ITC 4/49
Information Technology and Control
Vol. 49 / No. 4 / 2020
pp. 464-481
DOI 10.5755/j01.itc.49.4.25927

Leakage-Resilient Outsourced Revocable Certificateless Signature with a Cloud Revocation Server

Received 2020/04/25 | Accepted after revision 2020/10/13

http://dx.doi.org/10.5755/j01.itc.49.4.25927

HOW TO CITE: Tseng, Y.-M., Wu, J.-D., Huang, S. S., Tsai, T. T. (2020). Leakage-Resilient Outsourced Revocable Certificateless Signature with a Cloud Revocation Server. *Information Technology and Control*, 49(4), 464-481. https://doi.org/10.5755/j01.itc.49.4.25927

Leakage-Resilient Outsourced Revocable Certificateless Signature with a Cloud Revocation Server

Yuh-Min Tseng, Jui-Di Wu, Sen-Shan Huang and Tung-Tso Tsai
Department of Mathematics, National Changhua University of Education, Changhua 500, Taiwan

Corresponding author: ymtseng@cc.ncue.edu.tw

Certificateless public-key system (CL-PKS) is a significant public-key cryptography and it solves both the key escrow and certificate management problems. Outsourced revocable certificateless public-key system (OR-CL-PKS) with a cloud revocation server (CRS) not only provides a revocation mechanism, but also further outsources the revocation functionality to the CRS to reduce the computational burden of the key generation center (KGC). Recently, side-channel attacks have threatened some existing conventional cryptography (including CL-PKS). Indeed, adversaries can apply side-channel attacks to derive fractional constituents of private (or secret) keys to damage the security of these cryptographic protocols (or schemes). To withstand such attacks, leakage-resilient cryptography is an attractive approach. However, little research concerns with leakage-resilient certificateless cryptography. In this paper, the *first* leakage-resilient outsourced revocable certificateless signature (LR-ORCLS) scheme is presented. The proposed scheme allows adversaries to continually derive fractional constituents of private (or secret) keys and possesses overall unbounded leakage property. In the generic bilinear group (GBG) model, our scheme is shown to be existential unforgeable against adversaries. Finally, the comparisons between the proposed scheme and the previous revocable certificateless signature schemes are provided to demonstrate the merits of the proposed scheme.

KEYWORDS: Signature, Certificateless signature, Revocation, Side-channel attacks, Leakage-resilience.

1. Introduction

Certificateless public-key system (CL-PKS) [1] is a significant public-key cryptography. A CL-PKS setting includes two kinds of participants, namely, users and a key generation center (KGC). The KGC first applies the identity information of a user to derive her/his identity key, while the user also selects a secret key and sets the associated public key. Hence, the user's private key consists of two components, namely, identity key and self-selected secret key. Since the KGC is unable to know self-selected secret keys of users, the CL-PKS avoids both the key escrow problem in ID-based public-key systems (ID-PKS) [4, 18] and the certificate management in traditional public-key systems [8, 16].

In a public-key system, how to revoke compromised users from the system is an essential issue. In some circumstances, users' public keys have to be revoked before their expirations. The certificate revocation list (CRL) [11] is a well-known revocation method in traditional public-key systems. However, this method cannot be applied to both ID-PKS and CL-PKS settings because they do not employ the usage of certificates. Based on the revocation idea in [25], two revocable certificateless encryption schemes [19, 23] were proposed. In 2014, Sun *et al.* [21] presented a revocable certificateless signature (RCLS) scheme in the random oracle model. To enhance the security, Tsai *et al.* [24] proposed a new RCLS scheme in the standard model. In 2016, Hung *et al.* [12] presented a revocable certificateless short signature (RCLSS) scheme. In the RCLSS scheme, the signature size is only a group element. In all RCLS and RCLSS schemes mentioned above, the KGC is responsible for performing the revocation functionality. Recently, Du *et al.* [7] constructed an outsourced RCLS (ORCLS) scheme with a cloud revocation server (CRS). In the ORCLS scheme, the revocation functionality is outsourced to the CRS to reduce the computational burden of the KGC.

Recently, conventional cryptography has suffered from a new type of attack, called "side-channel attacks", such as timing attack [5, 14] and power analysis [15]. Adversaries can apply side-channel attacks to derive fractional constituent of a user's secret (or private) key to damage the security of conventional cryptography. To withstand such attacks, leakage-resilient cryptography is an attractive approach. Up to now, little research has benn concerned with leakage-resilient certificateless public-key cryptography. In the paper, our aim is to design the *first* leakage-resilient ORCLS (LR-ORCLS) scheme.

