IEEE Open Journal of the
Computer Society

Received 11 May 2020; accepted 9 July 2020. Date of publication 14 July 2020;
date of current version 30 July 2020. The review of this paper was arranged by Associate Editor Zhou Su.

Digital Object Identifier 10.1109/OJCS.2020.3008961

Leakage-Resilient Certificate-based Key Encapsulation Scheme Resistant to Continual Leakage

JUI-DI WU[1], YUH-MIN TSENG[1], SEN-SHAN HUANG[1], AND TUNG-TSO TSAI[2]

[1]Department of Mathematics, National Changhua University of Education, Chang-Hua 500, Taiwan
[2]Department of Research, Foxconn, Taipei City 114, Taiwan

CORRESPONDING AUTHOR: YUH-MIN TSENG. (e-mail: ymtseng@cc.ncue.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under contract no. MOST108-2221-E-018-004-MY2.

ABSTRACT In the past, the security of most public-key encryption or key encapsulation schemes is shown in an ideal model, where private keys, secret keys and random values are assumed to be absolutely secure to adversaries. However, this ideal model is not practical due to side-channel attacks in the sense that adversaries could gain partial information of these secret values involved in decryption operations by perceiving energy consumption or execution timing. In such a case, these schemes under the ideal model could suffer from side-channel attacks. Recently, leakage-resilient cryptography resistant to side-channel attacks is an emerging research topic. Certificate-based encryption (CBE) or certificate-based key encapsulation (CB-KE) schemes are a class of important public-key encryption. However, little work addresses the design of leakage-resilient CBE (LR-CBE) or leakage-resilient CB-KE (LR-CB-KE) schemes. In this paper, we present the *first* LR-CB-KE scheme with overall unbounded leakage property which permits adversaries to continuously gain partial information of the system secret key of a trusted certificate authority (CA), the private keys and certificates of users, and random values. In the generic bilinear group model, formal security analysis is made to prove that the proposed LR-CB-KE scheme is secure against chosen ciphertext attacks.

INDEX TERMS Leakage resilience, side-channel attacks, key encapsulation, public-key encryption, certificate-based public-key setting.

I. INTRODUCTION

In traditional public-key settings [1], [2], the certificate of a user is used to create a link between her/his identity and public key while a public-key infrastructure (PKI) is constructed to manage certificates of all users. Identity (ID)-based public-key settings [3], [4] were presented to eliminate the costs of both the PKI construction and certificate management. Unfortunately, all ID-based public-key settings have an inborn drawback, called the key escrow problem, in the sense that the private keys of all users are produced and known by a private key generator (PKG). To resolve the key escrow problem, Al-Riyami and Paterson [5] presented a new public-key setting, called certificateless public-key setting. Both the ID-based and certificateless public-key settings remove the usage of certificates, but they have to offer extra revocation mechanisms to revoke compromised/misbehaving users [6], [7].

In 2003, Gentry [8] presented the concept of certificate-based public-key setting without extra revocation mechanisms to resolve the key escrow problem. In the certificate-based public-key setting, there are two roles, namely, users and a trusted certificate authority (CA). A user first chooses a private key and produces the corresponding public key. The user then sends her/his identity and the corresponding public key to the CA. By the user's identity and public key, the CA computes and sends the associated certificate to the user. It is worth mentioning that the user must employ both her/his private key and certificate to decrypt a ciphertext and sign a message. In the past, based on the certificate-based public-key settings, numerous cryptographic primitives have been proposed, such as certificate-based encryption [9], [10] and certificate-based signature [11], [12].

Typically, the security of the public-key settings mentioned above is shown in an ideal model, where private keys, secret keys and random values are assumed to be absolutely secure to adversaries. Indeed, the ideal model is not practical due to side-channel attacks [13], [14] in the sense that adversaries could gain partial information of these secret values involved in computation operations by perceiving energy consumption or execution timing. In such a case, these cryptographic schemes based on the ideal model could suffer from such side-channel attacks and be insecure. Recently, the research of leakage-resilient cryptography resistant to side-channel attacks is an emerging research topic and has gained significant attention of cryptographers. These leakage- resilient cryptographic schemes allow adversaries to gain partial information of private keys, secret keys and random values while keeping the security of these leakage-resilient cryptographic schemes. In the past decade, numerous leakage-resilient cryptographic primitives based on the traditional public-key settings have been proposed, such as leakage-resilient signature schemes [15]–[17], leakage-resilient authenticated key exchange protocols [18]–[20] and leakage-resilient encryption schemes [21]–[26].

For leakage-resilient cryptography, there are two leakage models, namely, bounded leakage and continuous leakage models. In both models, adversaries could gain partial information of private keys, secret keys or random values involved in computation operations for each invocation of a cryptographic scheme. In the bounded leakage model [21], [23], the total amount of leakage information during the life time of the cryptographic scheme have to be bounded to a fixed ratio or bit-length. On the other hand, in the continuous leakage model [22], [26]–[28], adversaries are permitted to continuously gain partial information of these secret values for each invocation of the cryptographic scheme while the whole leakage amount is unbounded during the life time of the cryptographic scheme. Obviously, the continuous leakage model has less restrictions and possesses overall unbounded leakage property.

Indeed, certificate-based encryption (CBE) or certificate-based key encapsulation (CB-KE) schemes are a class of important public-key encryption. Up to now, little work addresses the design of leakage-resilient CBE (LR-CBE) or leakage-resilient CB-KE (LR-CB-KE) schemes. In this paper, we aim at the design of the first LR-CB-KE scheme with overall unbounded leakage property which permits adversaries to continuously gain partial information of the system secret key of the CA, the private key and certificate of users, and random values.

A. RELATED WORK

In the section, let us briefly review the related work of leakage-resilient encryption and key encapsulation schemes based on various kinds of public-key settings that includes traditional, ID-based and certificate-based public-key settings.

Based on a traditional public-key setting, Akavia *et al.* [21] presented the first leakage-resilient encryption (LRE) scheme and the associated bounded leakage model. Their LRE scheme is semantically secure against chosen plain-text attacks (CPA). In their bounded leakage model, adversaries are allowed to choose a leakage function with taking as input a user's private key and gain partial information of the user's private key through the output of the leakage function. By following Akavia *et al.*'s bounded leakage model, Naor and Segev [23] proposed a new LRE scheme which is semantically secure under adaptive chosen ciphertext attacks (CCA). Furthermore, they also presented a generic LRE scheme using the universal hash proofs. In order to improve the performance of Naor and Segev's LRE scheme, Li *et al.* [25] and Liu *et al.* [24] respectively proposed an efficient LRE scheme in the bounded leakage model. In the continual leakage model, Kiltz and Pietrzak [22] proposed the first LRE scheme. They employ the generic bilinear group (GBG) model [29] to prove the security of the proposed LRE scheme. Moreover, in the GBG model and the continual leakage model, Galindo *et al.* [26] also presented an efficient ElGamal-like LRE scheme. These LRE schemes mentioned above are constructed under traditional public-key settings.

In 2010, Brakerski *et al.* [30] proposed the first leakage-resilient ID-based encryption (LR-IBE) scheme in the continual leakage model. As mentioned earlier, in ID-based public-key settings, the PKG uses a system secret key to produce the private keys of all users. In such a case, adversaries are allowed to gain partial information of both the system secret key of the PKG in the key extract phase and the private keys of users in the decryption phase. However, Brakerski *et al.*'s continual leakage model does not allow adversaries to gain the system secret key of the PKG in the key extract phase. To remove this restriction, Yuen *et al.* [31] proposed an improvement on Brakerski *et al.*'s LR-IBE scheme. Based on composite order groups, Li *et al.* [32] proposed a new LR-IBE scheme in the post-challenge continuous auxiliary input model. Li *et al.*'s LR-IBE scheme is secure against chosen plain-text attacks in the standard model.

Based on certificate-based public-key settings, little work addresses the design of leakage- resilient certificate-based encryption (LR-CBE) or leakage-resilient certificate-based key encapsulation (LR-CB-KE) schemes. In 2016, the first LR-CBE scheme was proposed by Yu *et al.* [33]. In their LR-CBE scheme, adversaries are allowed to gain partial information of both the CA's system secret key in the certificate generation phase and the user's private key and certificate in the decryption phase. However, their LR-CBE scheme is constructed in the bounded leakage model. In addition, the ratio of leakage information for these secret values is fixed to 1/3. Based on the composite order bilinear group assumption, the security of Yu *et al.*'s scheme is proved to be secure against CCA attacks by using the dual system encryption technique. However, the performance of Yu *et al.*'s scheme is costly due to the dual system encryption technique. In 2018, Guo *et al.* [34] proposed an efficient LR-CBE scheme, but it only allows adversaries to gain partial information of both the user's private key and certificate in the bounded leakage model. In the continuous leakage model, Li *et al.* [35] proposed a new LR-CBE scheme. However, in their scheme, adversaries are allowed to gain only partial information of the user's private key and certificate, but random values involved in decryption phase and the CA's system secret key are disallowed to be leaked to adversaries.

B. CONTRIBUTION AND ORGANIZATION

According to the review above, the leakage models of these existing LR-CBE schemes [33]–[35] have several restrictions and do not offer complete leakage abilities of adversaries. In this paper, we first present a new continuous leakage model of LR-CB-KE schemes. The continuous leakage model is extended from the models of LR-CBE schemes presented in [33]–[35]. The continuous leakage model consists of two kinds of adversaries that include Type I adversary (uncertified entity) and Type II adversary (honest-but-curious CA). In the continuous leakage model, adversaries are allowed to continuously gain partial information of the CA's system secret key involved in the certificate generation phase, the user's private key and certificate involved in the decryption phase, and random values used in both phases. In addition, the whole leakage amount is unbounded during the life time of the LR-CB-KE scheme, namely, the continuous leakage model possesses overall unbounded leakage property.

In the new continuous leakage model, the first LR-CB-KE scheme with overall unbounded leakage property is proposed in this paper. In our scheme, adversaries are given the complete leakage abilities to continuously gain partial information of the CA's system secret key, the user's private key and certificate, and random values. The design principle of the proposed LR-CB-KE scheme is to employ the key refreshing technique [17], [22], [36] to update the CA's system secret key, and each user's private key and certificate after each invocation. In the key refreshing technique, the CA partitions the system secret key into two parts and updates the two parts after each certificate generation procedure. Similarly, each user respectively partitions her/his private key and certificate into two parts and updates them after each decryption procedure. It is worth mentioning that the CA's system public key and each user's public key are still retained unchangeable after the key refreshing procedure. In such a case, adversaries can continuously gain partial information about these two current secret parts, but not the CA's original system secret key, or users' original private keys and certificates. In the generic bilinear group model [29], formal security analysis is made to prove that the proposed LR-CB-KE scheme is secure against the chosen ciphertext attacks for two kinds of adversaries.

