal{H}_{\mathcal{I}} : \{H_i : \mathcal{X} \to \mathcal{Y}\}$ is a set of universal hash functions, then for any two random variables $A \leftarrow_R \mathcal{X}$ and C, we have

$$\mathrm{SD}((H_i(A), i), (U_y, i)) \leq \frac{1}{2}\sqrt{2^{-H_\infty(A)}|\mathcal{Y}|},$$

$$\mathrm{SD}((H_i(A), i, C), (U_y, i, C)) \leq \frac{1}{2}\sqrt{2^{-\widetilde{H}_\infty(A|C)}|\mathcal{Y}|},$$

where $i \leftarrow_R \mathcal{I}$ and $U_y \leftarrow_R \mathcal{Y}$.

Lemma 3 (Dodis et al., 2008) Let $\mathcal{H}_{\mathcal{I}} : \{H_i : \{0,1\}^{l_n} \to \{0,1\}^{l_m}\}$ be a set of universal hash functions. For any random variables A and C (such that $A \leftarrow_R \{0,1\}^{l_n}$ and $\widetilde{H}_\infty(A|C) \geq x$), we have $\mathrm{SD}((C, i, H_i(A)), (C, i, U_m)) \leq y$ as long as $l_m \leq x - 2\log(1/y)$, where $i \leftarrow_R \mathcal{I}$ and $U_m \leftarrow_R \{0,1\}^{l_m}$.

2.6 Key derivation functions

Definition 5 (Cramer and Shoup, 2003) Let $\mathbb{G} = (q, G, g)$ be generated by a PPT algorithm $\mathcal{G}(1^k)$. If for any PPT algorithm $\mathcal{A}$, advantage $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{KDF}}(k) = |\Pr[\mathcal{A}(R, \mathrm{KDF}(R)) = 1] - \Pr[\mathcal{A}(R, k_1^*, k_2^*) = 1]|$ is negligible, where $R \leftarrow_R G$ and $k_1^*, k_2^* \leftarrow_R Z_q^*$, then $\mathrm{KDF} : G \to Z_q^* \cdot Z_q^*$ is a secure key derivation function.

Note that if variable R has enough entropy, then output $\mathrm{KDF}(R)$ of key derivation function $\mathrm{KDF} : G \to Z_q^* \cdot Z_q^*$ will be random; i.e., for any PPT adversary, $\mathrm{KDF}(R)$ is a uniform distribution over $Z_q^* \cdot Z_q^*$.

2.7 Leakage oracle

The adversary's leakage attack on private key $\mathrm{SK_{id}}$ is modeled by giving adversary access to a leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ on private key $\mathrm{SK_{id}}$, and the adversary can query to gain the leakage about $\mathrm{SK_{id}}$. The formal definition of leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ is as follows:

Definition 6 (Alwen et al., 2009) A leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ is parameterized by a private key $\mathrm{SK_{id}}$, a leakage parameter λ, and a security parameter k. A query to the oracle consists of an efficient computable leakage function $f_i : \{0,1\}^* \to \{0,1\}^{\lambda_i}$, where f_i can be chosen adaptively. Leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ checks if the sum of λ_i, over all queries received so far, exceeds leakage parameter λ and ignores the query if this is the case. Otherwise, leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ responds with $f_i(\mathrm{SK_{id}})$.

Without loss of generality, we can assume that the adversary can query leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ only once with an efficient computable leakage function $f : \{0,1\}^* \to \{0,1\}^{\lambda}$ whose output $f(\mathrm{id})$ is at most λ bits.

3 Security model of the CL-PKE scheme

A CL-PKE scheme is composed of seven PPT algorithms: Setup, Set-Secret-Value, Partial-Key-Extract, Set-Private-Key, Set-Public-Key, Enc, and Dec. Due to lack of space, the definitions of these algorithms are omitted; for details, see Al-Riyami and Paterson (2003).

In the security proof of the CL-PKE scheme, we consider the following two adversary types:

Type I: Adversary $\mathcal{A}_1$ does not have access to the master secret key; however, it may replace the public key of any user. In addition, we make several restrictions on this type of adversary: (1) $\mathcal{A}_1$ cannot make a partial private key extraction query for challenge identity; (2) $\mathcal{A}_1$ cannot replace a public key for challenge identity before the challenge stage.

Type II: Adversary $\mathcal{A}_2$ does have access to the master secret key; however, it cannot replace the public key of any user. The restrictions on this type adversary are: (1) $\mathcal{A}_2$ cannot extract the private key for challenge identity; (2) $\mathcal{A}_2$ cannot replace the public key of any user.

In the leakage setting, we require that a CL-PKE scheme remain its original security even in the presence of key leakage. Now, we review a security notion of the LR-CL-PKE scheme, called 'indistinguishability of ciphertext under leakage-resilient chosen ciphertext attack 2' (LR-CCA2). This notion is defined by the following interaction games: Game_i $(i = 1, 2)$ between an adversary $\mathcal{A}_i$ $(i = 1, 2)$ and a challenger $\mathcal{C}$ under security parameter k and leakage parameter λ. From now on, let $\mathcal{M}$ denote the plaintext space, and ID the identity space of the user.

Game_1: This game is performed by a Type I adversary $\mathcal{A}_1$ and a challenger $\mathcal{C}$.

Setup: In this stage, public parameter Params and master secret key S_{msk} are generated by running setup algorithm $\mathrm{Setup}(1^k)$. Challenger $\mathcal{C}$ sends Params to adversary $\mathcal{A}_1$ while keeping S_{msk} secret.

Test stage 1: $\mathcal{A}_1$ performs a polynomially bounded number of queries, and each query may depend on the answers to the previous queries.

1. On receiving a query (id, partial key extraction): $\mathcal{C}$ runs $(s_{\mathrm{id}}, S_{\mathrm{id}}) = \text{Set-Secret-Value(id)}$ and $(P_{\mathrm{id}}, d_{\mathrm{id}}) = \text{Partial-Key-Extract}(S_{\mathrm{msk}}, \mathrm{id}, S_{\mathrm{id}})$, and sends $(P_{\mathrm{id}}, d_{\mathrm{id}})$ to $\mathcal{A}_1$.

2. On receiving a query (id, private key extraction): $\mathcal{C}$ runs $(s_{\mathrm{id}}, S_{\mathrm{id}}) = \text{Set-Secret-Value(id)}$ and $(P_{\mathrm{id}}, d_{\mathrm{id}}) = \text{Partial-Key-Extract}(S_{\mathrm{msk}}, \mathrm{id}, S_{\mathrm{id}})$, and sends $\mathrm{SK_{id}} = \text{Set-Private-Key}(d_{\mathrm{id}}, s_{\mathrm{id}})$ to $\mathcal{A}_1$.

3. On receiving a query (id, public key extraction): $\mathcal{C}$ runs $(s_{\mathrm{id}}, S_{\mathrm{id}}) = \text{Set-Secret-Value(id)}$ and $(P_{\mathrm{id}}, d_{\mathrm{id}}) = \text{Partial-Key-Extract}(S_{\mathrm{msk}}, \mathrm{id}, S_{\mathrm{id}})$, and sends $\mathrm{PK_{id}} = \text{Set-Public-Key}(P_{\mathrm{id}}, S_{\mathrm{id}})$ to $\mathcal{A}_1$.

4. At any time, $\mathcal{A}_1$ can replace public key $\mathrm{PK_{id}}$ of any user with its own.

5. On receiving a query (id, c, decryption): Private key $\mathrm{SK_{id}}$ is obtained through a private key extraction query for id, and $\mathcal{C}$ sends the corresponding result $M/\perp = \mathrm{Dec}(\mathrm{SK_{id}}, c)$ to $\mathcal{A}_1$.

6. On receiving a query (id, $f_i(\cdot)$, leakage): $\mathcal{C}$ runs leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ and returns the corresponding answer $f_i(\mathrm{SK_{id}})$, where $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ takes private key $\mathrm{SK_{id}}$ and efficient computable leakage function $f_i : \{0,1\}^* \to \{0,1\}^{\lambda_i}$ as inputs. In the whole process of leakage queries, the total length of $f_i(\mathrm{SK_{id}})$, all returned from leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$

about the same private key $\mathrm{SK_{id}}$, must be less than leakage parameter λ, i.e., $\sum_{j=1}^{i} f_j(\mathrm{SK_{id}}) \leq \lambda$. Otherwise, an invalid answer $\perp$ will be obtained. In other words, adversary $\mathcal{A}_1$ can obtain a certain amount of leakage on private key $\mathrm{SK_{id}}$ by asking leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$.