1.1. The Concept of Leakage-Resilient Cryptography

Let us introduce the concept of leakage-resilient cryptography here. Indeed, a cryptographic scheme typically includes several computational algorithms. Meanwhile, an adversary can apply side-channel attacks to derive fractional constituent of private (or secret) keys used in each computational algorithm. For representing leakage information, let f and $f(\pi)$, respectively, be a leakage function and its output, where π denotes the function input, such as private (or secret) keys. The bit length of the output $f(\pi)$ in each computational algorithm is bounded to a security parameter λ. For leakage-resilient cryptography, there are two leakage models, namely, bounded leakage model and continual leakage model. For the bounded leakage model [2, 13], the overall leakage bit sizes of private (or secret) keys in a cryptographic scheme is restricted during the life cycle. However, the restriction is unpractical. On the other hand, the most accredited model is the continual leakage model that permits adversaries complete leakage-invocated abilities and possesses overall unbounded leakage property [9, 28, 29, 30]. In the continual leakage model, there are four properties as indicated below:

- *Only computation leakage*: An adversary is only permitted to derive fractional constituent of private (or secret) keys involved in the computational algorithm.
- *Bounded leakage of single computational algorithm*: In each computational algorithm, the bit size of the leakage function output $f(\varpi)$ is bounded to a security parameter λ.
- *Independent leakage*: Fractional constituents of private (or secret) keys in any two computational algorithms are mutually independent. To realize the property, the private (or secret) keys must be updated after/before running each computational algorithm.

– *Overall unbounded leakage*: By the independent leakage property, the total bit size of leakage information is unbounded.

In the continual leakage model, let us first recall several previous leakage-resilient signature schemes based on traditional public-key, ID-PKS and CL-PKS settings. Galindo and Vivek [9] presented a leakage-resilient signature (LRS) scheme based on traditional public-key settings. Galindo and Vivek's scheme owns overall unbounded leakage property and its security is proved in the generic bilinear group (GBG) model [3]. For improving the computational performance of Galindo and Vivek's scheme, Tang *et al.* [22] then presented a modified LRS scheme. In ID-PKS settings, the first leakage-resilient ID-based signature was presented by Wu *et al.* [26]. In their scheme, adversaries are permitted to continually derive fractional constituent of private (or secret) keys. Moreover, Wu *et al.* proved that their scheme is existentially unforgeable against ID and adaptive chosen-message attacks in the GBG model. In CL-PKS settings, based on Xiong *et al.*'s leakage-resilient certificateless public-key encryption scheme [31], Zhou *et al.* [32] presented a leakage-resilient certificateless signcryption scheme under the bounded leakage model. In 2018, Wu *et al.* [27] defined a new adversary model of leakage-resilient CLS (LR-CLS) schemes by adding several key leakage queries under the continual leakage model. Meanwhile, Wu *et al.* [27] also presented the first LR-CLS scheme which was proved to be secure against adversaries in the new adversary model.

1.2. Contributions and Organization

Until now, no leakage-resilient RCLS (LR-RCLS) or ORCLS (LR-ORCLS) scheme withstanding side-channel attacks has been proposed. Indeed, a LR-RCLS scheme can be derived from a LR-ORCLS scheme because the KGC is responsible to play the roles of both the KGC and the CRS in the LR-ORCLS scheme. In the meantime, a LR-ORCLS scheme is better than a LR-RCLS scheme because the revocation functionality is outsourced to the CRS to reduce the computational burden of the KGC. Hence, we will aim at the design of the *first* LR-ORCLS scheme.

As mentioned earlier, several certificateless cryptographic schemes were proposed, namely, leakage-resilient certificateless signcryption (LR-CLSE) scheme scheme [31], leakage-resilient certificateless signcryption (LR-CLSE) scheme [32] and leakage-resilient certificateless signature (LR-CLS) scheme [27]. Table 1 lists the comparisons between the previous certificateless cryptographic schemes [31, 32, 27] and our LR-ORCLS scheme in terms of cryptographic functionality, overall leakage property, outsourced functionality and revocation functionality. Indeed, these certificateless cryptographic schemes [31, 32, 27] did not address the revocation problem. Indeed, in a public-key system, how to revoke compromised users from the system is an essential issue because the compromised users' public keys have to be revoked before their expirations [23, 25].