The remainder of the paper is organized as follows. Section II presents several preliminaries. In Section III, the syntax and security notions of LR-CB-KE schemes are given. A secure LR-CB-KE scheme resilient to continuous key leakage is proposed in Section IV. We analyze the security of the proposed LR-CB-KE scheme in Section V. Performance analysis is demonstrated in Section VI. Conclusion are discussed in Section VII.

II. PRELIMINARIES

In the section, we introduce several preliminaries that include the concepts of bilinear groups, the notions of generic bilinear group model and the basics of entropy.

A. BILINEAR PAIRINGS

Let $\mathbb{G} = < g >$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of the same prime order p. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$

1) Bilinearity: $\hat{e}(g^x, g^y) = \hat{e}(g, g)^{xy}$, for $x, y \in \mathbb{Z}_p^*$.
2) Non-degeneracy: $\hat{e}(g, g) \neq 1$.
3) Computability: $\hat{e}(u, v)$ is efficiently computable, for all $u, v \in \mathbb{G}$.

For the bilinear map $\hat{e}$, $\mathbb{G}$ is the base group and $\mathbb{G}_T$ denotes the target group. It is worth mentioning that $\hat{e}(g, g)$ is viewed as a generator of $\mathbb{G}_T$. For the detailed properties and implementations of both bilinear groups and bilinear maps, please refer to [4], [6], [37], [38].

B. GENERIC BILINEAR GROUP MODEL

In 1997, Shoup [39] presented the concepts of the generic group model, which is viewed as a security model for cryptographic schemes. In the generic group model, each group element is encoded to a unique string randomly chosen by a challenger. For executing a group operation, adversaries must issue a group query (oracle) to obtain the result. Namely, when an adversary sends two group elements to the challenger to perform the group operation, the challenger produces the resulting group element, and sends it to the adversary while recording it in a list maintained by the challenger. If an adversary can efficiently find a collision encoding of a group operation, we say that the adversary solves the computational hardness assumption, namely, the discrete logarithm problem of the group [40].

By extending the generic group model mentioned above, Boneh *et al.* [29] presented the generic bilinear group model. In this model, there are two groups $\mathbb{G}$ and $\mathbb{G}_T$ and each element of $\mathbb{G}$ and $\mathbb{G}_T$ is encoded by a distinct bit-string. To do so, the elements of $\mathbb{G}$ and $\mathbb{G}_T$ are encoded to bit- strings by two random injective maps $\xi : \mathbb{Z}_p \rightarrow \Xi$ and $\xi_T : \mathbb{Z}_p \rightarrow \Xi_T$, where Ξ and Ξ_T are the sets of bit-strings such that $\Xi \cap \Xi_T = \phi$ and $|\Xi| = |\Xi_T| = p$. In the generic bilinear group model, there exist three group queries that include the multiplication query Q_G on $\mathbb{G}$, the multiplication query Q_T on $\mathbb{G}_T$ and the bilinear pairing query Q_p from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}_T$. For any $x, y \in \mathbb{Z}_p^*$, three group queries respectively have the following properties.

- $Q_G(\xi(x), \xi(y)) \rightarrow \xi(x + y \bmod p)$.
- $Q_T(\xi_T(x), \xi_T(y)) \rightarrow \xi_T(x + y \bmod p)$.
- $Q_p(\xi(x), \xi(y)) \rightarrow \xi_T(xy \bmod p)$.

It is worth mentioning that $\xi(1) = g$ and $\hat{e}(g, g) = \xi_T(1) = g_T$ are respectively the generators of $\mathbb{G}$ and $\mathbb{G}_T$.

C. ENTROPY

The meaning of entropy in statistics is the measure of uncertainty. Let Y and Z be two finite random variables while

Pr[Y=y] and Pr[Z=z] are the associated probability distributions. A min-entropy is used to represent the worst-case predictability of a random variable. Two kinds of min- entropies are defined as below:

1) $H_\infty(Y) = -\log_2(\max_y \Pr[Y = y])$ represents the min-entropy of Y.
2) $\tilde{H}_\infty(Y|Z) = -\log_2(E_{z\leftarrow Z}[\max_y \Pr[Y = y|Z = z]])$ represents the average conditional min-entropy of Y under Z.

In 2008, Dodis *et al.* [41] presented the min-entropy of a finite random variable Y with the leakage information in Lemma 1. Galindo and Vivek [17] proved the probability distribution of a polynomial under the leakage information in Lemma 2.

Lemma 1. Let $f : Y \rightarrow \{0, 1\}^\lambda$ be a leakage function on a random variable Y and $f(Y)$ denotes the leakage information. We have $\tilde{H}_\infty(Y|f(Y)) \geqq H_\infty(Y) - \lambda$.

Lemma 2. Let $F \in \mathbb{Z}_p[Y_1, Y_2, \ldots, Y_n]$ represent a non-zero polynomial of degree at most d. P_i (for $i = 1, 2, \ldots, n$) are the associated probability distributions on Y_i such that $0 \leqq \lambda \leqq \log p$ and $H_\infty(P_i) \geqq \log p - \lambda$. If all $y_i \xleftarrow{P_i} \mathbb{Z}_p$ are mutually independent, we have the probability $\Pr[F(y_1, y_2, \ldots, y_n) = 0] \leqq \frac{d}{p}2^\lambda$. In addition, $\Pr[F(y_1, y_2, \ldots, y_n) = 0]$ is negligible if $\lambda < \log p - \omega(\log\log p)$.

III. SYNTAX AND SECURITY NOTIONS

In a LR-CBE or LR-CB-KE scheme, there are two roles that include users (senders and receivers) and a certificate authority (CA). The user first chooses a private key and produces the corresponding public key. The user then sends her/his identity and public key to the CA. By the user's identity and public key, the CA uses a system secret key to compute and send the associated certificate to the user. It is worth mentioning that a user must employ both her/his private key and certificate to decrypt a ciphertext. A LR-CBE scheme consists of five algorithms, namely, Setup, User key generation, Certificate generation, Encrypt and Decrypt algorithms. As the syntax of the LR-CBE scheme, in a LR-CB-KE scheme, the Encrypt and Decrypt algorithms may be remained or replaced with the Encapsulation and Decapsulation algorithms, respectively.

The design principle of the proposed LR-CB-KE scheme resistant to continual leakage is to employ the key refreshing technique [17], [22], [36] to update the CA's system secret key, and each user's private key and certificate after each invocation. In the key refreshing technique, the CA partitions the system secret key into two parts and updates the two parts after each certificate generation procedure. Similarly, each user respectively partitions her/his private key and certificate into two parts and updates them after each decryption procedure. For precisely describing the key refreshing procedure of Certificate generation and Decrypt algorithms, the initial system secret key SSK is divided into two parts $(SSK_{0,1}, SSK_{0,2})$. In addition, a user's initial private key USK and certificate CSK are divided into two parts $(USK_{0,1}, USK_{0,2})$ and $(CSK_{0,1}, CSK_{0,2})$, respectively. Moreover, in order to model the adversary's leakage capacity for i-th Certificate generation invocation, the current system secret key $(SSK_{i,1}, SSK_{i,2})$ must be updated according to the previous system secret key $(SSK_{i-1,1}, SSK_{i-1,2})$. For the same reason, in order to model the adversary's leakage capacity for j-th Decrypt invocation, the user's current private key $(USK_{j,1}, USK_{j,2})$ and certificate $(CSK_{j,1}, CSK_{j,2})$ are computed from the previous private key $(USK_{j-1,1}, USK_{j-1,2})$ and certificate $(CSK_{j-1,1}, CSK_{j-1,2})$, respectively.

In the following, the syntax and security notions of LR-CB-KE schemes resistant to continual key leakage are defined. It is worth mentioning that the presented syntax and security notions of LR-CB-KE schemes resistant to continual key leakage have two differences with the aforementioned LR-CBE schemes [33], [34] and is similar to the aforementioned LR-CBE scheme [35]. Two differences are the key refreshing procedures of the system secret key SSK and a user's private key USK and certificate CSK.

A. SYNTAX OF LR-CB-KE SCHEME

Here, we define the syntax of LR-CB-KE schemes resistant to continual key leakage.

Definition 1: A LR-CB-KE scheme consists of five algorithms as below:

– *Setup:* The CA takes a security parameter as input and performs this algorithm to produce the initial system secret key $SSK = (SSK_{0,1}, SSK_{0,2})$ and set public parameters PP. The CA chooses a symmetric encryption function $E()$ and the associated symmetric decryption function $D()$. Finally, the CA keeps $(SSK_{0,1}, SSK_{0,2})$ in secret and publishes PP.
– *User key generation:* A user performs this algorithm to produce her/his private key $USK = (USK_{0,1}, USK_{0,2})$ and the first partial public key UPK.
– *Certificate generation:* For the i-th round of *Certificate generation* algorithm, the CA takes as input a user's identity and partial public key UPK to perform this algorithm to produce the user's certificate CSK and the corresponding partial public key CPK. Meanwhile, the CA must update the current system secret key $(SSK_{i,1}, SSK_{i,2})$ computed from $(SSK_{i-1,1}, SSK_{i-1,2})$. In addition, CSK and CPK are sent to the user. Afterwards, the user sets the initial certificate $CSK = (CSK_{0,1}, CSK_{0,2})$ and the complete public key (UPK, CPK).
– *Encrypt (Encapsulation):* By taking as input a plaintext m and a receiver's identity ID and public key (UPK, CPK), a sender performs this algorithm to produce an encryption key EK, a public value C and $CT = E_{EK}(m)$, where $E()$ is the employed symmetric encryption function. Finally, this algorithm returns the ciphertext (C, CT).
– *Decrypt (Decapsulation):* For the j-th *Decrypt* round of a user with private key $(USK_{j-1,1}, USK_{j-1,2})$ and certificate $(CSK_{j-1,1}, CSK_{j-1,2})$), upon receiving the ciphertext (C, CT), the user adopts the ciphertext C to

obtain the symmetric encryption key EK and then decrypts CT to obtain the plain-text m. Meanwhile, the user updates the current private key ($USK_{j,1}$, $USK_{j,2}$) and certificate ($CSK_{j,1}$,$CSK_{j,2}$) computed from ($USK_{j-1,1}$, $USK_{j-1,2}$) and ($CSK_{j-1,1}$, $CSK_{j-1,2}$), respectively.