Challenge stage: In this stage, $\mathcal{A}_1$ outputs two equal-length messages $M_0, M_1 \in \mathcal{M}$ and a challenge identity $\mathrm{id}^* \in \mathrm{ID}$. On receiving M_0, M_1, and id^*, $\mathcal{C}$ chooses a random bit $b \leftarrow_R \{0,1\}$, and sends challenge ciphertext $c_b \leftarrow \mathrm{Enc}(\mathrm{PK_{id^*}}, M_b)$ to $\mathcal{A}_1$. Note that id^* has not been queried to extract a private key or replace a public key for challenge identity at any time, and cannot be equal to an identity for which the public key has been replaced.

Test stage 2: This stage is similar to test stage 1. However, id^* has not been queried to extract a private key or replace a public key at any time, and cannot be equal to an identity for which the public key has been replaced. Except for the above constraints, in this stage, no decryption queries should be made on challenge ciphertext c_b and challenge identity id^*. Also, the leakage query is banned; i.e., in this stage, $\mathcal{A}_1$ cannot be leaked on the private key of any user from leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$.

Output: $\mathcal{A}_1$ outputs a bit $b' \in \{0,1\}$ as the guess of b. If $b' = b$, then $\mathcal{A}_1$ wins.

In Game_1, the advantage of $\mathcal{A}_1$ is defined as $\mathrm{Adv}^{\text{LR-CCA2}}_{\text{CL-PKE},\mathcal{A}_1}(k,\lambda) = |\Pr[\mathcal{A}_1 \text{ wins}] - 1/2|$.

Game_2: This game is performed by a Type II adversary $\mathcal{A}_2$ and a challenger $\mathcal{C}$.

Setup: In this stage, public parameter Params and master secret key S_{msk} are generated by running setup algorithm $\mathrm{Setup}(1^k)$. Challenger $\mathcal{C}$ sends Params and S_{msk} to adversary $\mathcal{A}_2$. Note that in this game, S_{msk} is controlled by $\mathcal{A}_2$.

Test stage 1: This stage with Game_1 is the same; however, in Game_2, $\mathcal{A}_2$ can compute any user's partial private key for itself by using the master secret key, and cannot replace the public key of any user.

Challenge stage: In this stage, two equal-length messages $M_0, M_1 \in \mathcal{M}$ and a challenge identity $\mathrm{id}^* \in \mathrm{ID}$ are submitted by $\mathcal{A}_2$, where id^* has not been issued as a private key extraction query. $\mathcal{C}$ sends challenge ciphertext $c_b \leftarrow \mathrm{Enc}(\mathrm{PK_{id^*}}, M_b)$ to $\mathcal{A}_2$, where $b \leftarrow_R \{0,1\}$.

Test stage 2: In this stage, challenge identity id^* has not been issued as a private key extraction query. Also, no decryption queries should be made on challenge ciphertext c_b and challenge identity id^*, and the leakage query is banned.

Output: $\mathcal{A}_2$ outputs a bit $b' \in \{0,1\}$ as the guess of b. If $b' = b$, then $\mathcal{A}_2$ wins.

In Game_2, the advantage of $\mathcal{A}_2$ is defined as $\mathrm{Adv}^{\text{LR-CCA2}}_{\text{CL-PKE},\mathcal{A}_2}(k,\lambda) = |\Pr[\mathcal{A}_2 \text{ wins}] - 1/2|$.

Definition 7 (LR-CCA2 security) A CL-PKE scheme is secure against adaptive leakage-resilient chosen ciphertext attacks if for any PPT adversary $\mathcal{A}_i$ $(i = 1,2)$, advantage $\mathrm{Adv}^{\text{LR-CCA2}}_{\text{CL-PKE},\mathcal{A}_i}(k,\lambda)$ $(i = 1,2)$ in above interactive game Game_i $(i = 1,2)$ is negligible. The only restriction is that the total number of leakage bits on the private key is bounded by some positive integer λ.

Consistent with the studies conducted by Naor and Segev (2012), Liu et al. (2013), Li et al. (2013), and Qin et al. (2015), in interaction games Game_1 and Game_2, adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ are given access to only leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$ prior to receiving the challenge. This is a necessary restriction because, otherwise, one could easily win the distinguishing game.

4 CCA2 secure LR-CL-PKE scheme

In this section, we show an LR-CL-PKE scheme in the bounded-leakage model based on the hardness of the classical DDH assumption and the target collision resistance of the one-way hash function.

4.1 Constructions

Our LR-CL-PKE scheme Π consists of the following algorithms, each of which is described in detail.

4.1.1 Setup

In the setup stage, PKG performs the following operations:

Let $\mathbb{G} = (q, G, g)$ be generated by a PPT algorithm $\mathcal{G}(1^k)$ with security parameter k, where G is a group of order prime q, and g is a generator of G.

Let $H : G{\cdot}G{\cdot}\{0,1\}^{l_{\mathrm{m}}}{\cdot}\{0,1\}^{l_{\mathrm{t}}} \to Z_q^*$, $H_1 : \mathrm{ID}{\cdot}G{\cdot}G \to Z_q^*$, and $H_2 : \{0,1\}^{l_{\mathrm{m}}} \to Z_q^*$ be three target collision resistant one-way hash functions, where l_{m} denotes the length of the plaintext. Thus, the message space is $\mathcal{M} = \{0,1\}^{l_{\mathrm{m}}}$.

Let $\mathrm{Ext} : G{\cdot}\{0,1\}^{l_{\mathrm{t}}} \to \{0,1\}^{l_{\mathrm{m}}}$ be an average-case $(\log q - \lambda, \varepsilon)$-strong extractor, where λ is a fixed

leakage parameter, and ε be negligible in security parameter k.

Let KDF : $G \to Z_q^* \cdot Z_q^*$ be a secure key derivative function.

Choose a random value $S_{\mathrm{msk}} \leftarrow_R Z_q^*$ as the master secret key, and compute $P_{\mathrm{Pub}} = g^{S_{\mathrm{msk}}}$. In addition, all of the algorithms described in Sections 4.1.2–4.1.4 are given Params $= \langle q, G, g, P_{\mathrm{Pub}}, H, H_1, H_2, \mathrm{Ext}, \mathrm{KDF}\rangle$ as part of their inputs.

4.1.2 Key generation

The key generation is performed by the following algorithms:

SetSecretValue(id): Choose a secret value $s_{\mathrm{id}} \leftarrow_R Z_q^*$, and compute $S_{\mathrm{id}} = g^{s_{\mathrm{id}}}$. Output s_{id} and S_{id}.

PartialKeyExtract($S_{\mathrm{msk}}, \mathrm{id}, S_{\mathrm{id}}$): Choose a random value $r \leftarrow_R Z_q^*$, and compute $P_{\mathrm{id}} = g^r$ and $d_{\mathrm{id}} = r + S_{\mathrm{msk}} H_1(\mathrm{id}, P_{\mathrm{id}}, S_{\mathrm{id}}) \bmod q$. Output d_{id} and P_{id}.

SetPublicKey($\mathrm{id}, P_{\mathrm{id}}, S_{\mathrm{id}}$): Set $\mathrm{PK}_{\mathrm{id}} = (S_{\mathrm{id}}, P_{\mathrm{id}})$ as the complete public key of user id.

SetPrivateKey($\mathrm{id}, d_{\mathrm{id}}, s_{\mathrm{id}}$): Set $\mathrm{SK}_{\mathrm{id}} = (s_{\mathrm{id}}, d_{\mathrm{id}})$ as the complete private key of user id.

Note that we can use the interaction protocol (Fig. 1) between user id and trusted third-party PKG to describe key generation.

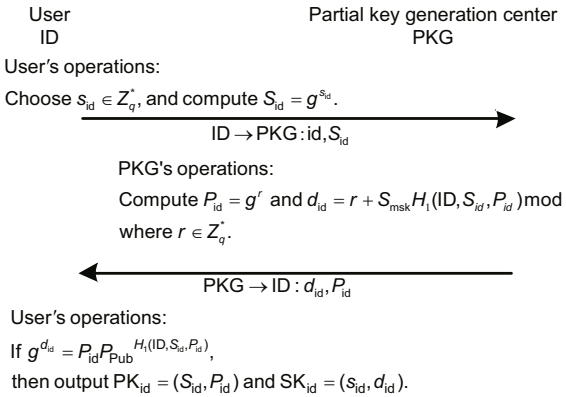

Fig. 1 Key generation algorithm

4.1.3 Encryption

To encrypt a plaintext $M \in \mathcal{M}$, encryption algorithm $c \leftarrow \mathrm{Enc}(\mathrm{PK}_{\mathrm{id}}, M)$ takes public key $\mathrm{PK}_{\mathrm{id}} = (S_{\mathrm{id}}, P_{\mathrm{id}})$ as input, and outputs a ciphertext $c = (U_1, U_2, e, v, S)$.