In this article, the *first* LR-ORCLS scheme is proposed. We first define the syntax of LR-ORCLS

Table 1
Comparisons between previous certificateless cryptographic schemes and our scheme

Scheme	Cryptographic functionality	Overall leakage property	Revocation functionality	Outsourced functionality
Xiong *et al.*'s LR-CLPKE scheme [31]	Encryption	Bounded	No	No
Zhou *et al.*'s LR-CLSE scheme [32]	Signcryption	Bounded	No	No
Wu *et al.*'s LR-CLS scheme [27]	Signature	Unbounded	No	No
Our LR-ORCLS scheme	Signature	Unbounded	Yes	Yes

schemes which consists of three participants, namely, a KGC, a CRS and users (signers and verifiers). The KGC is responsible to generate each user's identity key. At each period, the CRS generates the time update keys of all non-revoked users. Hence, a user's private key consists of three components, namely, identity key, time update key and self-selected secret key. By adding several key leakage queries, we define a new adversary model of LR-ORCLS schemes, which consists of three types of adversaries, namely, Type I (outsider), Type II (revoked user) and Type III (honest-but-curious KGC). In the new adversary model, adversaries are permitted to continually derive fractional constituent of the KGC's master secret key, the CRS's cloud secret key and a signer's secret key used in the associated algorithms. The proposed scheme is shown to be existential unforgeable against Types I, II and III adversaries. Finally, the comparisons between the proposed scheme and the previously related RCLS/ORCLS schemes are provided to demonstrate the merits of the proposed scheme.

The remains of this paper are organized as below. Section 2 demonstrates several preliminaries. The syntax and adversary model of LR-ORCLS schemes are defined in Section 3. In Section 4, a novel LR-ORCLS scheme is proposed. In Section 5, the security of the proposed scheme is formally shown. The comparisons between our scheme and several previous RCLS/ORCLS schemes are given in Section 6. Conclusions and future work are presented in Section 7.

2. Preliminaries

2.1. Bilinear Groups

Let G and G_T denote, respectively, an additive group and a multiplicative group of a prime order p. Let P be an arbitrary generator of G. A bilinear pairing $\hat{e}$: $G\times G \rightarrow G_T$ is an admissible mapping with three properties.

1. Bilinear property: $\hat{e}(aP, bP)=\hat{e}(P, P)^{ab}$, for $a, b\in Z_q^*$.
2. Non-degenerate property: $\hat{e}(P, P)\neq 1$, which is viewed as a generator of G_T.
3. Efficient computable property: $\hat{e}(P, Q)$ is computed efficiently, where $Q=cP\in G$ and $c\in Z_p^*$.

For more detailed definitions of groups, maps and the related parameters, a reader refers to several literatures [4, 17].

2.2. Generic Bilinear Group (GBG) Model and Security Assumption

By extending the generic group model [20], Boneh *et al.* [3] presented the generic bilinear group (GBG) model. This model is applied to prove the security of cryptographic schemes/protocols. In this model, if an adversary can find a collision of a group with a large order, it is said that the discrete logarithm problem on the group is resolved.

In the GBG model, there are an additive group G and a multiplicative group G_T with the same prime order p. In this model, two random injective mappings Φ_G: $Z_p\rightarrow\xi_G$ and Φ_T: $Z_p\rightarrow\xi_T$ are, respectively, applied to encode all elements of G and G_T to distinct bit-strings. In which, ξ_G denotes the set of the encoded bit-strings of G and ξ_T is the set of the encoded bit-strings of G_T. Two sets satisfy $\xi_G\cap\xi_T=\phi$ and $|\xi_G|=|\xi_T|=p$. In the GBG model, if adversaries want to perform three group operations, they must issue the corresponding public queries Q_G, Q_T and Q_p to a challenger in a security game. Two queries Q_G and Q_T, respectively, denote the addition on G and the multiplication on G_T. The query Q_p denotes the bilinear pairing $\hat{e}$. For $s, t\in Z_p^*$, three queries satisfy the following properties.

- $Q_G(\Phi_G(s), \Phi_G(t))\rightarrow\Phi_G(s+t \bmod p)$.
- $Q_T(\Phi_T(s), \Phi_T(t))\rightarrow\Phi_T(s+t \bmod p)$.
- $Q_p(\Phi_G(s), \Phi_G(t))\rightarrow\Phi_T(st \bmod p)$.