B. SECURITY NOTIONS OF LR-CB-KE SCHEME

In the presence of the continual key leakage, adversaries can gain partial information of secret values involved in computation operations of a LR-CB-KE scheme, namely, adversaries are allowed to continuously gain partial information of the CA's system secret key and random values involved in the *Certificate generation* phase, and the user's private key, certificate and random values involved in the *Decrypt* phase. For representing the leaked partial information gained by adversaries, two leakage functions $f_{CG,i}$ and $h_{CG,i}$ are defined to model the abilities of adversaries for the i-th round of *Certificate generation* algorithm. Meanwhile, two leakage functions $f_{D,j}$ and $h_{D,j}$ are defined to model the abilities of adversaries for the j-th Decrypt round of a user. The output bit-length of each leakage function is bounded to λ, namely, $|f_{CG,i}|$, $|h_{CG,i}|$, $|f_{D,j}|$, $|h_{D,j}| \leq \lambda$, where λ denotes the leakage parameter. The outputs of four leakage functions are defined as below:

- $\Lambda f_{CG,i} = f_{CG,i}$ ($SSK_{i-1,1}$, *random values*).
- $\Lambda h_{CG,i} = h_{CG,i}$ ($SSK_{i-1,2}$, *random values*).
- $\Lambda f_{D,j} = f_{D,j}(USK_{j-1,1}, CSK_{j-1,1}$, *random values*).
- $\Lambda h_{D,j} = h_{D,j}(USK_{j-1,2}, CSK_{j-1,2}$, *random values*).

It is worth mentioning that the EK denotes the symmetric encryption key EK used in the j-th *Decrypt* round of the user.

In the following, we present a new continuous leakage model (adversary model) of LR-CB-KE schemes. The continuous leakage model is extended from the models of LR-CBE schemes defined in [33]–[35]. In this model, adversaries are allowed to continuously gain partial information of the CA's system secret key involved in the certificate generation phase, the user's private key and certificate involved in the decryption phase, and random values used in both phases. In addition, the whole leakage amount is unbounded during the life time of the LR-CB-KE scheme, namely, the continuous leakage model possesses overall unbounded leakage property. The continuous leakage model consists of two kinds of adversaries that include Type I adversary (uncertified entity) and Type II adversary (honest-but-curious CA).

- Type I adversary simulates the role of an uncertified entity (outsider) who may gain the private key of any entity by replacing the public key of the entity, but cannot know the entity's certificate and the CA's system secret key.
- Type II adversary simulates the role of the honest-but-curious CA who possesses the system secret key and certificates of all users. But, it is disallowed to gain the private key of any entity by replacing the public key of the entity.

Next, a security game $G_{LR-CB-KE}$ is presented to define the abilities of adversaries for the LR-CB-KE schemes in the continual leakage model.

Definition 2 ($G_{LR-CB-KE}$): The security game $G_{LR-CB-KE}$ defines the interactions between an adversary A and a challenger B in a LR-CB-KE scheme. If no probabilistic polynomial-time (PPT) adversary A (including Types I and II adversaries) with a non-negligible advantage wins the security game $G_{LR-CB-KE}$, we say that the LR-CB-KE scheme is semantically secure against chosen ciphertext attacks in the continual leakage model.

- *Setup phase:* B takes as input a security parameter τ and performs the $Setup$ algorithm to produce the CA's initial system secret key $SSK = (SSK_{0,1}, SSK_{0,2})$ and set public parameters PP. If A is of Type I adversary, B keeps $SSK = (SSK_{0,1}, SSK_{0,2})$ in secret. If A is of Type II adversary, B sends $SSK = (SSK_{0,1}, SSK_{0,2})$ to A. In addition, PP is published and sent to A.
- *Query phase:* A can issue numerous queries adaptively as follows:
  - *User key generation query* (ID): Upon receiving the request with ID, B performs the *User key generation* algorithm to produce the user's initial private key $USK = (USK_{0,1}, USK_{0,2})$ and the first partial public key UPK.
  - *Private key query* (ID): Upon receiving the request with ID, B sends the user's initial private key $USK = (USK_{0,1}, USK_{0,2})$ to A. It is worth mentioning that this query is disallowed if the *Public key replace query* (ID) has ever been issued.
  - *Certificate generation query* (ID, UPK): Upon receiving the request with identity ID and the first partial public key UPK, B returns the user's initial certificate $CSK = (CSK_{0,1}, CSK_{0,2})$ and the second partial public key CPK.
  - *Certificate generation leak query* (i, $f_{CG,i}$, $h_{CG,i}$): Upon receiving the request with the i-th *Certificate generation query* and two leakage functions $f_{CG,i}$, $h_{CG,i}$, B responds the leakage information ($\Lambda f_{CG,i}$, $\Lambda h_{CG,i}$) to A, where ($\Lambda f_{CG,i}$, $\Lambda h_{CG,i}$) denotes the leakage information of the CA's current system secret key $SSK = (SSK_{i-1,1}, SSK_{i-1,2})$.
  - *Public key retrieve query* (ID): Upon receiving the request with ID, B responds the associated public key (UPK, CPK).
  - *Public key replace query* (ID, (UPK', CPK')): Upon receiving the request with ID and the new public key $(UPK,' CPK')$, B keeps track of the replacement.
  - *Decrypt (Decapsulation) query* (ID, (C, CT)): Upon receiving the request with identity ID and the ciphertext (C, CT), B uses the associated private key USK and certificate CSK to produce the encryption key EK and decrypt the message $m = D_{EK}(CT)$. B sends m and EK to A.
  - *Decrypt (Decapsulation) leak query* (ID, j, $f_{D,j}, h_{D,j}$): Upon receiving the request with the j-th *Decrypt query* of identity ID and two leakage functions $f_{D,j}, h_{D,j}$, B responds the leakage information $(\Lambda f_{D,j}, \Lambda h_{D,j})$ to A. If A

is of Type I adversary, $(\Lambda f_{D,j}, \Lambda h_{D,j})$ denotes partial information of the user's current certificate $CSK = (CSK_{j-1,1}, CSK_{j-1,2})$. If A is of Type II adversary, $(\Lambda f_{D,j}, \Lambda h_{D,j})$ denotes partial information of the user's current private key $USK = (USK_{j-1,1}, USK_{j-1,2})$.

- *Challenge phase:* A sends a plain-text pair (m_0^*, m_1^*) and a target identity ID^* to B. B selects a random unbiased bit $b \in \{0, 1\}$ and performs the *Encrypt* algorithm with $(PP, ID^*, m_b^*, (UPK^*, CPK^*))$ to generate C^* and EK^*, where (UPK^*, CPK^*) is the associated public key of the user with identity ID^*. Finally, B generates $CT^* = E_{EK^*}(m_b^*)$, where $E()$ is the employed symmetric encryption function. Finally, B returns the ciphertext (C^*, CT^*) to A. In addition, two conditions must be satisfied.
  1) If A is of Type I adversary, the *Certificate generation query* (ID^*, UPK^*) has never been issued.
  2) If A is of Type II adversary, both the *Private key query* (ID^*) and *Public key replace query* (ID^*) have never been issued.
- *Guess phase:* A outputs a bit $b' \in \{0, 1\}$ and wins the security game $G_{LR-CB-KE}$ if $b' = b$. The advantage of A is defined by $\mathrm{Adv}_A(\tau) = |\Pr[b' = b] - 1/2|$.

IV. THE PROPOSED LR-CB-KE SCHEME

Here, we propose the first LR-CB-KE scheme with overall unbounded leakage property. As the syntax of Definition 1, the LR-CB-KE scheme consists of five algorithms as below:

- ***Setup***: The CA takes a security parameter τ as input, and selects two groups $\mathbb{G}$, $\mathbb{G}_T$ of a large prime order p and the associated bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ as presented in Section 2.1. The CA then chooses a generator g of $\mathbb{G}$. The CA also chooses a symmetric encryption function $E()$ and the associated symmetric decryption function $D()$. In addition, the CA sets the public parameters PP and the initial system secret key $(SSK_{0,1}, SSK_{0,2})$ by performing the following steps:
  1) Pick a random integer $s \in \mathbb{Z}_p^*$, and compute the system secret key $SSK = g^s$ and the associated system public key $SPK = \hat{e}(g^s, g)$.
  2) Pick a random integer $a \in \mathbb{Z}_p^*$ and set the initial system secret key $(SSK_{0,1}, SSK_{0,2}) = (g^a, SSK \cdot g^{-a})$. It is obvious that $SK = SSK_{0,1} \cdot SSK_{0,2}$.
  3) Pick two random integers $\mu, \nu \in \mathbb{Z}_p^*$, and compute $U = g^\mu$ and $V = g^\nu$.
  4) Set and publish $PP = (\mathbb{G}, \mathbb{G}_T, p, g, \hat{e}, SPK, U, V, E, D)$.
- ***User Key generation***: Without loss of generality, a user with identity ID randomly selects $t, b \in \mathbb{Z}_p^*$, and computes the associated private key $USK = g^t$ and the first partial public key $UPK = \hat{e}(g^t, g)$. Finally, the user sets her/his initial private key $(USK_{0,1}, USK_{0,2}) = (g^{-b}, g^{t+b})$, where $USK_{0,1} \cdot USK_{0,2} = USK$.
- ***Certificate generation***: For the i-th round of *Certificate generation* algorithm, the CA with the current system secret key $(SSK_{i-1,1}, SSK_{i-1,2})$ takes as input a user's identity ID and partial public key UPK, and produces the user's certificate CSK and second partial public key CPK by performing the following steps.
  1) Randomly choose $c, d \in \mathbb{Z}_p^*$, set the bit-string $BX = ID||UPK$ and compute $X = BX \bmod p$, where BX is viewed as the corresponding integer of the bit-string.
  2) Compute the user's second partial public key $CPK = g^d$, a temporary value $TI_{CG} = SSK_{i-1,1} \cdot (U \cdot V^X)^d$ and the user's certificate $CSK = SSK_{i-1,2} \cdot TI_{CG}$.
  3) Update the current system secret key $(SSK_{i,1} = SSK_{i-1,1} \cdot g^c, SSK_{i,2} = SSK_{i-1,2} \cdot g^{-c})$.
  4) Return the user's second partial public key CPK and certificate CSK.

  Upon receiving the second partial public key CPK and certificate CSK, the user sets her/his initial certificate as below:
  1) Randomly choose $h \in \mathbb{Z}_p^*$ and set the user's initial certificate $(CSK_{0,1}, CSK_{0,2}) = (g^h, CSK \cdot g^{-h})$.
  2) Set the user's public key (UPK, CPK).
- ***Encrypt(Encapsulation)***: Taking as input a plain-text m, and a receiver's identity and associated public key (UPK, CPK), a sender performs the following steps to produce the ciphertext (C, CT).
  1) Choose a random integer $r \in \mathbb{Z}_p^*$, set the bit-string $BX = ID||UPK$ and compute $X = BX \bmod p$, where BX is viewed as the corresponding integer of the bit-string.
  2) Compute $C = g^r$, $EK_1 = (UPK)^r = \hat{e}(g^t, g)^r$ and $EK_2 = (SPK \cdot \hat{e}(CPK, U \cdot V^X))^r$.
  3) Generate the encryption key $EK = EK_1 \oplus EK_2$ and $CT = E_{EK}(m)$.