1. Compute $U_1 = g^{r_1}$ and $U_2 = g^{r_2}$, where $r_1, r_2 \leftarrow_R Z_q^*$.

2. Compute $e = \mathrm{Ext}({S_{\mathrm{id}}}^{r_2}({P_{\mathrm{id}}{P_{\mathrm{Pub}}}^{h_{\mathrm{id}}}})^{r_1}, S) \oplus M$, where $S \leftarrow_R \{0,1\}^{l_t}$ and $h_{\mathrm{id}} = H_1(\mathrm{id}, S_{\mathrm{id}}, P_{\mathrm{id}})$.

3. Compute $V = {S_{\mathrm{id}}}^{r_1}({P_{\mathrm{id}}{P_{\mathrm{Pub}}}^{h_{\mathrm{id}}}})^{r_2\mu}$, where $\mu = H(U_1, U_2, e, S)$.

4. Compute $v = r_1 k_1 H_2(e) + r_2 k_2 \bmod q$, where $(k_1, k_2) = \mathrm{KDF}(V)$.

5. Set $c = (U_1, U_2, e, v, S)$ as ciphertext of M, and output ciphertext c.

Note that the randomness extraction in step 2 is performed by an average-case strong extractor $\mathrm{Ext} : G \cdot \{0,1\}^{l_t} \to \{0,1\}^{l_m}$; however, step 3 is implemented through a special universal hash function $\mathcal{H}_\mu(d, f) = df^\mu$ as an average-case strong extractor, where $\mu \in Z_q^*$, $d = {S_{\mathrm{id}}}^{r_1}$, and $f = ({P_{\mathrm{id}}{P_{\mathrm{Pub}}}^{h_{\mathrm{id}}}})^{r_2}$ in our proposal, which is defined in Example 1 in Section 2 with $L = 1$. Thus, variable V has enough entropy, and from the adversary's view, the value of v is random. Therefore, in our proposal, all of the elements of the ciphertext will be random, and no PPT adversary can be leaked on the private key from the corresponding ciphertext.

4.1.4 Decryption

A ciphertext $c = (U_1, U_2, e, v, S)$ can be decrypted by decryption algorithm $M = \mathrm{Dec}(\mathrm{SK}_{\mathrm{id}}, c)$ with private key $\mathrm{SK}_{\mathrm{id}} = (s_{\mathrm{id}}, d_{\mathrm{id}})$.

Compute $V' = {U_1}^{s_{\mathrm{id}}}{U_2}^{\mu d_{\mathrm{id}}}$, where $\mu = H(U_1, U_2, e, S)$.

The consistency of ciphertext c is checked by $g^v \stackrel{?}{=} {U_1}^{k_1' H_2(e)}{U_2}^{k_2'}$, where $(k_1', k_2') = \mathrm{KDF}(V')$. If this equation does not hold, output dummy value $\perp$; otherwise, output $M = \mathrm{Ext}({U_2}^{s_{\mathrm{id}}}{U_1}^{d_{\mathrm{id}}}, S) \oplus e$ as the plaintext of c.

4.2 Correctness

The correctness of our LR-CL-PKE scheme is obtained from the following equations:

$$\begin{aligned} V' = {U_1}^{s_{\mathrm{id}}}{U_2}^{\mu d_{\mathrm{id}}} &= g^{r_1 s_{\mathrm{id}}} g^{r_2 \mu d_{\mathrm{id}}} \\ &= {S_{\mathrm{id}}}^{r_1}({P_{\mathrm{id}}{P_{\mathrm{Pub}}}^{h_{\mathrm{id}}}})^{r_2\mu} = V, \\ {U_2}^{s_{\mathrm{id}}}{U_1}^{d_{\mathrm{id}}} &= g^{r_2 s_{\mathrm{id}}} g^{r_1 d_{\mathrm{id}}} \\ &= {S_{\mathrm{id}}}^{r_2}({P_{\mathrm{id}}{P_{\mathrm{Pub}}}^{h_{\mathrm{id}}}})^{r_1}, \\ g^v = g^{r_1 k_1 H_2(e) + r_2 k_2} &= {U_1}^{k_1' H_2(e)}{U_2}^{k_2'}, \end{aligned}$$

where $\mu = H(U_1, U_2, e, S)$, $h_{\mathrm{id}} = H_1(\mathrm{id}, S_{\mathrm{id}}, P_{\mathrm{id}})$, and $g^{d_{\mathrm{id}}} = P_{\mathrm{id}}{P_{\mathrm{Pub}}}^{h_{\mathrm{id}}}$.

5 Proof of security

For any unknown exponents $a, b, c \in Z_q^*$, we call tuple (g, g^a, g^b, g^c) a 'DH instance' if $c = ab$, and call tuple (g, g^a, g^b, g^c) a 'non-DH instance' if $c \neq ab$. Note that in this study, we consider only the private key's leakage, and the leakage of the master secret key is omitted; i.e., in the proof of security, any PPT adversary can be leaked only on the private key from leakage oracle $\mathcal{O}^{\lambda,k}_{\mathrm{SK_{id}}}(\cdot)$. In addition, in Game_1 and Game_2, the leakage queries are performed by adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ before the challenge stage.

Theorem 1 If the DDH assumption is intractable and one-way hash function H is target collision resistant, then for any leakage parameter $\lambda \leq \log q - l_{\mathrm{m}} - \omega(\log k)$, our LR-CL-PKE scheme Π is LR-CCA2 secure.

To prove the above theorem, we will prove Lemmas 4 and 5. Lemma 4 shows that our LR-CL-PKE scheme Π is secure resistant against the Type I attacker, whose behavior is described in Game_1. Lemma 5 shows that our LR-CL-PKE scheme Π is secure resistant against the Type II attacker, whose behavior is described in Game_2.

Lemma 4 If the DDH assumption is intractable and the hash function H is target collision resistant, then for any leakage parameter $\lambda \leq \log q - l_{\mathrm{m}} - \omega(\log k)$, our LR-CL-PKE scheme Π is CCA2-secure resistant against the Type I adversary.

Proof If a Type I adversary $\mathcal{A}_1$ can break the security of our proposal Π with non-ignorable advantage ε_1, then there exists an adversary $\mathcal{B}$ that can break the hardness of the classical DDH assumption with obvious advantage $\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1}(\lambda, k) \in \big[(1-\frac{q_{\mathrm{S}}}{2^k})\frac{\varepsilon_1}{\mathrm{e}(q_{\mathrm{D}}+q_{\mathrm{S}}+1)}, (1-\frac{q_{\mathrm{S}}}{2^k})(1-\frac{1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1})^{q_{\mathrm{D}}+q_{\mathrm{S}}}\frac{\varepsilon_1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1} + \frac{2^{\lambda}q_{\mathrm{D}}}{q-q_{\mathrm{D}}+1} + 2^{\frac{\lambda}{2}-1}\big]$, where k is the security parameter, q is the prime order of the underlying group, q_{S} is the number of private key extraction queries, q_{D} is the number of decryption queries, and e is the base of the natural logarithm.

Now, we show how to use Type I adversary $\mathcal{A}_1$ to construct an adversary $\mathcal{B}$ that can break the hardness of the DDH assumption. First, adversary $\mathcal{B}$ receives a challenge instance (g, g^a, g^b, g^c) from the challenger of the DDH assumption, where $a, b, c \in Z_q^*$. According to the hardness of the DL problem, the exponents of the challenge instance are unknown for adversary $\mathcal{B}$. Thereafter, $\mathcal{B}$ chooses an identity $\mathrm{id}_j \in \mathrm{ID}$ as the challenge identity. In addition, five lists L_1, L_{p}, L_{P}, L_{S}, and L_{D} are maintained by $\mathcal{B}$ to keep track of the answers to the corresponding queries. These lists are initially empty; i.e., L_1 will be used to keep track of the answers of random oracle H_1, L_{p} will be used to keep track of the answers of partial key extraction queries, L_{P} will be used to keep track of the answers to the partial key extraction queries, L_{S} will be used to keep track of the answers to the public key extraction queries, and L_{D} will be used to keep track of the answers to the decryption queries.

$\mathcal{B}$ can simulate the challenger's execution of each stage of Game_1 for $\mathcal{A}_1$ as follows.