Let P be a generator of G, we have $P=\Phi_G(1)$ and $\hat{e}(P, P)=\Phi_T(1)$.

After finishing the security game, if an adversary discovers a collision in G or G_T, it is said that the discrete logarithm problem on G or G_T is resolved. The discrete logarithm assumption is presented as given below.

- **Discrete logarithm (DL) assumption:** Let G and G_T be an additive group and a multiplicative group of a prime order p, respectively. Given a group element $sP\in G$ or $\hat{e}(P, P)^s\in G_T$ for unknown $s\in Z_p^*$, no algorithm A with non-negligible probability is able to derive a in polynomial time.

2.3. The Measure of Leakage Information

Let us introduce the concept of entropy here. The entropy of a random variable denotes the measure of uncertainty in statistical mechanics. A secret (or private) key can be viewed as a finite random variable. Let Z and $\Pr[Z=z]$ be a finite random variable

and the associated probability with $Z=z$. In addition, the min-entropy of Z is the predictability value of the largest probability with $Z=z$. The min-entropy and average conditional min-entropy are, respectively, presented as given below.

1 Min-entropy of Z:
$H_\infty(Z) = -\log_2(\max_z \Pr[Z = z])$.

2 Average conditional min-entropy of Z with $E=e$:
$\widetilde{H}_\infty(Z|E = e) = -\log_2(E_{e\leftarrow E}[\max \Pr[Z = z|E = e]])$.

To address the security influence due to partial leakage of a secret (or private) key, Dodis *et al.* [6] presented a consequence as indicated below.

Lemma 1. Let Z denote a secret key (i.e. a random variable). Let λ be the maximal bit-length of leakage information. Let h: $Z\rightarrow\{0,1\}^\lambda$ be a leakage function with input Z. Under the event $h(Z)$, the average conditional min-entropy on Z is $\widetilde{H}_\infty(Z|h(Z))\geqq H_\infty(Z)-\lambda$.

Typically, several secret keys (i.e. multiple random variables) are involved in the computational algorithms of a cryptographic scheme/protocol. To measure the security influence due to partial leakage of polynomials with multiple random variables, Galindo and Vivek [9] presented the following consequences.

Lemma 2. Let $Q_1, Q_2,..., Q_n$ be random variables with probability distributions. Let $F\in Z_p[Q_1, Q_2,..., Q_n]$ denote a polynomial with total degree at most d. Let P_i denote probability distributions on Z_p while $H_\infty(P_i) \geqq \log p-\lambda$ holds, where $i=1, 2, ..., n$ and $0\leqq\lambda\leqq\log p$. If all $Q_i = q_i \leftarrow Z_p$ with probability distribution P_i are independently selected, we have the probability $\Pr[F(Q_1=q_1, Q_2=q_2, ..., Q_n=q_n)=0]\leqq(d/p)2^\lambda$.

Corollary 1. $\Pr[F(Q_1=q_1, Q_2=q_2, ..., Q_n=q_n)=0]$ is negligible if $\lambda<(1-\varepsilon)\log p$, where ε denotes a positive value.

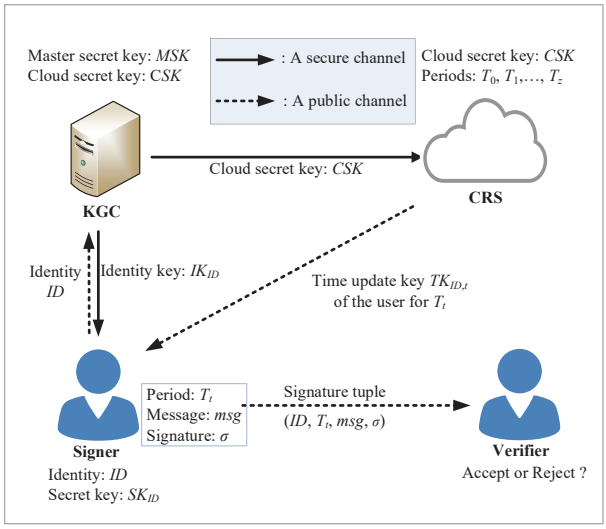
3. Syntax and Adversary Model

In this section, the syntax and adversary model of LR-ORCLS schemes are defined.