  Finally, the sender returns the ciphertext (C, CT) to the receiver.
- ***Decrypt(Decapsulation)***: For the j-th *Decrypt* round of a user with private key $(USK_{j-1,1}, USK_{j-1,2})$ and certificate $(CSK_{j-1,1}, CSK_{j-1,2}))$, upon receiving the ciphertext (C, CT), the user adopts the ciphertext C to compute the symmetric encryption key EK which is used to decrypt CT to obtain the plain-text m by performing the following steps.
  1) Compute $EKI_1 = \hat{e}(C, USK_{j-1,1})$ and $EKI_2 = \hat{e}(C, CSK_{j-1,1})$.
  2) Compute $EK_1' = EKI_1 \cdot \hat{e}(C, USK_{j-1,2})$ and $EK_2' = EKI_2 \cdot \hat{e}(C, CSK_{j-1,2})$.
  3) Set the encryption key $EK' = EK_1' \oplus EK_2'$ and decrypt the plain-text $m = D_{EK}(CT)$.
  4) Choose a random integer $k \in \mathbb{Z}_p^*$, and update the user's current private key $(USK_{j,1} = USK_{j-1,1} \cdot g^k, USK_{j,2} = USK_{j-1,2} \cdot g^{-k})$ and current certificate $(CSK_{j,1} = CSK_{j-1,1} \cdot g^k, CSK_{j,2} = CSK_{j-1,2} \cdot g^{-k})$.

By $USK = USK_{0,1} \cdot USK_{0,2} = \cdots = USK_{j-1,1} \cdot USK_{j-1,2} = USK_{j,1} \cdot USK_{j,2}$ and $CSK = CSK_{0,1} \cdot CSK_{0,2} =$

$\cdots = CSK_{j-1,1} \cdot CSK_{j-1,2} = CSK_{j,1} \cdot CSK_{j,2}$, we show the correctness of recovering the encryption key as follows.

$$\begin{aligned} EK &= EK_1 \oplus EK_2 \\ &= (UPK)^r \oplus (SPK \cdot \hat{e}(CPK, U \cdot V^X))^r \\ &= \hat{e}(g^t, g)^r \oplus (\hat{e}(g^s, g) \cdot \hat{e}(g^d, U \cdot V^X))^r \\ &= \hat{e}(g^r, g^t) \oplus (\hat{e}(g, g^s) \cdot \hat{e}(g, U \cdot V^X)^d)^r \\ &= \hat{e}(g^r, USK) \oplus \hat{e}(g^r, SSK \cdot (U \cdot V^X)^d) \\ &= \hat{e}(C, USK) \oplus \hat{e}(C, CSK) \\ &= \hat{e}(C, USK_{0,1} \cdot USK_{0,2}) \\ &\qquad \oplus \hat{e}(C, CSK_{0,1} \cdot CSK_{0,2}) \\ &= \hat{e}(C, USK_{0,1}) \cdot \hat{e}(C, USK_{0,2}) \\ &\qquad \oplus \hat{e}(C, CSK_{0,1}) \cdot \hat{e}(C, CSK_{0,2}) \\ &= EK'. \end{aligned}$$

V. SECURITY ANALYSIS

By the security game $G_{LR-CB-KE}$ of Definition 2 mentioned in Section 3.2, there are two types of adversaries, namely, Type I adversary (uncertified entity) and Type II adversary (honest-but-curious CA). By the *Certificate generation extract leak* query, Type I adversary can gain partial information of the CA's system secret key and random values in the *Certificate generation* phase. Since Type II adversary has possessed the CA's system secret key, it does not need to issue the *Certificate generation extract leak* query. On the other hand, by the *Decrypt leak* query, both adversaries can gain partial information of the user's private key, certificate and random values in the Decrypt phase. Theorems 1 and 2, respectively, demonstrate that our LR-CB-KE scheme is semantically secure against chosen ciphertext attacks of both Types I and II adversaries in the continual leakage model.

Theorem 1: In generic bilinear group model, the proposed LR-CL-KE scheme is semantically secure against chosen ciphertext attacks of Type I adversary (A_I, uncertified entity) in the continual leakage model.

Proof: The generic bilinear group model introduced in Section 2.2 is used in the security game $G_{LR-CB-KE}$ of the proposed LR-CL-KE scheme. In this model, there are two groups $\mathbb{G}$ and $\mathbb{G}_T$ and each element of $\mathbb{G}$ and $\mathbb{G}_T$ is encoded by a distinct bit-string. In addition, three group queries are provided that include the group query Q_G on $\mathbb{G}$, the group query Q_T on $\mathbb{G}_T$ and a bilinear map query Q_p from $\mathbb{G} \times \mathbb{G}$ to $\mathbb{G}_T$. In such a case, three queries Q_G, Q_T and Q_p must be added in the security game $G_{LR-CB-KE}$ played by a challenger B and an adversary A_I. It is worth mentioning that the challenger B is responsible to encode each element of $\mathbb{G}$ and $\mathbb{G}_T$ by a distinct bit-string.

- *Initial Setup:* In the phase, B takes as input a security parameter τ and performs the $Setup$ algorithm to produce the CA's system secret key SSK and public parameters $PP = (\mathbb{G}, \mathbb{G}_T, p, g, \hat{e}, SPK = SSK \cdot g, U, V, E, D)$. In addition, several lists are constructed to record the queries issued by A_I.
  - B constructs two lists L_G and L_T to record all elements of $\mathbb{G}$ and $\mathbb{G}_T$, respectively.
    1) L_G records all elements of G with the form $(\Theta G_{m,n,r}, \Xi G_{m,n,r})$. $\Theta G_{m,n,r}$ denotes an element of G represented by a multivariate polynomial with coefficients in $\mathbb{Z}_p$ and variates in $\mathbb{G}$. $\Xi G_{m,n,r})$ denotes the corresponding encoded bit-string of $\Theta G_{m,n,r}$. The indices m, n and r, respectively, represent the type of query, the n-th query, and the r-th element. Initially, four elements $(g, \Xi G_{I,1,1})$, $(U, \Xi G_{I,1,2})$, $(V, \Xi G_{I,1,3})$ and $(SSK, \Xi G_{I,1,4})$ are added in L_G.
    2) L_T records all elements of $\mathbb{G}_T$ with the form $(\Theta T_{m,n,r}, \Xi T_{m,n,r})$. $(\Theta T_{m,n,r}$ denotes an element of $\mathbb{G}$ represented by a multivariate polynomial with coefficients in $\mathbb{Z}_p$ and variates in $\mathbb{G}/\mathbb{G}_T$. $\Xi T_{m,n,r}$ denotes the corresponding encoded bit-string of $(\Theta T_{m,n,r}$. Three indices m, n and r are the same as those in L_G. Initially, an element $(SPK, \Xi T_{I,1,1})$ is added in L_T.

    Upon receiving the related queries issued by A_I in the *Query* phase described later, B employs two rules to maintain L_G and L_T as below.
    1) When B receives the transformation request of a multivariate polynomial $\Theta G_{m,n,r}/\Theta T_{m,n,r}$, B returns the associated bit-string $\Xi G_{m,n,r}/\Xi T_{m,n,r}$ if $\Theta G_{m,n,r}/\Theta T_{m,n,r}$ has been recorded in L_G/L_T. Otherwise, B chooses a distinct and random bit-string $\Xi G_{m,n,r}/\Xi T_{m,n,r}$ and records $(\Theta G_{m,n,r}, \Xi G_{m,n,r})/(\Theta T_{m,n,r}, \Xi T_{m,n,r})$ in L_G/L_T. Also, B returns the bit-string $\Xi G_{m,n,r}/\Xi T_{m,n,r}$.
    2) When B receives the transformation request of a bit-string $\Xi G_{m,n,r}/\Xi T_{m,n,r}$, B responds the corresponding multivariate polynomial $\Theta G_{m,n,r}/\Theta T_{m,n,r}$ in L_G/L_T.
  - B constructs a list L_K of tuples with form $(ID_i$, $replace$, ΘUSK_i, ΘUPK_i, ΘCSK_i, $\Theta CKP_i)$ to record the private key USK_i, certificate CSK_i and the public key (UPK_i, CPK_i) of the user U_i with identity ID_i, where ID_i is in $\mathbb{Z}_p^*$, and ΘUSK_i, ΘUPK_i, ΘCSK_i, ΘCKP_i are multivariate polynomials recorded in L_G or L_T. The field of the replace denotes the status of public key replacement and is initially set to *"false"*. Whenever A_I issues the Public key replace query (ID_i), B sets the field of the replace for ID_i to be *"true"*.
  - Finally, B sends the corresponding bit-strings of several public parameters $\{g, U, V, SPK\}$ to A_I.
- *Query:* In the phase, A_I may adaptively issue the following queries at most q times. Note that the challenger B is responsible to encode the corresponding relations between multivariate polynomials and bit-strings in L_G or L_T. Therefore, when B received a bit-string that

does not exist in L_G or L_T, B responses the failure of the game.