Setup: Public parameter Params and master secret key S_{msk} are generated by setup algorithm $\mathrm{Setup}(1^k)$. Adversary $\mathcal{B}$ sends Params to adversary $\mathcal{A}_1$. Also, random oracle H_1 is controlled by $\mathcal{B}$ as follows.

On receiving a query (id, S_{id}, P_{id}, random oracle H_1):

Search a tuple $\langle \mathrm{id}, S_{\mathrm{id}}, P_{\mathrm{id}}, h_1 \rangle$ from L_1 with index id, and return h_1 as the answer.

Note that in this case, we can surely find $\langle \mathrm{id}, S_{\mathrm{id}}, P_{\mathrm{id}}, h_1 \rangle$ from L_1, because tuple $\langle \mathrm{id}, S_{\mathrm{id}}, P_{\mathrm{id}}, h_1 \rangle$ will be added to L_1 in the public key extraction query for identity id.

Test stage 1: $\mathcal{B}$ answers $\mathcal{A}_1$'s queries as follows:

Partial-Key-Extract: On receiving a partial key extraction query (id, S_{id}, partial key extract):

If $\langle \mathrm{id}, S_{\mathrm{id}}, P_{\mathrm{id}}, d_{\mathrm{id}} \rangle$ can be found from L_{p} with index id, then return P_{id} and d_{id} as the answer; oltherwise, choose $d_{\mathrm{id}}, h_1 \leftarrow_R Z_q^*$, and compute $P_{\mathrm{id}} = g^{d_{\mathrm{id}}}({P_{\mathrm{Pub}}}^{h_1})^{-1}$. Also, add tuple $\langle \mathrm{id}, S_{\mathrm{id}}, P_{\mathrm{id}}, d_{\mathrm{id}} \rangle$ to L_{p} and $\langle \mathrm{id}, S_{\mathrm{id}}, P_{\mathrm{id}}, h_1 \rangle$ to L_1. Finally, return P_{id} and d_{id} as the answer.

Public-Key-Extract: On receiving a public key extraction query (id, public key extract):

If $\langle \mathrm{id}, \mathrm{PK_{id}} \rangle$ can be found from L_{P} with index id, then return $\mathrm{PK_{id}}$ as the answer; otherwise, do the following:

1. If $\mathrm{id} = \mathrm{id}_j$, set $S_{\mathrm{id}} = g^a$ (implicitly setting $s_{\mathrm{id}} = a$). Thereafter, run the above simulation algorithm for partial key extraction, taking id and S_{id} as inputs to obtain partial keys P_{id} and d_{id}. Finally, return $\mathrm{PK_{id}} = (S_{\mathrm{id}}, P_{\mathrm{id}})$ as the answer. Note that in this case, element d_{id} of private key $\mathrm{SK_{id}}$ is controlled by $\mathcal{B}$.

2. If $\mathrm{id} \neq \mathrm{id}_j$, choose $s_{\mathrm{id}} \leftarrow_R Z_q^*$, and compute $S_{\mathrm{id}} = g^{s_{\mathrm{id}}}$. Thereafter, run the above simulation algorithm for partial key extraction, taking id

and S_{id} as inputs to obtain partial keys P_{id} and d_{id}. Also, add tuple $\langle \mathrm{id}, \mathrm{PK_{id}} = (S_{\mathrm{id}}, P_{\mathrm{id}})\rangle$ to L_{P}, and $\langle \mathrm{id}, \mathrm{SK_{id}} = (s_{\mathrm{id}}, d_{\mathrm{id}})\rangle$ to L_{S}. Finally, return $\mathrm{PK_{id}}$ as the answer.

Private-Key-Extract: On receiving a private key extraction query (id, private key extract):

If $\langle \mathrm{id}, \mathrm{SK_{id}}\rangle$ can be found from L_{S} with index id, then return $\mathrm{SK_{id}}$ as the answer; otherwise, do the following:

1. If $\mathrm{id} = \mathrm{id}_j$, return an invalid answer $\perp$ and terminate.

2. If $\mathrm{id} \neq \mathrm{id}_j$, run the above simulation algorithm for public key extraction, taking id as input (Note that in this query, a tuple $\langle \mathrm{id}, \mathrm{SK_{id}}\rangle$ will be added to L_{S}), and search a tuple $\langle \mathrm{id}, \mathrm{SK_{id}}\rangle$ from L_{S} with index id. Finally, return $\mathrm{SK_{id}}$ as the answer.

Replace-Public-Key-Queries: $\mathcal{A}_1$ can replace public key $\mathrm{PK_{id}}$ of any user with its own.

Decryption-Queries: On receiving a decryption query (id, $c = (U_1, U_2, e, v, S)$, decryption), search a tuple $\langle \mathrm{id}, \mathrm{PK_{id}}\rangle$ from L_{P} with index id, and there are three cases:

1. If $\langle \mathrm{id}, \mathrm{PK_{id}} = (S_{\mathrm{id}}, P_{\mathrm{id}})\rangle \in L_{\mathrm{P}}$ and $\mathrm{id} \neq \mathrm{id}_j$:

(1) Search a tuple $\langle \mathrm{id}, \mathrm{SK_{id}} = (s_{\mathrm{id}}, d_{\mathrm{id}})\rangle$ from L_{S} with index id, and compute $V' = U_1^{s_{\mathrm{id}}} U_2^{\mu d_{\mathrm{id}}}$, where $\mu = H(U_1, U_2, e, S)$.

(2) If $g^v = U_1^{k_1' H_2(e)} U_2^{k_2'}$, where $(k_1', k_2') = \mathrm{KDF}(V')$, return $M = \mathrm{Ext}(U_1^{d_{\mathrm{id}}} U_2^{s_{\mathrm{id}}}, S) \oplus e$; otherwise, return an invalid answer $\perp$ and terminate.

2. If $\langle \mathrm{id}, \mathrm{PK_{id}} = (S_{\mathrm{id}}, P_{\mathrm{id}})\rangle \in L_{\mathrm{P}}$ and $\mathrm{id} = \mathrm{id}_j$:

Compute $\mu = H(U_1, U_2, e, S)$.

If there exist two values $r_1, r_2 \in Z_q^*$ such that $V' = S_{\mathrm{id}}^{r_1} (P_{\mathrm{id}} {P_{\mathrm{Pub}}}^{h_1})^{r_2 \mu}$, $(k_1', k_2') = \mathrm{KDF}(V')$, and $g^v = U_1^{k_1' H_2(e)} U_2^{k_2'}$, then return $M = \mathrm{Ext}(S_{\mathrm{id}}^{r_2} (P_{\mathrm{id}} {P_{\mathrm{Pub}}}^{h_1})^{r_1}, S) \oplus e$; otherwise, return an invalid answer $\perp$ and terminate.

3. If $\langle \mathrm{id}, \mathrm{PK_{id}} = (A_{\mathrm{id}}, B_{\mathrm{id}})\rangle \notin L_{\mathrm{P}}$, it means that the public key of user id is replaced by $\mathcal{A}_1$.

In this case, we assume that public key $\mathrm{PK_{id}} = (S_{\mathrm{id}}, P_{\mathrm{id}})$ of user id is replaced by a random value $\mathrm{PK_{id}'} = (A_{\mathrm{id}}, B_{\mathrm{id}})$. Compute $\mu = H(U_1, U_2, e, S)$.

If there exist two values $r_1, r_2 \in Z_q^*$ subject to constraints $V' = A_{\mathrm{id}}^{r_1} (B_{\mathrm{id}} {P_{\mathrm{Pub}}}^{h_1})^{r_2 \mu}$, $(k_1', k_2') = \mathrm{KDF}(V')$, and $g^v = U_1^{k_1' H_2(e)} U_2^{k_2'}$, then return $M = \mathrm{Ext}(A_{\mathrm{id}}^{r_2} (B_{\mathrm{id}} {P_{\mathrm{Pub}}}^{h_1})^{r_1}, S) \oplus e$; otherwise, return an invalid answer $\perp$ and terminate.

Leakage-Queries: On receiving an identity id and an efficient computable leakage function $f_i : \{0,1\}^* \to \{0,1\}^{\lambda_i}$, adversary $\mathcal{B}$ can return the corresponding answer $f_i(\mathrm{SK_{id}})$ by using leakage oracle $\mathcal{O}_{\mathrm{SK_{id}}}^{\lambda,k}(\cdot)$ with private key $\mathrm{SK_{id}}$ and leakage function $f_i(\cdot)$. In the leakage query process, the total length of leakage $f_i(\mathrm{SK_{id}})$ all returned from leakage oracle $\mathcal{O}_{\mathrm{SK_{id}}}^{\lambda_C,k}(\cdot)$ on the same private key $\mathrm{SK_{id}}$ must be less than leakage parameter λ, i.e., $\sum_{j=1}^{i} f_j(\mathrm{SK_{id}}) \leq \lambda$. Otherwise, an invalid answer $\perp$ will be obtained.