3.1. Syntax of LR-ORCLS Schemes

Let us firstly present the system operation of LR-ORCLS schemes here. In a LR-ORCLS scheme, there are three participants, namely, a key generation center (KGC), a cloud revocation server (CRS) and users (signers and verifiers). The KGC generates identity keys of all users while the CRS generates the time update keys of all non-revoked users at each period T_t. In addition, each user (signer) chooses a secret key by oneself. Without the loss of generality, a signer with identity ID at period T_t wants to sign a message msg. The signer uses his/her identity key, time update key and secret key to generate a signature σ and sends the signature tuple (ID, T_t, msg, σ) to a verifier. The system operation of the LR-ORCLS scheme is depicted in Figure 1.

Figure 1
The system operation of a LR-ORCLS scheme



Some notations are summarized below.

- MSK: the KGC's master secret key.
- MPK: the KGC's master public key.
- CSK: the CRS's cloud secret key.
- CPK: the CRS's cloud public key.
- ID: the identity of a user, where $ID\in\{0, 1\}^*$.
- SK_{ID}: the secret key of the user ID.
- PK_{ID}: the public key of the user ID.
- T_t: a period $T_t\in\{0, 1\}^*$, for $t=0, 1, ..., z$, where $z+1$ denotes the amount of periods.
- IK_{ID}: the identity key of the user ID.
- $TK_{ID,t}$: the time update key of the user ID at period T_t.
- msg: a message.
- σ: a signature.

By extending the syntaxes in [7, 27], the syntax of LR-ORCLS schemes is formally defined as below.

Definition 1. A LR-ORCLS scheme includes eight algorithms:

- *Setup*: The KGC first sets the master secret key $MSK=(MSK_{0,1}, MSK_{0,2})$, the master public key MPK, the cloud secret key $CSK=(CSK_{0,1}, CSK_{0,2})$ and the cloud public key CPK while securely sending the cloud secret key CSK to the CRS. The KGC sets z+1 periods $T_0, T_1, ..., T_z$ while publishing public parameters PP.
- *Identity key extract*: For the i-th execution of *Identity key extract* algorithm, the algorithm includes two sub-algorithms *IKExtract*-1 ($MSK_{i-1,1}$) and *IKExtract*-1 ($MSK_{i-1,2}$). Firstly, the *KGC* sets the new master secret key ($MSK_{i,1}, MSK_{i,2}$) by using ($MSK_{i-1,1}$, $MSK_{i-1,2}$). By taking as input a user ID, the *KGC* computes the user's identity key IK_{ID} and partial public key Q_{ID}. Finally, the *KGC* securely sends IK_{ID} and Q_{ID} to the user.
- *Time update key extract*: For the j-th execution of *Time update key extract* algorithm, the algorithm includes two sub-algorithms *TKExtract*-1 ($CSK_{j-1,1}$) and *TKExtract*-1 ($CSK_{j-1,2}$). The CRS sets the new cloud secret key ($CSK_{j,1}$, $CSK_{j,2}$) by using ($CSK_{j-1,1}$, $CSK_{j-1,2}$). By taking a non-revoked user ID and the current period T_t as input, the CRS generates and sends the user's time update key $TK_{ID,t}$ and partial public key $R_{ID,t}$ to the user.
- *Set secret key*: A user ID selects her/his secret key $SK_{ID}=(SK_{ID,0,1}, SK_{ID,0,2})$ and computes the partial public key PK_{ID}.
- *Set private key*: The private key of a user ID at period T_t consists of three components, namely, IK_{ID}, $TK_{ID,t}$ and SK_{ID}. The user also sets $IK_{ID}=(IK_{ID,0,1}, IK_{ID,0,2})$ and $SK_{ID}=(SK_{ID,0,1}, SK_{ID,0,2})$.
- *Set public key*: Upon receiving the partial public keys Q_{ID}, $R_{ID,t}$ and PK_{ID} of a user ID at period T_t, the user sets her/his public key tuple (Q_{ID}, $R_{ID,t}$, PK_{ID}).
- *Signing*: For the k-th execution of a user ID at period T_t, this algorithm includes two sub-algorithms *Signing*-1 ($IK_{ID,k-1,1}$, $TK_{ID,t}$, $SK_{ID,k-1,1}$) and *Signing*-2 ($IK_{ID,k-1,2}$, $SK_{ID,k-1,2}$). The signer sets the new identity key ($IK_{ID,k,1}$, $IK_{ID,k,2}$) and secret key ($SK_{ID,k,1}$, $SK_{ID,k,2}$) by using ($IK_{ID,k-1,1}$, $IK_{ID,k-1,2}$) and ($SK_{ID,k-1,1}$, $SK_{ID,k-1,2}$), respectively. By taking a message msg as input, the user applies ($IK_{ID,k,1}$, $IK_{ID,k,2}$), $TK_{ID,t}$ and ($SK_{ID,k,1}$, $SK_{ID,k,2}$) to compute a signature σ. The signature tuple is (ID, T_t, msg, σ).
- *Verifying*: By taking (ID, T_t, msg, σ) as input, a verifier returns either "accept" or "reject".