- *Group query* $Q_G(\Xi G_{Q,i,1}, \Xi G_{Q,i,2}, OP)$: Upon receiving the i-th request with two bit-strings $(\Xi G_{Q,i,1}, \Xi G_{Q,i,2})$ in L_G and an OP (multiplication/division) operation, B performs the following steps to send the resulting bit-string $\Xi G_{Q,i,3}$.
  1) Transform $\Xi G_{Q,i,1}$ and $\Xi G_{Q,i,2}$ respectively to obtain the corresponding multivariate polynomials $\Theta G_{Q,i,1}$ and $\Theta G_{Q,i,2}$ in L_G.
  2) If OP is the multiplication operation, compute the polynomial $\Theta G_{Q,i,3} = \Theta G_{Q,i,1} + \Theta G_{Q,i,2}$. If OP is the division operation, compute $\Theta G_{Q,i,3} = \Theta G_{Q,i,1}$ - $\Theta G_{Q,i,2}$.
  3) Transform the resulting polynomial $\Theta G_{Q,i,3}$ and return the corresponding bit-string $\Xi G_{Q,i,3}$.
- *Group query* $Q_T(\Xi T_{Q,i,1}, \Xi T_{Q,i,2}, OP)$: Upon receiving the i-th request with two bit-strings $(\Xi T_{Q,i,1}, \Xi T_{Q,i,2})$ in L_T and an OP (multiplication or division) operation, B performs similar steps mentioned in the *Group query* Q_G and return the corresponding bit-string $\Xi T_{Q,i,3}$.
- *Pairing query* $Q_P(\Xi G_{P,i,1}, \Xi G_{P,i,2})$: Upon receiving the i-th request with two bit-strings $(\Xi G_{P,i,1}, \Xi G_{P,i,2})$ in L_G, B performs the following steps:
  1) Transform $\Xi G_{P,i,1}$ and $\Xi G_{P,i,2}$ respectively to obtain the corresponding polynomials $\Theta G_{P,i,1}$ and $\Theta G_{P,i,2}$.
  2) Compute the polynomial $\Theta T_{P,i,1} = \Theta G_{P,i,1} \cdot \Theta G_{P,i,2}$.
  3) Transform $\Theta T_{P,i,1}$ in L_T and return the corresponding bit-string $\Xi T_{P,i,1}$.
- *Private key query* (ID_i): Upon receiving this request with identity ID_i, if ID_i has been recorded in L_K and the replace field is "false," B first obtains the polynomial ΘUSK_i of the user's private key and transforms it to return the corresponding bit-string ΞUSK_i to A_I. If ID_i has been not recorded in L_K, B issues the *User key generation query* (ID_i) to return ΞUSK_i to A_I.
- *Certificate generation query* (ID_i, UPK_i): Upon receiving the i-th request with identity ID_i and the first partial public key UPK_i, B performs the following steps:
  1) Choose a new variate $TG_{CG,i,1}$ in $\mathbb{G}$ to represent the certificate CPK_i, and set the polynomial $\Theta CPK_i = TG_{CG,i,1}$.
  2) Transform UPK_i to get the corresponding bit-string ΞUPK_i and set the bit-string $X = ID||\Xi UPK_i$.
  3) Choose a new variate $TG_{CG,i,2}$ in $\mathbb{G}$ and compute the second partial public key $\Theta CSK_i = SSK + TG_{CG,i,2} \cdot (U + X \cdot V)$ while updating (ID_i, $false$, ΘUSK_i, ΘUPK_i, ΘCPK_i, ΘCSK_i) in L_K.
  4) Record ΘCPK_i and ΘCSK_i in L_G to return the corresponding bit-strings ΞCPK_i and ΞCSK_i to A_I.
- *Certificate generation leak query* $(i, f_{CG,i}, h_{CG,i})$: For the i-th *Certificate generation query*, A_I can issue the *Certificate generation leak query* only once by providing two leakage functions $f_{CG,i}$ and $h_{CG,i}$ such that $|f_{CG,i}| \leq \lambda$ and $|h_{CG,i}| \leq \lambda$. Upon receiving this request, B computes and sends the leakage information $(\Lambda f_{CG,i}, \Lambda hCG, i)$ to A_I, where $\Lambda f_{CG,i} = f_{CG,i}(SSK_{i-1,1}, c, d)$ and $\Lambda h_{CG,i} = h_{CG,i}(SSK_{i-1,2}, c, TI_{CG})$.
- *Public key retrieve query (ID_i):* Upon receiving the request with identity ID_i, B searches the list L_K to gain the user's public key (UPK_i, CPK_i) and send the corresponding bit-strings $(\Xi UPK_i, \Xi CPK_i)$ to A_I.
- *Public key replace query* $(ID_i, (\Xi UPK_i,' \Xi CPK_i'))$*:* Upon receiving the request with identity ID_i and her/his new public key $(\Xi UPK_i,' \Xi CPK_i')$, B first transforms $(\Xi UPK_i,' \Xi CPK_i')$ to obtain the corresponding polynomials $(\Theta UPK_i,' \Theta CPK_i')$ and updates $(ID_i, true, null, \Theta UPK_i, null, \Theta CKP_i)$ in L_K.
- *Decrypt (Decapsulation) query* $(ID, (C, CT))$: Upon receiving the request with identity ID and the ciphertext (C, CT), B performs the following steps to gain the encryption key EK and the plain-text m:
  1) B uses ID to find the user's private key $(\Theta USK, \Theta CSK)$ in L_K.
  2) B transforms the ciphertext C to the polynomial ΘC in L_G and computes two polynomials $\Theta EK_1 = \Theta USK \cdot \Theta C$ and $\Theta EK_2 = \Theta CSK \cdot \Theta C$. Moreover, B transforms ΘEK_1 and ΘEK_2 to bit-strings ΞEK_1 and ΞEK_2, respectively. Hence, B can gain the encryption key $EK = \Xi EK_1 \oplus \Xi EK_2$. Finally, B returns the encryption key EK and the plain-text $m = D_{EK}(CT)$ to A_I.
- *Decrypt (Decapsulation) leak query* $(ID, j, f_{D,j}, h_{D,j})$: Upon receiving the request with the j-th *Decrypt query* of identity ID and two leakage functions $f_{D,j}, h_{D,j}$, B computes the leakage information $(\Lambda f_{D,j}, \Lambda h_{D,j})$ and returns it to A_I, where $\Lambda f_{D,j} = f_{D,j}(USK_{j-1,1}, CSK_{j-1,1}, k)$ and $\Lambda h_{D,j} = h_{D,j}(USK_{j-1,2}, CSK_{j-1,2}, k, EKI_1, EKI_2, EK)$. It is worth mentioning that for the j-th *Decrypt query*, A_I can issue the *Decrypt leak query* only once.

– *Challenge phase :* The adversary A_I chooses and sends a target identity ID^* with public key (UPK^*, CPK^*) and a plain-text pair (m_0^*, m_1^*) to the challenger B. The *Certificate generation query* (ID^*, UPK^*) is disallowed to be issued in the *Query* phase. B performs the following steps.
  1) B uses ID^* to find the public key (UPK^*, CPK^*) in L_K. If UPK^* is not recorded in L_K, B issues the *User key generation query* (ID^*). Moreover, if CPK^* is not recorded in L_K, B also issues the *Certificate generation query* (ID^*, UPK^*). In any case, B can gain the public key (UPK^*, CPK^*) of ID^* in L_K.

2) B randomly chooses a new variate $TG_{Ch,i,1}$ in $\mathbb{G}$ and sets $\Theta C^* = TG_{Ch,i,1}$. B then gains the bit-string ΞC^* by transforming ΘC^* in L_G.
3) B transforms UPK^* to gain the bit-string ΞUPK^* and sets $X = ID^*||\Xi UPK^*$.
4) B computes $\Theta EK_1 = TG_{Ch,i,1} \cdot \Theta UPK^*$ and $\Theta EK_2 = TG_{Ch,i,1} \cdot (SPK + CPK^* \cdot (U + X \cdot V))$.
5) B transforms ΘEK_1^* and ΘEK_2^* to the bit-strings ΞEK_1^* and ΞEK_2^*, respectively. B then computes the encryption key $EK^* = \Xi EK_1^* \oplus \Xi EK_2^*$.
6) B chooses a random bit $b \in \{0, 1\}$ and computes $CT^* = E_{EK^*}(m_b^*)$. Finally, B sends (C^*, CT^*) to A_I.

– *Guess phase*. A_I outputs a bit $b' \in \{0, 1\}$ and wins the security game $G_{LR-CB-KE}$ if $b' = b$.

In the following, the advantage that A_I wins the security game $G_{LR-CB-KE}$ is evaluated. Let us first evaluate the numbers of elements and the maximal degrees of polynomials in L_G and L_T.

1) In the *Query* phase, A_I may issue queries at most q times and six kinds of queries might add elements in L_G and L_T.
   - Initially, 4 elements and 1 element are respectively stored in L_G and L_T.
   - For each Q_G, Q_T and Q_P query, at most 3 elements are added in L_G or L_T.
   - For each *User key generation query*, at most 1 element is respectively added in L_G and L_T.
   - For each *Certificate generation query*, at most 3 elements are added in L_G.
   - For each *Decrypt query*, at most 3 and 2 elements are respectively added in L_G and L_T. Let q_O denote the total number of Q_G, Q_T and Q_P queries issued by A_I and B. Let q_{UKG}, q_{CG} and q_D, respectively, denote the numbers of issuing *User key generation query*, *Certificate generation query* and *Decrypt query*. Let $|L_G|$ and $|L_T|$, respectively, denote the numbers of elements in L_G and L_T. Thus, we have $|L_G| + |L_T| \leqq 5 + 3q_O + 2q_{UKG} + 3q_{CG} + 5q_D \leqq 5q$.
2) The maximal degree of multivariate polynomials in L_G is at most 2 and discussed as below.
   - ΘUSK, ΘCPK and all new variates, have degree 1.
   - The private key ΘCSK has degree 2.
   - In the group query Q_G, because of $\Theta G_{Q,i,3} = \Theta G_{Q,i,1} + \Theta G_{Q,i,2}$, the degree of polynomial $\Theta G_{Q,i,3}$ is the maximal degree of $\Theta G_{Q,i,1}$ or $\Theta G_{Q,i,2}$.
3) The maximal degree of multivariate polynomials in L_T is at most 4 and discussed as below.
   - All new variates have degree 1.
   - The public key ΘSPK has degree 2.
   - In the group query Q_P, the resulting polynomial in L_T has degree at most 4 because it is computed by two polynomials of degree 2 in L_G.
   - In the *User key generation query*, ΘUPK has degree 2.
   - In the *Decrypt query*, $\Theta EK_1 = \Theta USK \cdot \Theta C$ has degree 3, and $\Theta EK_2 = \Theta CSK \cdot \Theta C$ has degree 4.
   - In the group query Q_T, because of $\Theta T_{Q,i,3} = \Theta T_{Q,i,1} + \Theta T_{Q,i,2}$, the degree of $\Theta T_{Q,i,3}$ is the maximal degree of $\Theta T_{Q,i,1}$ or $\Theta T_{Q,i,2}$.

   For judging the collision of two polynomials in L_G and L_T, each variable in L_G and L_T must be assigned a value. We say that the collision happens if resulting values of two polynomials with inputting these variable values are equal. Without loss of generality, let n be the total number of variables in L_G and L_T. B selects a random value in $\mathbb{Z}_p^*$ for each variable in L_G and L_T, denoted by $v_1, v_2, \ldots, v_n$. It is said that A_I wins the security game $G_{LR-CB-KE}$ if one of the following two cases occurs:
   – ***Case 1***: A_I can find a collision element in L_G or L_T. Namely, A_I finds two polynomials ΘG_i and ΘG_j in L_G such that the equality $\Theta G_i(v_1, v_2, \ldots, v_n) = \Theta G_j(v_1, v_2, \ldots, v_n)$, or two polynomials ΘT_i and ΘT_j in L_T such that the equality $\Theta T_i(v_1, v_2, \ldots, v_n) = \Theta T_j(v_1, v_2, \ldots, v_n)$.
   – ***Case 2***: A_I can output a correct bit $b' = b$ in the *Guess* query.

In the following, let's evaluate the advantage of A_I in the security game $G_{LR-CB-KE}$. We first discuss A_I's success probability in the security game $G_{LR-CB-KE}$ without issuing *Certificate generation leak query* and *Decrypt leak query*. Afterward, the success probability of the situation with issuing *Certificate generation leak query* and *Decrypt leak query* is measured.