Challenge: Two equal-length messages M_0, $M_1 \in \mathcal{M}$ and a challenge identity $\mathrm{id}^* \in \mathrm{ID}$ are submitted by adversary $\mathcal{A}_1$, where identity id^* has never appeared in a private key extraction query or a leakage query with at most λ-bit leakage.

If $\mathrm{id}^* \neq \mathrm{id}_j$, return an invalid answer $\perp$ and terminate. Otherwise, do the following:

1. Perform a partial key extraction query taking id^* as input, and search two tuples $\langle \mathrm{id}^*, S_{\mathrm{id}^*}, P_{\mathrm{id}^*}, h_1\rangle$ and $\langle \mathrm{id}^*, S_{\mathrm{id}^*}, P_{\mathrm{id}^*}, d_{\mathrm{id}^*}\rangle$ from L_1 and L_{p} with index id^*, respectively. Note that $S_{\mathrm{id}^*} = g^a$.

2. Choose $r_1 \leftarrow_R Z_q^*$, compute $U_1^* = g^{r_1}$, and set $U_2^* = g^b$ (implicitly setting $r_2 = b$).

3. Choose $S^* \leftarrow_R \{0,1\}^{l_{\mathrm{t}}}$ and compute $e^* = \mathrm{Ext}(g^c (U_1^*)^{d_{\mathrm{id}^*}}, S^*) \oplus M_b$ and $V^* = (S_{\mathrm{id}^*})^{r_1 \mu} (U_2^*)^{d_{\mathrm{id}^*}}$, where $b \leftarrow_R \{0,1\}$ and $\mu = H(U_1^*, U_2^*, e^*, S^*)$.

4. Choose $v^* \leftarrow_R Z_q^*$ such that $g^{v^*} = {U_1^*}^{k_1 H_2(e^*)} {U_2^*}^{k_2}$, where $(k_1, k_2) = \mathrm{KDF}(V^*)$.

5. Finally, return $c_b = (U_1^*, U_2^*, e^*, v^*, S^*)$ as a challenge ciphertext of M_b to $\mathcal{A}_1$.

Test stage 2: In this phase, adversary $\mathcal{B}$ answers $\mathcal{A}_1$'s queries in the same way as it did in test stage 1. However, in this stage, id^* cannot be issued as a partial or private key extraction query, whereas $\mathcal{A}_1$ can replace the public key freely. In addition, no decryption queries should be made on ciphertext c_b and identity id^*, and the leakage queries are banned.

The decryption queries can be answered in the same way as in test stage 1. Here, we just repeat the following important case:

Decryption-Queries. On receiving a decryption query (id^*, c, decryption), where $c = (U_1, U_2, e, v, S)$ and $c \neq c_b$:

Search a tuple $\langle \mathrm{id}, S_{\mathrm{id}^*}, P_{\mathrm{id}^*}, h_1\rangle$ from L_1 with index id^*, and compute $\mu = H(U_1, U_2, e, S)$.

If there exist two values $r_1, r_2 \in Z_q^*$ subject to constraints $(k_1', k_2') = \mathrm{KDF}(S_{\mathrm{id}^*}^{r_1} (P_{\mathrm{id}^*} {P_{\mathrm{Pub}}}^{h_1})^{r_2 \mu})$ and $g^v = U_1^{k_1' H_2(e)} U_2^{k_2'}$, then return $M =$

$\mathrm{Ext}(S_{\mathrm{id}^*}^{r_2}(P_{\mathrm{id}^*}{P_{\mathrm{Pub}}}^{h_1})^{r_1}, S) \oplus e$. Otherwise, return an invalid answer $\perp$.

Output: Finally, $\mathcal{A}_1$ outputs b' as the guess of random bit b. If $b' = b$, then $\mathcal{B}$ outputs $\gamma = 1$ (which means $ab = c$); otherwise, it outputs $\gamma = 0$ (which means $ab \neq c$).

Lemma 4 is proved through Claims 1 and 2. In Claim 1, adversary $\mathcal{B}$ is given a tuple (g, g^a, g^b, g^c), which is just a DH instance; in this case, a normal ciphertext will be received by $\mathcal{A}_1$. However, in Claim 2, tuple (g, g^a, g^b, g^c) is a non-DH instance; in this case, $\mathcal{A}_1$ receives an abnormal ciphertext, and can output a correct guess only with the help of leakage on the private key.

Claim 1 If tuple (g, g^a, g^b, g^c) is a DH instance, and adversary $\mathcal{A}_1$ can break the security of our LR-CL-PKE scheme Π with non-ignorable advantage ε_1, then adversary $\mathcal{B}$ can be used to break the hardness of the DDH assumption. Advantage $\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Claim1}}(k)$ of adversary $\mathcal{B}$ is described as

$$\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Claim1}}(k) = \left(1-\frac{q_{\mathrm{S}}}{2^k}\right)\left(1-\frac{1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1}\right)^{q_{\mathrm{D}}+q_{\mathrm{S}}} \cdot \frac{\varepsilon_1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1}.$$

Because (g, g^a, g^b, g^c) is a DH instance, the above interaction game is identical to that of the actual attack from the adversary's view; i.e., the real and the simulated attacks are identical. Therefore, if (g, g^a, g^b, g^c) is a DH instance, $\mathcal{A}_1$ cannot obtain any valuable information on the private key in the above interaction game.

Let $\mathcal{E}_1$ be the event that $\mathcal{B}$ does not terminate in the query stage, $\mathcal{E}_2$ the event that $\mathcal{B}$ does not terminate in the challenge stage, and $\mathcal{E}_3$ the event that $\mathcal{A}_1$ does not submit a private key extraction query with challenge identity id^*. In the above game, $\mathcal{A}_1$ chooses q_{D} user identities in Decryption-Queries, chooses q_{S} user identities in Private-Key-Extract, and chooses an identity in the challenge stage. Thus, we can obtain

$$\begin{cases} \Pr[\mathcal{E}_1] = (1-\delta)^{q_{\mathrm{D}}+q_{\mathrm{S}}}, \\ \Pr[\mathcal{E}_2] = \delta, \\ \Pr[\mathcal{E}_3] = 1 - \dfrac{q_{\mathrm{S}}}{2^k}, \end{cases}$$

where $\delta = \frac{1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1}$ denotes the probability that $\mathcal{A}_1$ chooses a challenge identity id^*, such that $\mathrm{id}^* = \mathrm{id}_j$.

Therefore, the probability that $\mathcal{B}$ does not terminate and $\mathcal{A}_1$ does not submit a private key extraction query with challenge identity id^* is described as

$$\begin{aligned} \Pr[\mathcal{E}_1 \wedge \mathcal{E}_2 \wedge \mathcal{E}_3] &= \left(1-\frac{q_{\mathrm{S}}}{2^k}\right)(1-\delta)^{q_{\mathrm{D}}+q_{\mathrm{S}}}\delta \\ &= \left(1-\frac{q_{\mathrm{S}}}{2^k}\right)\left(1-\frac{1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1}\right)^{q_{\mathrm{D}}+q_{\mathrm{S}}} \\ &\quad \cdot \frac{1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1}. \end{aligned}$$

As discussed above, if $\mathcal{A}_1$ can break our proposal Π with non-ignorable advantage ε_1, $\mathcal{B}$ does not terminate and $\mathcal{A}_1$ does not submit a private key extraction query with challenge identity id^*. By ignoring the key leakage functions, we find that adversary $\mathcal{B}$ can solve the DDH problem with noticeable advantage $\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Claim1}}(k) = (1-\frac{q_{\mathrm{S}}}{2^k})(1-\frac{1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1})^{q_{\mathrm{D}}+q_{\mathrm{S}}}\frac{\varepsilon_1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1}$.

Note that when $q_{\mathrm{D}} + q_{\mathrm{S}}$ is large enough, $(1-\frac{1}{q_{\mathrm{D}}+q_{\mathrm{S}}+1})^{q_{\mathrm{D}}+q_{\mathrm{S}}}$ is close to $1/\mathrm{e}$, where e is the base of the natural logarithm. Therefore, we can obtain

$$\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Claim1}}(k) \geq \left(1-\frac{q_{\mathrm{S}}}{2^k}\right)\frac{\varepsilon_1}{\mathrm{e}(q_{\mathrm{D}}+q_{\mathrm{S}}+1)}.$$

Claim 2 If tuple (g, g^a, g^b, g^c) is a non-DH instance, and adversary $\mathcal{B}$ can be used to solve the DDH problem with the help of the leakage on the private key, then advantage $\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Claim2}}(\lambda, k)$ of adversary $\mathcal{B}$ is described as follows:

$$\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Claim2}}(\lambda, k) \leq \frac{2^{\lambda} q_{\mathrm{D}}}{q - q_{\mathrm{D}} + 1} + 2^{\frac{\lambda}{2}-1}.$$

From now on, let a ciphertext $c = (U_1, U_2, e, v, S)$ be a valid ciphertext if $\log_g U_1 \neq \log_g U_2$, and set a ciphertext $c' = (U_1', U_2', e', v', S')$ as an invalid ciphertext if $\log_g U_1 = \log_g U_2$ even if it can pass the decryption algorithm consistency check. Let Reject be the event in which decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ rejects all invalid ciphertexts in the decryption query stage.

Next, we consider two different cases in which event Reject occurs or does not occur.

1. If tuple (g, g^a, g^b, g^c) is a non-DH instance and decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ rejects all invalid ciphertexts (i.e., event Reject occurs), then the advantage of adversary $\mathcal{B}$ breaking the DDH assumption is described as $\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Reject}}(\lambda, k) \leq 2^{\frac{\lambda}{2}-1}$.

Let Leak denote the output information of all leakage functions $f_i(\mathrm{SK}_{\mathrm{id}^*})(i \geq 1)$ acting on private

key $\mathrm{SK}_{\mathrm{id}^*}$ of challenge identity id^*. The value of Leak reaches 2^λ at most, i.e., $|\mathrm{Leak}| \leq 2^\lambda$.

In our proposal, all of the ciphertext elements will be random from the adversary's view, and no PPT adversary can be leaked on the private key from the corresponding ciphertext. Adversary $\mathcal{A}_1$'s view consists of public key $\mathrm{PK} = (S_{\mathrm{id}^*}, P_{\mathrm{id}^*})$, challenge ciphertext $c_b = (U_1^*, U_2^*, e^*, v^*, S^*)$, and λ-bit leakage Leak on private key $\mathrm{SK}_{\mathrm{id}^*} = (s_{\mathrm{id}^*}, d_{\mathrm{id}^*})$. Furthermore, $\mathcal{A}_1$ cannot obtain any valuable information on the private key by performing a decryption query for an valid ciphertext. Because all of the invalid ciphertext will be rejected by decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$, $\mathcal{A}_1$ can output a guess only with the help of leakage on the private key.

According to Lemma 1, we can obtain

$$\begin{aligned}&\widetilde{H}_\infty(\mathrm{SK}_{\mathrm{id}^*}|\mathrm{PK}_{\mathrm{id}^*}, c_b, \mathrm{Params}, \mathrm{Leak})\\&= \widetilde{H}_\infty(s_{\mathrm{id}^*}, d_{\mathrm{id}^*}|S_{\mathrm{id}^*}, \mathrm{Leak})\\&\geq \log q - \lambda,\end{aligned}$$

where $s_{\mathrm{id}^*}, d_{\mathrm{id}^*} \leftarrow_R Z_q^*$. Furthermore, P_{id^*}, c_b, and Params do not contain any valuable information on private key $\mathrm{SK}_{\mathrm{id}^*}$.

Thus, the probability that any PPT adversary guesses private key $\mathrm{SK}_{\mathrm{id}^*}$ is at most

$$2^{-\widetilde{H}_\infty(\mathrm{SK}_{\mathrm{id}^*}|\mathrm{PK}_{\mathrm{id}^*}, c_b, \mathrm{Params}, \mathrm{Leak})} \leq \frac{2^\lambda}{q}.$$

According to Lemma 2, we find that the advantage of $\mathcal{A}_1$ outputting M_b is bounded by $\frac{1}{2}\sqrt{q\frac{2^\lambda}{q}} = 2^{\frac{\lambda}{2}-1}$. Therefore, if tuple (g, g^a, g^b, g^c) is a non-DH instance and decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ rejects all invalid ciphertext, then we find that the advantage of adversary $\mathcal{B}$ breaking the hardness of the DDH assumption is described as follows:

$$\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Reject}}(\lambda, k) = |\Pr[b' = b|\mathrm{Reject}] \leq 2^{\frac{\lambda}{2}-1}.$$

In our LR-CL-PKE scheme, we find that given $\mathrm{Params}, c_b, \mathrm{PK}_{\mathrm{id}^*}$, and λ-bit leakage on secret key $\mathrm{SK}_{\mathrm{id}^*}$, the average minimum entropy of variable $U_1^{d_{\mathrm{id}^*}} U_2^{s_{\mathrm{id}^*}}$ is at least $\log q - \lambda$. Because $\mathrm{Ext} : G \times \{0,1\}^{l_{\mathrm{t}}} \to \{0,1\}^{l_{\mathrm{m}}}$ is an average-case $(\log q - \lambda, \varepsilon)$-strong extractor, and according to Lemma 3, we find that $l_{\mathrm{m}} \leq \log q - \lambda - 2\log(1/\varepsilon)$. Because ε is negligible (i.e., $2\log(1/\varepsilon) = \omega(\log k)$), we will have $\lambda \leq \log q - l_{\mathrm{m}} - \omega(\log k)$.

2. If tuple (g, g^a, g^b, g^c) is a non-DH instance and decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ does not reject all invalid ciphertext in the decryption query stage (i.e., event Reject does not occur), then the probability that event Reject does not occur is described as $\Pr[\overline{\mathrm{Reject}}] \leq \frac{2^\lambda q_{\mathrm{D}}}{q - q_{\mathrm{D}} + 1}$.

Let $c' = (U_1', U_2', e', v', S')$ be the first invalid ciphertext submitted by adversary $\mathcal{A}_1$. Now we consider the following two different cases about the first invalid ciphertext c', which would be accepted by decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$:

1. $(U_1', U_2', e', S') \neq (U_1^*, U_2^*, e^*, S^*)$ and $\mu' = \mu$, where $\mu' = H(U_1', U_2', e', S')$ and $\mu = H(U_1^*, U_2^*, e^*, S^*)$, which means a hash collision occurs. According to Definition 1, we find that the advantage of adversary $\mathcal{B}$ breaking the target collision resistance of hash function H is negligible. Therefore, this case cannot occur.

2. $(U_1', U_2', e', S') \neq (U_1^*, U_2^*, e^*, S^*)$ and $\mu' \neq \mu$. In this case, we will discuss the probability that adversary $\mathcal{A}_1$ creates an invalid ciphertext that can be accepted by decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$.

Because $\widetilde{H}_\infty(\mathrm{SK}_{\mathrm{id}^*}|\mathrm{PK}_{\mathrm{id}^*}, c_b, \mathrm{Params}, \mathrm{Leak}) \geq \log q - \lambda$, we find that adversary $\mathcal{A}_1$ guesses correctly for private key $\mathrm{SK}_{\mathrm{id}^*}$ with a probability of at most $2^\lambda/q$. Furthermore, given public key $\mathrm{PK}_{\mathrm{id}^*}$ and challenge ciphertext c_b, each invalid ciphertext c' determines a unique tuple $(s_{\mathrm{id}^*}, d_{\mathrm{id}^*}) \in Z_q^* \cdot Z_q^*$. Therefore, if decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ rejects the first invalid ciphertext, adversary $\mathcal{A}_1$ may learn more valuable information on the private key $\mathrm{SK}_{\mathrm{id}^*}$, and the probability that $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ accepts the second invalid ciphertext increases to $\frac{2^\lambda}{q-1}$. If more invalid ciphertext is rejected by $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$, the probability increases. Thus, the probability that $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ accepts the i^{th} invalid ciphertext is $\frac{2^\lambda}{q-i+1}$. Therefore, the probability that $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ accepts an invalid ciphertext is at most $\frac{2^\lambda}{q-q_{\mathrm{D}}+1}$, considering that there are q_{D} decryption queries in all. Thus, we find that the probability that $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ does not reject all invalid ciphertext is at most $\frac{2^\lambda q_{\mathrm{D}}}{q-q_{\mathrm{D}}+1}$.

As discussed above, if tuple (g, g^a, g^b, g^c) is a non-DH instance, and decryption oracle $\mathrm{Dec}^{\mathcal{O}}(\mathrm{SK}_{\mathrm{id}}, \cdot)$ does not reject all invalid ciphertext, we will obtain $\Pr[\overline{\mathrm{Reject}}] \leq \frac{2^\lambda q_{\mathrm{D}}}{q-q_{\mathrm{D}}+1}$.