- ***Without issuing Certificate generational leak query and Decrypt leak query***: Assume that A_I is disallowed to issue the *Certificate generation leak query* or *Decrypt leak query*. The success probability of A_I wins the security game $G_{LR-CB-KE}$ consists of two cases as below.
  – ***Case 1***: This case denotes the success probability that A_I can find a collision in L_G or L_T. Let ΘG_i and ΘG_j be two distinct polynomials in L_G. The success probability of finding a collision in L_G is equal to the probability that $\Theta G_C = \Theta G_i - \Theta G_j$ is a zero polynomial, i.e. $\Theta G_C(v_1, v_2, \ldots, v_n) = 0$. By Lemma 2, since the maximal degree of the polynomials in L_G is at most 2 and the leakage bit-length $\lambda = 0$, the probability of $\Theta G_C(v_1, v_2, \ldots, v_n) = 0$ is at most $2/p$. In such a case, the success probability of finding a collision in L_G is at most $(2/p)\binom{|L_G|}{2}$ because there are $\binom{|L_G|}{2}$ distinct pairs $(\Theta G_i, \Theta G_j)$ in L_G. By the similar way, the success probability of finding a collision in L_T is at most $(4/p)\binom{|L_T|}{2}$ because the maximal degree of polynomials in L_T is at most 4. In addition, we have $|L_G| + |L_T| \leqq 5 + 3q_O + 2q_{UKG} + 3q_{CG} + 5q_D \leqq 5q$. Therefore, the success probability that Case 1 occurs, denoted by Pr[Case 1], satisfies the inequality

$$\begin{aligned} Pr[Case1] &\leqq (2/p)\binom{|L_G|}{2} + (4/p)\binom{|L_T|}{2} \\ &\leqq (4/p)(|L_G| + |L_T|)^2 \\ &\leqq 100q^2/p. \end{aligned}$$

  – ***Case 2***: If Case 1 does not occur, A_I gain no useful information about guessing a correct bit b in the *Guess* phase. Hence, the success probability of outputting the correct bit $b' = b$ is 1/2. Therefore, the success probability that *Case* 2 occurs, denoted by Pr[Case 2], satisfies the inequality Pr[Case 2]≦1/2.

By ***Cases*** 1 and 2, the success probability that A_I wins $G_{LR-CB-KE}$ without issuing *Certificate generation leak query* and *Decrypt leak query*, denoted by $\Pr_{A-I-NL}$, satisfies the inequality

$$\begin{aligned}\Pr_{A-I-NL} &\leqq Pr[Case\ 1] + Pr[Case\ 2] \\ &\leqq 100q^2/p + (1/2).\end{aligned}$$

Therefore, the advantage of A_I wins $G_{LR-CB-KE}$ without issuing *Certificate generation leak query* and *Decrypt leak query*, denoted by Adv_{A-I-NL}, satisfies the inequality

$$\begin{aligned}Adv_{A-I-NL} &\leqq |100q^2/p + (1/2) - (1/2)| \\ &= 100q^2/p = O(q^2/p).\end{aligned}$$

If $q = poly(\log p)$, Adv_{A-I-NL} is negligible.

- ***With issuing Certificate generation leak query and Decrypt leak query***: Here, A_I is allowed to issue the *Certificate generation leak query* and *Decrypt leak query*. In the *Certificate generation leak query*, two leakage functions $f_{CG,i}$ and $h_{CG,i}$ are used in the i-th *Certificate generation leak query* while $\Lambda f_{CG,i}$ and $\Lambda h_{CG,i}$ denote the corresponding leakage outputs, respectively. In the *Decrypt leak query*, two leakage functions $f_{D,j}$ and $h_{D,j}$ are used in the j-th *Decrypt leak query* while $\Lambda f_{D,j}$ and $\Lambda h_{D,j}$ denote the corresponding outputs, respectively. The output bit-lengths of four leakage functions are bounded by λ. The leaked information about $\Lambda f_{CG,i} = f_{CG,i}(SSK_{i-1,1}, c, d)$ and $\Lambda h_{CG,i} = h_{CG,i}(SSK_{i-1,2}, c, TI_{CG})$ is discussed as follows. It is worth mentioning that for the same identity, A_I is allowed to issue the *Certificate generation leak query* only once.

■ c: c is randomly chosen for each user's identity and used to update the CA's current system secret key. At most 2λ bits of c is helpless to gain the current system secret key SSK for A_I.

■ $(SSK_{i-1,1}, SSK_{i-1,2})$: By the multiplicative blinding technique [17], [22], [36], the CA's system secret key SSK satisfies the equality $SSK = SSK_{i-1,1} \cdot SSK_{i-1,2} = SSK_{i,1} \cdot SSK_{i,2}$. Note that the leaked information of both $SSK_{i-1,1}$ and $SSK_{i-1,2}$ is independent of that of both $SSK_{i,1}$ and $SSK_{i,2}$. Hence, at most λ bits of $SSK_{i-1,1}$ and $SSK_{i-1,2}$ are leaked to A_I.

■ d: d is randomly chosen for each user's identity and used to produce the user's certificate CSK. Thus, A_I can leak at most λ bits of CSK.

■ TI_{CG}: The temporary value TI_{CG} is used to compute the certificate CSK. Also, A_I can leak at most λ bits of CSK.

The leaked information about $f_{D,j}(USK_{j-1,1}, CSK_{j-1,1}, k)$ and $h_{D,j}(USK_{j-1,2}, CSK_{j-1,2}, k, EKI_1, EKI_2, EK)$ is discussed as below:

■ k: k is randomly chosen for each *Decrypt query* and used to update the user's private key USK and certificate CSK. At most 2λ bits of k is helpless to gain the complete private key USK and certificate CSK for A_I.

■ $(USK_{j-1,1}, USK_{j-1,2})$: Since A_I may issue the *Private key query* with identity ID^*, it possesses the user's complete private key USK. Hence, the leakage information is useless to A_I.

■ $(CSK_{j-1,1}, CSK_{j-1,2})$: The user's certificate CSK satisfies the equality $CSK = CSK_{j-1,1} \cdot CSK_{j-1,2} = CSK_{j,1} \cdot CSK_{j,2}$. Note that the leaked information of both $CSK_{j-1,1}$ and $CSK_{j-1,2}$ is independent of that of both $CSK_{j,1}$ and $CSK_{j,2}$. Hence, at most λ bits of $CSK_{j-1,1}$ and $CSK_{j-1,2}$ are leaked to A_I, namely, at most λ bits of CSK are leaked to A_I.

■ (EKI_1, EKI_2, EK): Since these keys are randomly chosen for each encryption. Hence, at most λ bits about the encryption key EK are leaked to A_I.

Now, let us discuss the success probability that A_I wins the game $G_{LR-CB-KE}$ with issuing the *Certificate generation leak query* and *Decrypt leak query*, denoted by $\Pr_{A-I}$. Since A_I can replace the public key by the *Public key replace query*, it may know the target user's private key USK completely. The useful information of outputting a correct bit b' is determined by the leakage information about the target user's certificate CSK and the CA's system secret key SSK. For convenience, three events of $\Pr_{A-I}$ are defined as follows.

1) The event $ECSK$ denotes that A_I may gain the user's certificate key CSK completely from the leakage information $\Lambda f_{D,j}$ and $\Lambda h_{D,j}$. In addition, denotes the complement event of $ECSK$.
2) The event $ESSK$ denotes that A_I may gain the CA's system secret key SSK completely from the leakage information $\Lambda f_{CG,i}$ and $\Lambda h_{CG,i}$. In addition, denotes the complement event of $ESSK$.
3) The event EB denotes that A_I may output a correct b'.

The success probability $\Pr_{A-I}$ with issuing *Certificate generation leak query* and *Decrypt leak query* satisfies the inequality

$$\begin{aligned}\Pr_{A-I} &= \Pr[EB] \\ &= \Pr[EB \wedge (ECSK \vee ESSK)]\end{aligned}$$

$$+\Pr[EB \wedge (\overline{ECSK} \wedge \overline{ESSK})]$$
$$\leqq \Pr[ECSK \vee ESSK]$$
$$+\Pr[EB \wedge (\overline{ECSK} \wedge \overline{ESSK})].$$

Under the condition $\overline{ECSK} \wedge \overline{ESSK}$, the helpful information to output a correct bit is at most λ bits of the encryption key EK. Since A_I have 1/2 probability to guess the correct bit, $\Pr[EB|(\overline{ECSK} \wedge \overline{ESSK})]$ is still 1/2 on average. Thus, we have

$$\Pr_{A-I} \leqq \Pr[ECSK \vee ESSK] + 1/2.$$

Hence, the advantage of A_I with issuing *Certificate generation leak query* and *Decrypt leak query*, denoted by Adv_{A-I}, satisfies the inequality

$$Adv_{A-I} \leqq |\Pr_{A-I} - 1/2| = \Pr[ECSK \vee ESSK].$$

In the situation without issuing *Certificate generation leak query* and *Decrypt leak query*, A_I's advantage has the inequality $Adv_{A-NL} \leqq 100q^2/p = O(q^2/p)$. Since A_I can learn at most 2λ bits of the user's certificate CSK, the CA's system secret key SSK or the encryption key EK, we have

$$Adv_{A-I} \leqq Adv_{A-NL} \cdot 2^{2\lambda} \leqq O((q^2/p) \cdot 2^{2\lambda}).$$

By Lemma 2, if $\lambda < \log p - \omega(\log\log p)$, Adv_{A-I} is negligible.

■

Theorem 2. In generic bilinear group model, the proposed LR-CL-KE scheme is semantically secure against chosen ciphertext attacks of Type II adversary (A_{II}, honest-but-curious CA) in the continual leakage model.