Therefore, from the results of the above discussion, we can obtain

$$\begin{aligned}&\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1,\mathrm{Claim2}}(\lambda,k)=|\Pr[b'=b]-\frac{1}{2}|\\&\le|\Pr[b'=b|\mathrm{Reject}]+\Pr[\overline{\mathrm{Reject}}]|\\&\le 2^{\frac{\lambda_C}{2}-1}+\frac{2^\lambda q_\mathrm{D}}{q-q_\mathrm{D}+1}.\end{aligned}$$

Thus, from Claims 1 and 2, we find

$$\begin{aligned}&\left(1-\frac{q_\mathrm{S}}{2^k}\right)\left(1-\frac{1}{q_\mathrm{D}+q_\mathrm{S}+1}\right)^{q_\mathrm{D}+q_\mathrm{S}}\frac{\varepsilon_1}{q_\mathrm{D}+q_\mathrm{S}+1}\\&\le\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1}(\lambda,k)\le\left(1-\frac{q_\mathrm{S}}{2^k}\right)\left(1-\frac{1}{q_\mathrm{D}+q_\mathrm{S}+1}\right)^{q_\mathrm{D}+q_\mathrm{S}}\\&\cdot\frac{\varepsilon_1}{q_\mathrm{D}+q_\mathrm{S}+1}+\frac{2^\lambda q_\mathrm{D}}{q-q_\mathrm{D}+1}+2^{\frac{\lambda}{2}-1}.\end{aligned}$$

When $q_\mathrm{D}+q_\mathrm{S}$ is large enough, we have

$$\begin{aligned}&\left(1-\frac{q_\mathrm{S}}{2^k}\right)\left(1-\frac{1}{q_\mathrm{D}+q_\mathrm{S}+1}\right)^{q_\mathrm{D}+q_\mathrm{S}}\frac{\varepsilon_1}{q_\mathrm{D}+q_\mathrm{S}+1}\\&\ge\left(1-\frac{q_\mathrm{S}}{2^k}\right)\frac{\varepsilon_1}{\mathrm{e}\,(q_\mathrm{D}+q_\mathrm{S}+1)}.\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_1}(\lambda,k)\in\Bigg[&\left(1-\frac{q_\mathrm{S}}{2^k}\right)\frac{\varepsilon_1}{\mathrm{e}\,(q_\mathrm{D}+q_\mathrm{S}+1)},\\&\left(1-\frac{q_\mathrm{S}}{2^k}\right)\left(1-\frac{1}{q_\mathrm{D}+q_\mathrm{S}+1}\right)^{q_\mathrm{D}+q_\mathrm{S}}\\&\cdot\frac{\varepsilon_1}{q_\mathrm{D}+q_\mathrm{S}+1}+\frac{2^\lambda q_\mathrm{D}}{q-q_\mathrm{D}+1}+2^{\frac{\lambda}{2}-1}\Bigg].\end{aligned}$$

In summary, if the assumed DDH is intractable and hash function H is target collision resistant, then we can assume that for any leakage parameter $\lambda\le\log q-l_\mathrm{m}-\omega(\log k)$, our LR-CL-PKE scheme $\Pi=(\mathrm{Setup},\mathrm{KeyGen},\mathrm{Enc},\mathrm{Dec})$ is CCA2-secure resistant against a Type I adversary.

Lemma 5 If the DDH assumption is intractable, and the one-way hash function H is target collision resistant, then for any leakage parameter $\lambda\le\log q-l_\mathrm{m}-\omega(\log k)$, our LR-CL-PKE scheme Π is CCA2-secure resistant against the Type II adversary.

Note that in the setup stage of Game_2, challenger $\mathcal{C}$ sends public parameter Params and master secret key S_msk to adversary $\mathcal{A}_2$. This is the only difference between Lemmas 4 and 5. However, in the proof of Lemma 4, no exponent of challenge instance (g,g^a,g^b,g^c) is contained in the master secret key; i.e., adversary $\mathcal{A}_2$ can obtain the complete master secret key from challenger $\mathcal{C}$. Thus, we can use the same method to prove Lemma 5.

In summary, we find that if a Type II adversary $\mathcal{A}_2$ can break the security of our proposal Π with non-ignorable advantage ε_2, then there exists an adversary $\mathcal{B}$ that can break the hardness of the classical DDH assumption with obvious advantage $\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_2}(\lambda,k)$, described as

$$\begin{aligned}\mathrm{Adv}^{\mathrm{DDH}}_{\mathcal{B},\mathcal{A}_2}(\lambda,k)\in\Bigg[&\left(1-\frac{q_\mathrm{S}}{2^k}\right)\frac{\varepsilon_1}{\mathrm{e}\,(q_\mathrm{D}+q_\mathrm{S}+1)},\\&\left(1-\frac{q_\mathrm{S}}{2^k}\right)\left(1-\frac{1}{q_\mathrm{D}+q_\mathrm{S}+1}\right)^{q_\mathrm{D}+q_\mathrm{S}}\\&\cdot\frac{\varepsilon_1}{q_\mathrm{D}+q_\mathrm{S}+1}+\frac{2^\lambda q_\mathrm{D}}{q-q_\mathrm{D}+1}+2^{\frac{\lambda}{2}-1}\Bigg].\end{aligned}$$

Due to lack of space, the proof of Lemma 5 is omitted.

6 Comparisons

In this section, we compare our results to the best prior constructions (Naor and Segev, 2012; Li et al., 2013; Liu et al., 2013; Xiong et al., 2013; Qin et al., 2015) in terms of basic parameters' performance and computation efficiency.

6.1 Performance analysis

Table 1 shows that in these schemes (Naor and Segev, 2012; Li et al., 2013; Liu et al., 2013; Qin et al., 2015), some elements of ciphertext can be written as a function of private key; i.e., these constructions cannot ensure that all elements of the ciphertext are random. Thus, in the leakage setting, any PPT adversary can be leaked on the private key from the corresponding ciphertext. However, in our proposal, from the adversary's view, all of ciphertext elements will be random, and no PPT adversary can be leaked on the private key from the corresponding ciphertext.

In the bounded-leakage model, the existing LR-CL-PKE scheme (Xiong et al., 2013) is constructed based on composite-order bilinear groups, and can achieve only CCA1 security. However, our proposal can maintain CCA2 security in the leakage setting, and the length-of-bits leakage is up to $\lambda\le\log q-l_\mathrm{m}-\omega(\log k)$. In addition, our proposal is presented based on the prime-order group. In particular, the striking advantage of our method is the key leakage ratio, which can be upto $1/2$.

Note that LR-CL-PKE-Xiong is constructed based on composite-order bilinear groups.

Therefore, in Table 1, the lengths of the private key, the public key, and the ciphertext are described as $3(n+2)|G|$, $3|G_T|+|\pi|$, and $3(n+2)|G|+3|G_T|$, respectively, because G and G_T are two groups of order prime q.

6.2 Efficiency analysis

Table 2 summarizes the computation costs of the above-mentioned constructions. The computation efficiency is determined by the computation costs of algorithms KeyGen, Enc, and Dec. When evaluating the computation efficiency, the hash function and the exclusive-or (XOR) operation are ignored.

From Table 2, we find that these existing constructions (Naor and Segev, 2012; Li et al., 2013; Liu et al., 2013; Qin et al., 2015) and our proposal have the same computation efficiency; however, our proposal performs better because no PPT adversary can obtain valuable information on the private key from the corresponding ciphertext, and the key leakage ratio can be up to $1/2$.

Compared with the other constructions, LR-CL-PKE-Xiong is constructed based on composite-order bilinear groups, and the computation efficiency is lower.

7 Conclusions

In the existing constructions (Naor and Segev, 2012; Li et al., 2013; Liu et al., 2013; Qin et al., 2015), any PPT adversary can be leaked on the private key from the corresponding ciphertext. Also, the LR-CL-PKE scheme (Xiong et al., 2013) is constructed based on composite-order bilinear groups, and cannot achieve CCA2 security. To achieve better performance, we introduced a new method to construct a more practical LR-CL-PKE scheme without sacrificing CCA2 security. We presented a concrete construction and proved its security based on the hardness of the classical DDH assumption. Security analysis showed that our proposal not only resists leakage attacks, but also achieves better performance, because no PPT adversary can be leaked on the private key from the corresponding ciphertext. The key leakage ratio of our proposal is the best among all these existing constructions (Naor and Segev, 2012; Li et al., 2013; Liu et al., 2013; Xiong et al., 2013;

Table 1 Comparison of basic parameters

Scheme	l_{SK}	l_{PK}	l_{C}	SecLev	Leakage parameter	Assumption	L_{Ratio}
LR-PKE-NS	$6\|q\|$	$3\|G\|$	$3\|G\|+l_{\mathrm{t}}+l_{\mathrm{m}}$	CCA2	$\log q-l_{\mathrm{m}}-\omega(\log k)$	DDH	1/6
LR-PKE-Liu	$6\|q\|$	$3\|G\|$	$4\|G\|+\|q\|$	CCA2	$\log q-\omega(\log k)$	DDH	1/6
LR-PKE-Li	$4\|q\|$	$2\|G\|$	$3\|G\|+l_{\mathrm{t}}+l_{\mathrm{m}}$	CCA2	$\log q-l_{\mathrm{m}}-\omega(\log k)$	DDH	1/4
LR-PKE-Qin	$4\|q\|$	$2\|G\|$	$3\|G\|+l_{\mathrm{t}}+l_{\mathrm{SE}}$	CCA2	$\log q-l_{\mathrm{k}}-\omega(\log k)$	DDH	1/4
LR-CL-PKE-Xiong	$3(n+2)\|G\|$	$3\|G_T\|+\|\pi\|$	$3(n+2)\|G\|+3\|G_T\|$	CCA1	$(n-2c-1)\log q$	$\mathrm{IA}_{\mathrm{COBG}}$	1/3
Our proposal Π	$2\|q\|$	$2\|G\|$	$2\|G\|+2\|q\|+l_{\mathrm{m}}$	CCA2	$\log q-l_{\mathrm{m}}-\omega(\log k)$	DDH	1/2

1. Let l_{SK} denote the length of the private key, l_{PK} the length of the public key, l_{C} the length of the ciphertext, SecLev the security level, and $L_{\mathrm{Ratio}}=\dfrac{\text{size of leakage}}{\text{size of private key}}$ the key leakage ratio
2. Let $|G|$ and $|G_T|$ denote the length of the element in groups G and G_T respectively, $|q|$ the length of the element in field Z_q^*, l_{t} the length of the random seed, l_{m} the length of the plaintext, l_{k} the length of the encapsulation key, l_{SE} the length of the ciphertext generated through a symmetric encryption scheme, $|\pi|$ the length of the proof created through a noninteractive zero-knowledge proof system, and $\mathrm{IA}_{\mathrm{COBG}}$ the intractability assumptions of composite-order bilinear groups
3. In LR-CL-PKE-Xiong, $n\geq 2$ is an integer, and c is any fixed positive constant

Table 2 Comparison of computation efficiency

Scheme	KeyGen	Enc	Dec	Total
LR-PKE-NS	$3E_{\mathrm{D}}$	$3E_{\mathrm{S}}+E_{\mathrm{D}}+E_{\mathrm{Ext}}$	$2E_{\mathrm{D}}+E_{\mathrm{Ext}}$	$3E_{\mathrm{S}}+6E_{\mathrm{D}}+2E_{\mathrm{Ext}}$
LR-PKE-Liu	$3E_{\mathrm{D}}$	$2E_{\mathrm{S}}+2E_{\mathrm{D}}$	$2E_{\mathrm{D}}$	$2E_{\mathrm{S}}+7E_{\mathrm{D}}$
LR-PKE-Li	$2E_{\mathrm{D}}$	$3E_{\mathrm{S}}+E_{\mathrm{D}}+E_{\mathrm{Ext}}$	$2E_{\mathrm{D}}+E_{\mathrm{Ext}}$	$3E_{\mathrm{S}}+5E_{\mathrm{D}}+2E_{\mathrm{Ext}}$
LR-PKE-Qin	$2E_{\mathrm{D}}$	$2E_{\mathrm{S}}+2E_{\mathrm{D}}+E_{\mathrm{Ext}}+E_{\mathrm{SE}}$	$2E_{\mathrm{D}}+E_{\mathrm{Ext}}+E_{\mathrm{SE}}$	$2E_{\mathrm{S}}+6E_{\mathrm{D}}+2E_{\mathrm{Ext}}+2E_{\mathrm{SE}}$
LR-CL-PKE-Xiong	$E_{\mathrm{S}}+(2n+4)E_{\mathrm{D}}$	$(n+1)E_{\mathrm{S}}+E_{\mathrm{D}}+E_{\mathrm{e}}$	$(n+2)E_{\mathrm{e}}$	$(n+2)E_{\mathrm{S}}+(2n+5)E_{\mathrm{D}}+(n+3)E_{\mathrm{e}}$
Our proposal Π	$2E_{\mathrm{S}}$	$2E_{\mathrm{S}}+2E_{\mathrm{D}}+E_{\mathrm{Ext}}$	$E_{\mathrm{S}}+3E_{\mathrm{D}}+E_{\mathrm{Ext}}$	$5E_{\mathrm{S}}+5E_{\mathrm{D}}+2E_{\mathrm{Ext}}$

Let E_{SE} denote the cost of the symmetric encryption and decryption, E_{Ext} the cost of the average-case strong extractor, E_{S} the cost of the single exponentiation operation, E_{D} the cost of the double exponentiation operation, and E_{e} the cost of the bilinear pairing operation

Qin et al., 2015). We believe that the good performance makes our proposal have significant value in practical applications.

References

Akavia A, Goldwasser S, Vaikuntanathan V, 2009. Simultaneous hardcore bits and cryptography against memory attacks. 6th Theory of Cryptography Conf, p.474-495. https://doi.org/10.1007/978-3-642-00457-5_28

Al-Riyami SS, Paterson KG, 2003. Certificateless public key cryptography. 9th Int Conf on the Theory and Application of Cryptology and Information Security, p.452-473. https://doi.org/10.1007/978-3-540-40061-5_29

Alwen J, Dodis Y, Wichs D, 2009. Leakage-resilient public-key cryptography in the bounded-retrieval model. 29th Annual Int Conf on Advances in Cryptology, p.36-54. https://doi.org/10.1007/978-3-642-03356-8_3

Chen R, Mu Y, Yang G, et al., 2016a. Strong authenticated key exchange with auxiliary inputs. *Des Cod Crypt*, 85(1):145-173. https://doi.org/10.1007/s10623-016-0295-3

Chen R, Mu Y, Yang G, et al., 2016b. Strongly leakage-resilient authenticated key exchange. Cryptographers' Track at the RSA Conf, p.19-36. https://doi.org/10.1007/978-3-319-29485-8_2

Cramer R, Shoup V, 2003. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J Comput*, 33(1):167-226. https://doi.org/10.1137/S0097539702403773

Dodis Y, Ostrovsky R, Reyzin L, et al., 2008. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. *SIAM J Comput*, 38(1):97-139. https://doi.org/10.1137/060651380

Li J, Teng M, Zhang Y, et al., 2016. A leakage-resilient CCA-secure identity-based encryption scheme. *Comput J*, 59(7):1066-1075. https://doi.org/10.1093/comjnl/bxv128

Li S, Zhang F, Sun Y, et al., 2013. Efficient leakage-resilient public key encryption from DDH assumption. *Clust Comput*, 16(4):797-806. https://doi.org/10.1007/s10586-013-0253-z

Liu S, Weng J, Zhao Y, 2013. Efficient public key cryptosystem resilient to key leakage chosen ciphertext attacks. Cryptographers' Track at the RSA Conf, p.84-100. https://doi.org/10.1007/978-3-642-36095-4_6

Naor M, Segev G, 2012. Public-key cryptosystems resilient to key leakage. *SIAM J Comput*, 41(4):772-814. https://doi.org/10.1137/100813464

Qin B, Liu S, Chen K, 2015. Efficient chosen-ciphertext secure public-key encryption scheme with high leakage-resilience. *IET Inform Secur*, 9(1):32-42. https://doi.org/10.1049/iet-ifs.2013.0173

Shamir A, 1984. Identity-based cryptosystems and signature schemes. Workshop on the Theory and Application of Cryptographic Techniques, p.47-53. https://doi.org/10.1007/3-540-39568-7_5

Xiong H, Yuen T, ZHANG C, et al., 2013. Leakage-resilient certificateless public key encryption. Proc 1st ACM Workshop on Asia Public-Key Cryptography, p.13-22. https://doi.org/10.1145/2484389.2484394

Yu Q, Li J, Zhanga Y, et al., 2016. Certificate-based encryption resilient to key leakage. *J Syst Softw*, 116:101-102. https://doi.org/10.1016/j.jss.2015.05.066

Zhou Y, Yang B, Zhang W, 2016. Provably secure and efficient leakage-resilient certificateless signcryption scheme without bilinear pairing. *Disc Appl Math*, 204:185-202. https://doi.org/10.1016/j.dam.2015.10.018