Proof: Let A_{II} be of Type II adversary who simulates an honest-but-curious CA. Thus, A_{II} knows the CA's system secret key and does not need to issue the *Certificate generation query* and *Certificate generation leak query* to the challenger B in the security game $G_{LR-CB-KE}$

- *Setup Phase:* As the $Setup$ phase in the proof of Theorem 1, B takes as input a security parameter τ and performs the $Setup$ algorithm to produce the CA's system secret key SSK and public parameters $PP = (\mathbb{G}, \mathbb{G}_T, p, g, \hat{e}, SPK = SSK \cdot g, U, V, E, D)$ of the proposed LR-CL-KE scheme. Also, three lists L_G, L_T and L_K are constructed to record the queries issued by A_{II}. Finally, B sends the corresponding bit-strings of several public parameters g, U, V, SPK to A_{II}. Since A_{II} is an honest-but-curious CA, B also sends the corresponding bit-strings of the CA's system secret key SSK to A_{II}.
- *Query phase:* Since A_{II} knows the CA's system secret key, it does not need to issue the *Certificate generation query* and *Certificate generation leak query* to B. In this phase, A_{II} can adaptively issue the following queries at most q times.
  - Q_G, Q_T, Q_P, *User key generation query, Private key query, Public key replace query:* These queries are identical to those queries presented in the proof of Theorem 1.
  - *Public key retrieve query* (ID_i): Upon receiving the request with identity ID_i, B uses ID_i to search the list L_K to gain the user's public key (UPK_i, CPK_i), and returns the corresponding bit-strings $(\Xi UPK_i, \Xi CPK_i)$ to A_{II}. It is worth mentioning that since A_{II} is of Type II adversary, A_{II} possesses the CA's system secret key SSK. In this case, B uses the queries Q_G,Q_T and Q_P to gain the corresponding polynomials of CPK_i and CSK_i of identity ID_i while updating the record of ID_i in the list L_K.
  - *Decrypt (Decapsulation) query* (ID, (C, CT)): Upon receiving the request with identity ID and the ciphertext (C, CT), B performs the following steps to gain the encryption key EK and the plain-text m:
    1) B uses ID to find the user's private key $(\Theta USK, \Theta CSK)$ in L_K. If the user's private key ΘUSK is not recorded in L_K, B issues the *User key generation query* (ID). Moreover, if the certificate ΘCSK is not recorded in L_K, B uses the records of the queries Q_G, Q_T and Q_P to gain the corresponding polynomials of ΘCPK and ΘCSK of identity ID.
    2) B transforms the ciphertext C to the polynomial ΘC in L_G, and computes two polynomials $\Theta EK_1 = \Theta USK \cdot \Theta C$ and $\Theta EK_2 = \Theta CSK \cdot \Theta C$. Moreover, B transforms ΘEK_1 and ΘEK_2 to obtain the corresponding bit-strings ΞEK_1 and ΞEK_2, respectively. Hence, B can gain the encryption key $EK = \Xi EK_1 \oplus \Xi EK_2$. Finally, B returns the encryption key EK and the plain-text $m = D_{EK}(CT)$ to A_{II}.
  - *Decrypt (Decapsulation) leak query* $(ID, j, f_{D,j}, h_{D,j})$: Upon receiving the request with the j-th *Decrypt query* for identity ID and two leakage functions $f_{D,j}$, $h_{D,j}$, B computes and returns the leakage information $(\Lambda f_{D,j}, \Lambda h_{D,j})$ to A_{II}, where $\Lambda f_{D,j} = f_{D,j}(USK_{j-1,1}, CSK_{j-1,1}, k)$ and $\Lambda h_{D,j} = h_{D,j}(USK_{j-1,2}, CSK_{j-1,2}, k, EKI_1, EKI_2, EK)$. It is worth mentioning that for the j-th *Decrypt query*, A_{II} can issue the *Decrypt leak query* only once.
- *Challenge phase:* This phase is similar to the $Challenge$ phase described in the proof of Theorem 1. The only difference is that ID^* is disallowed to be issued in the *Private key query* and *Public key replace query* in the *Query* phase since A_{II} is an honest-but-curious CA.
- *Guess phase:* The adversary A_{II} outputs $b' \in \{0, 1\}$. If $b' = b$, we say that A_{II} wins the game $G_{LR-CB-KE}$.

By similar arguments in the proof of Theorem 1, the total number of elements in both L_G and L_T satisfies the inequality $|L_G| + |L_T| \leqq 5 + 3q_O + 2q_{UKG} + 4q_D \leqq 4q$. The maximal degrees of multivariate polynomials in L_G and L_T are at most 2 and 4, respectively.

- ***Without issuing Decrypt leak query***: By similar arguments in the proof of Theorem 1, we have Pr[Case

$1] \leqq (2/p)^{(2^{|L_G|})} + (4/p)^{(2^{|L_T|})} \leqq (4/p)(|L_G| + |L_T|)^2 \leqq 64q^2/p$ and Pr[Case 2] $\leqq$ 1/2. The success probability that A_{II} wins $G_{LR-CB-KE}$ without issuing *Decrypt leak query*, denoted by $\Pr_{A-II-NL}$, satisfies the inequality

$$\Pr_{A-II-NL} \leqq Pr[Case\ 1] + Pr[Case\ 2]$$
$$\leqq 64q^2/p + (1/2).$$

Therefore, the advantage of A_{II} wins $G_{LR-CB-KE}$ without issuing *Decrypt leak query*, denoted by $Adv_{A-II-NL}$, satisfies the inequality

$$Adv_{A-II-NL} \leqq |64q^2/p + (1/2) - (1/2)|$$
$$= 64q^2/p = O(q^2/p).$$

If $q = poly(\log p)$, $Adv_{A-II-NL}$ is negligible.

- ***With issuing Decrypt leak query***: A_{II} is allowed to issue the *Decrypt leak query*. By $\Lambda f_{D,j}$ and $\Lambda h_{D,j}$, the leakage information about $f_{D,j}(USK_{j-1,1}, CSK_{j-1,1}, k)$ and $h_{D,j}(USK_{j-1,2}, CSK_{j-1,2}, k, EKI_1, EKI_2, EK)$ are discussed as follows:

■ $(CSK_{j-1,1}, CSK_{j-1,2})$: Since A_{II} knows the user's complete certificate CSK, it does not need to obtain the leakage information of CSK.

■ k: k is randomly chosen for each *Decrypt query* and is used to update the user's private key USK. At most 2λ bits of the user's complete private key USK is leaked to A_{II}.

■ $(USK_{j-1,1}, USK_{j-1,2})$: The user's private key USK satisfies the equality $USK = USK_{j-1,1} \cdot USK_{j-1,2} = USK_{j,1} \cdot USK_{j,2}$. Note that the leaked information of both $USK_{j-1,1}$ and $USK_{j-1,2}$ is independent of that of both $USK_{j,1}$ and $USK_{j,2}$. Hence, at most λ bits of $USK_{j-1,1}$ and $USK_{j-1,2}$ are leaked to A_{II}, namely, at most λ bits of USK are leaked to A_{II}.

■ (EKI_1, EKI_2, EK): At most λ bits of the encryption key EK are leaked to A_{II}.

Now, let us evaluate the success probability that A_{II} wins the game $G_{LR-CB-KE}$ with issuing the *Decrypt leak query*, denoted by $\Pr_{A-II}$. Since A_{II} simulates the role of CA who owns the complete system secret key SSK, it may know the target user's certificate CSK completely. The useful information of outputting a correct bit b' is determined by the leakage information about the target user's private key USK. For convenience, two events of $\Pr_{A-II}$ are defined as follows:

1) The event $EUSK$ denotes that A_{II} may gain the user's private key USK completely from the leakage information $\Lambda f_{D,j}$ and $\Lambda h_{D,j}$. In addition, denotes the complement event of $EUSK$.
2) The event EB denotes that A_{II} may output a correct b'.

TABLE I **Executing Time of Two Operations on Mobile Device and PC**

	T_{bp}	T_{ex}
Mobile device with 624 MHz processor	96.2 ms	30.67 ms
PC with 3 GHz processor	20.1 ms	6.38 ms

The success probability $\Pr_{A-II}$ with issuing the *Decrypt leak query* satisfies the inequality

$$\Pr_{A-II} = \Pr[EB]$$
$$= \Pr[EB \wedge EUSK] + \Pr[EB \wedge \overline{EUSK}]$$
$$\leqq \Pr[EUSK] + \Pr[EB \wedge \overline{EUSK}].$$

Under the condition $\overline{EUSK}$, the helpful information to output a correct bit is at most λ bits of the encryption key EK. Since A_{II} have 1/2 probability to guess the correct bit, $\Pr[EB \wedge \overline{EUSK}]$ is still 1/2 on average. Thus, we have

$$\Pr_{A-II} \leqq \Pr[EUSK] + 1/2.$$

Hence, the advantage of A_{II} with issuing the *Decrypt leak query*, denoted by Adv_{A-II}, satisfies the inequality

$$Adv_{A-II} \leqq |\Pr_{A-II} - 1/2| = \Pr[EUSK].$$

In the situation without issuing *Decrypt leak query*, A_{II}'s advantage has the inequality $Adv_{A-II-NL} \leqq 64q^2/p = O(q^2/p)$. Since A_{II} can learn at most 2λ bits of the user's certificate USK or the encryption key EK, we have

$$Adv_{A-II} \leqq Adv_{A-II-NL} \cdot 2^{2\lambda} \leqq O((q^2/p) \cdot 2^{2\lambda}).$$

By Lemma 2, if $\lambda < \log p - \omega(\log\log p)$, Adv_{A-II} is negligible. ■

VI. PERFORMANCE ANALYSIS

In this section, we demonstrate the performance analysis of the proposed LR-CB-KE scheme. Two notations are defined to represent the computation costs of two operations.

- T_{bp}: The computation cost of performing a bilinear pairing operation $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.
- T_{ex}: The computation cost of performing an exponentiation or inverse operation on $\mathbb{G}$ or $\mathbb{G}_T$.

The computation cost of performing a multiplication operation on $\mathbb{G}$ or $\mathbb{G}_T$ is negligible because it is smaller than both T_{bp} and T_{ex} [37], [38]. Table I lists the simulation results (executing time, in milliseconds) of two operations on both mobile device and PC platforms [42]. The security option of the bilinear group order is equal to the security level of 1024-bit RSA. In addition, the mobile device platform is a Linux-based personal digital assistant equipped with a 624-MHz PXA270 processor. The PC platform is a Microsoft-window-based desktop equipped with a 3-GHz Pentium processor. Table II demonstrates the comparisons between the previously proposed LR-CBE schemes [33]–[35] and our LR-CB-KE scheme in terms of security properties, the computation costs and executing time (in milliseconds) of the encryption and

TABLE II Comparisons Between the Previously Proposed LR-CBE Schemes and Our LR-CB-KE Scheme

	Yu *et al.*'s scheme [33]	Guo *et al.*'s scheme [34]	Li *et al.*'s scheme [35]	Our scheme
Leakage of a user's private key and certificate	Yes	Yes	Yes	Yes
Leakage of system secret key	Yes	No	No	Yes
Leakage of random values	Yes	No	No	Yes
Overall leakage amount	Bounded	Bounded	Unbounded	Unbounded
Encryption				
- Computation cost	T_{bp}+$4T_{ex}$	T_{bp}+$4T_{ex}$	T_{bp}+$3T_{ex}$	$2T_{bp}$+$4T_{ex}$
- Executing time on mobile device	218.88ms	218.88ms	188.21ms	315.08 ms
- Executing time on PC	45.62ms	45.62ms	39.24ms	65.72 ms
Decryption				
- Computation cost	$4T_{bp}$+T_{ex}	$2T_{bp}$+$3T_{ex}$	T_{bp}+$5T_{ex}$	$4T_{bp}$+$2T_{ex}$
- Executing time on mobile device	415.47ms	284.41ms	249.55ms	446.14 ms
- Executing time on PC	86.78ms	59.34ms	52ms	93.16 ms

decryption phases. Although the performance of our scheme is worse than the existing LR-CBE schemes, our scheme can be efficiently implemented on both mobile device and PC platforms. The executing time of both phases implemented on the mobile device is less than 0.5 second while that implemented on the PC is less than 0.1 second. However, the leakage models of these existing LR-CBE schemes [33]–[35] have several restrictions and do not offer complete leakage abilities of adversaries as mentioned in earlier section. The point is that our LR-CB-KE scheme with overall unbounded leakage property which permits adversaries to continuously gain partial information of the system secret key of a trusted certificate authority (CA), the private keys and certificates of users, and random values.

VII. CONCLUSION

In the past, the leakage models of the existing LR-CBE or LR-CB-KE schemes have several restrictions and do not offer complete leakage abilities of adversaries. In this article, a new continuous leakage model of LR-CB-KE scheme has been defined. The new model allows adversaries to continuously gain partial information of a user's private key and certificate, the CA's system secret key, and random values. In the new continuous leakage model, the first LR-CB-KE scheme with overall unbounded leakage property has been proposed. In the generic bilinear group model, we formally demonstrated that the proposed LR-CB-KE scheme is semantically secure against chosen ciphertext attacks (CCA1) of both adversaries. Finally, performance analysis is given to demonstrate that the proposed LR-CB-KE scheme can be efficiently implemented on bothmobile device and PC platforms.

REFERENCES

[1] R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[2] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.

[3] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. CRYPTO'84*, 1984, vol. 196, pp. 47–53.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. CRYPTO'01*, 2001, vol. 2139, pp. 213–229.

[5] S. S. Al-Riyami, and K. G. Paterson, "Certificateless public key cryptography," in *Proc. ASIACRYPT'03*, 2003, vol. 2894, pp. 452–473.

[6] Y.-M. Tseng and T.-T. Tsai, "Efficient revocable ID-based encryption with a public channel," *Comput. J.*, vol. 55, no. 4, pp. 475–486, 2012.

[7] T.-T. Tsai and Y.-M. Tseng, "Revocable certificateless public key encryption," *IEEE Syst. J.*, vol. 9, no. 3, pp. 824–833, Sep. 2015.

[8] C. Gentry, "Certificate-based encryption and the certificate revocation problem," in *Proc. EUROCRYPT'03*, 2003, vol. 2656, pp. 272–293.

[9] D. Galindo, P. Morillo, and C. Rafols, "Improved certificate-based encryption in the standard model," *J. Syst. Softw.*, vol. 81, no. 7, pp. 1218–1226, 2008.

[10] Y. Lu and J. Li, "Efficient certificate-based encryption scheme secure against key replacement attacks in the standard model," *J. Inf. Sci. Eng.*, vol. 30, no. 5, pp. 1553–1568, 2014.

[11] J. Li, Z. Wang, and Y. Zhang, "Provably secure certificate-based signature scheme without pairings," *Inf. Sci.*, vol. 233, pp. 313–320, 2013.

[12] Y.-H. Hung, S.-S. Huang, and Y.-M. Tseng, "A short certificate-based signature scheme with provable security," *Inf. Technol. Control*, vol. 45, no. 3, pp. 243–253, 2016.

[13] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proc. CRYPTO'96*, 1996, vol. 1163, pp. 104–113.

[14] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. CRYPTO'99*, 1999, vol. 1666, pp. 388–397.

[15] J. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in *Proc. CRYPTO'09*, 2009, vol. 5677, pp. 36–54.

[16] S. Faust, C. Hazay, J. B. Nielsen, P. S. Nordholt, and A. Zottarel, "Signature schemes secure against hard-to-invert leakage," in *Proc. ASIACRYPT'12*, 2012, vol. 7658, pp. 98–115.

[17] D. Galindo and S. Virek, "A practical leakage-resilient signature scheme in the generic group model," in *Proc. SAC'12*, 2013, vol. 7707, pp. 50–65.

[18] Y. Dodis, K. Haralambiev, A. Lopez-Alt, and D. Wichs, "Efficient public-key cryptography in the presence of key leakage," in *Proc. ASIACRYPT'10*, 2010, vol. 6477, pp. 613–631.

[19] J. Alawatugoda, D. Stebila, and C. Boyd, "Continuous after-the-fact leakage-resilient eck-secure key exchange," in *Proc. IMA Int. Conf. Cryptography Coding*, 2015, pp. 277–294.

[20] J.-D. Wu, Y.-M. Tseng, and S.-S. Huang, "Efficient leakage-resilient authenticated key agreement protocol in the continual leakage eCK model," *IEEE Access*, vol. 6, no. 1, pp. 17130–17142, 2018.

[21] A. Akavia, S. Goldwasser, and V. Vaikuntanathan, "Simultaneous hardcore bits and cryptography against memory attacks," in *Proc. TCC'09*, 2009, vol. 5444, pp. 474–495.

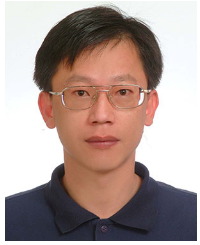
[22] E. Kiltz and K. Pietrzak, "Leakage resilient Elgamal encryption," in *Proc. ASIACRYPT'10*, 2010, vol. 6477, pp. 595–612.

[23] M. Naor and G. Segev, "Public-key cryptosystems resilient to key leakage," *SIAM J. Comput.*, vol. 41, no. 4, pp. 772–814, 2012.

[24] S. Liu, J. Weng, and Y. Zhao, "Efficient public key cryptosystem resilient to key leakage chosen ciphertext attacks," in *Proc. CTRSA'13*, 2013, vol. 7779, pp. 84–100.

[25] S. Li, F. Zhang, Y. Sun, and L. Shen, "Efficient leakage-resilient public key encryption from DDH assumption," *Cluster Comput.*, vol. 16, no. 4, pp. 797–806, 2013.
[26] D. Galindo, J. Grobschadl, Z. Liu, P.K. Vadnala, and S. Vivek, "Implementation of a leakage-resilient ElGamal key encapsulation mechanism," *J. Cryptographic Eng.*, vol. 6, no. 3, pp. 229–238, 2016.
[27] J.-D. Wu, Y.-M. Tseng, and S.-S. Huang, "An identity-based authenticated key exchange protocol resilient to continuous key leakage," *IEEE Syst. J.*, vol. 13, no. 4, pp. 3968–3979, Dec. 2019.
[28] J.-D. Wu, Y.-M. Tseng, S.-S. Huang, and T.-T. Tsai, "Leakage-resilient certificate-based signature resistant to side-channel attacks," *IEEE Access*, vol. 7, no. 1, pp. 19041–19053, 2019.
[29] D. Boneh, X. Boyen, and E. J. Goh, "Hierarchical identity-based encryption with constant size ciphertext," in *Proc. EUROCRYPT'05*, 2005, vol. 3494, pp. 440–456.
[30] Z. Brakerski, Y.T. Kalai, J. Katz, and V. Vaikuntanathan, "Cryptography resilient to continual memory leakage," in *Proc. 51st Annu. IEEE Symp. Foundations Comput. Sci.*, 2010, pp. 501–510.
[31] T.-H. Yuen, S. S. M. Chow, Y. Zhang, and S.-M. Yiu, "Identity-based encryption resilient to continual auxiliary leakage," in *Proc. EUROCRYPT'12*, 2012, vol. 7237, pp. 117–134.
[32] J. Li, Y. Guo, Q. Yu, Y. Lu, and Y. Zhang, "Provably secure identity-based encryption resilient to post-challenge continuous auxiliary input leakage," *Secur. Commun. Netw.*, vol. 9, no. 10, pp. 1016–1024, 2016.
[33] Q. Yu, J. Li, Y. Zhang, W. Wu, X. Huang, and Y. Xiang, "Certificate-based encryption resilient to key leakage," *J. Syst. Softw.*, vol. 116, pp. 101–112, 2016.
[34] Y. Guo, J. Li, Y. Lu, Y. Zhang, and F. Zhang, "Provably secure certificate-based encryption with leakage resilience," *Theor. Comput. Sci.*, vol. 711, pp. 1–10, 2018.
[35] J. Li, Y. Guo, Q. Yu, Y. Lu, Y. Zhang, and F. Zhang, "Continuous leakage-resilient certificate-based encryption," *Inf. Sci.*, vol. 355-356, pp. 1–14, 2016.
[36] J.-D. Wu, Y.-M. Tseng, and S.-S. Huang, "Leakage-resilient ID-based signature scheme in the generic bilinear group model," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 3987–4001, 2016.
[37] M. Scott, N. Costigan, and W. Abdulwahab, "Implementing cryptographic pairings on smart-cards," in *Proc. CHES'06*, 2006, vol. 4249, pp. 134–147.
[38] M. Scott, "On the efficient implementation of pairing-based protocols," in *Proc. Cryptography Coding*, 2011, vol. 7089, pp. 296–308.
[39] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *Proc. EUROCRYPT'97*, 1997, vol. 1233, pp. 256–266.
[40] U. Maurer and S. Wolf, "Lower bounds on generic algorithms in groups," in *Proc. EUROCRYPT'98*, 1998, vol. 1403, pp. 72–84.
[41] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.
[42] H. Xiong and Z. Qin, "Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 7, pp. 1442–1455, Jul. 2015.

JUI-DI WU received the B.S. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2006. He received the M.S. degree and the Ph.D. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2008 and 2020, respectively. His research interests include applied cryptography, pairing-based cryptography, and leakage-resilient cryptography.



YUH-MIN TSENG received the BS degree from National Chiao Tung University, Hsinchu, Taiwan, in 1988, the MS degree from National Taiwan University, Taipei, Taiwan, in 1990, and the Ph.D. degree from National Chung Hsing University, Taichung, Taiwan, in 1999. He is currently a professor in the Department of Mathematics and the dean of the College of Science, National Changhua University of Education, Changhua, Taiwan. His research interests include cryptography, network security, computer network, and mobile communications. In 2006, he received the Wilkes Award from the British Computer Society. He has published more than 100 scientific journal and conference papers on various research areas of cryptography, security, and computer network. He serves as an Editor of several international journals. He is a member of the IEEE Computer Society, IEEE Communications Society, and the Chinese Cryptology and Information Security Association (CCISA).

SEN-SHAN HUANG received the Ph.D. from the University of Illinois at Urbana-Champaign under the supervision of Professor Bruce C. Berndt. He is currently a professor in the Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include number theory, cryptography, and network security.

TUNG-TSO TSAI received the M.S. degree from the Department of Applied Mathematics, National Hsinchu University of Education, Taiwan, in 2009. He received the Ph.D. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2014. He is currently a Senior Engineer in the Department of Research, Foxconn, Taipei, Taiwan. His research interests include applied cryptography and pairing-based cryptography.