3.2. Adversary Model of LR-ORCLS Schemes

In the continual leakage model, six leakage functions $f_{IKE,i}$, $h_{IKE,i}$, $f_{TKE,j}$, $h_{TKE,j}$, $f_{SIG,k}$ and $h_{SIG,k}$ are applied to model an adversary's capabilities. Namely, the outputs of these leakage functions denote fractional constituents of the private (or secret) keys used in the associated algorithms. Here, ($f_{IKE,i}$, $h_{IKE,i}$) is used to derive fractional constituents of the master secret key ($MSK_{i,1}$, $MSK_{i,2}$) involved in the i-th execution of *Identity key extract* algorithm. ($f_{TKE,j}$, $h_{TKE,j}$) is used to derive fractional constituents of the cloud secret key ($CSK_{j,1}$, $CSK_{j,2}$) involved in the j-th execution of *Time update key extract* algorithm. Furthermore, ($f_{SIG,k}$, $h_{SIG,k}$) is applied to derive fractional constituents of the identity key ($IK_{ID,k,1}$, $IK_{ID,k,2}$) and secret key ($SK_{ID,k,1}$, $SK_{ID,k,2}$) involved in the k-th execution of *Signing* algorithm of a user ID. The private (or secret) keys leaked by the adversary in the associated computational algorithms are bounded to λ bits, where λ is a leakage parameter. Namely, the output lengths of $f_{IKE,i}$, $h_{IKE,i}$, $f_{TKE,j}$, $h_{TKE,j}$, $f_{SIG,k}$ and $h_{SIG,k}$ are bounded to λ bits, namely, $|f_{IKE,i}|, |h_{IKE,i}|, |f_{TKE,j}|, |h_{TKE,j}|, |f_{SIG,k}|, |h_{SIG,k}| \le \lambda$, where $|fun|$ represents the output length of the function fun. The inputs/outputs of six leakage functions are presented as below.

- $\Lambda f_{IKE,i}=f_{IKE,i}(MSK_{i,1}, Rf_{IKE,i})$.
- $\Lambda h_{IKE,i}=h_{IKE,i}(MSK_{i,2}, Rh_{IKE,i})$.
- $\Lambda f_{TKE,j}=f_{TKE,j}(CSK_{j,1}, Rf_{TKE,j})$.
- $\Lambda h_{TKE,j}=h_{TKE,j}(CSK_{j,2}, Rh_{TKE,j})$.
- $\Lambda f_{SIG,k}=f_{SIG,k}(IK_{ID,k,1}, SK_{ID,k,1}, Rf_{SIG,k})$.
- $\Lambda h_{SIG,k}=h_{SIG,k}(IK_{ID,k,2}, SK_{ID,k,2}, Rh_{SIG,k})$.

Here, $Rf_{IKE,i}$, $Rh_{IKE,i}$, $Rf_{TKE,j}$, $Rh_{TKE,j}$, $Rf_{SIG,k}$ and $Rh_{SIG,k}$ represent the random values involved in the associated computational algorithms.

By extending the security notions in the RCLS and ORCLS schemes [7, 12, 21, 24], the adversary model of LR-ORCLS schemes contains three types of adversaries namely, Type I (A_I, outsider), Type II (A_{II}, revoked user) and Type III (A_{III}, honest-but-curious KGC). By providing the associated leak queries, three

types of adversaries in the LR-ORCLS scheme are presented below.

- Type I adversary (A_I, outsider): A_I is permitted to retrieve the secret key SK_{ID} and time update key $TK_{ID,t}$ of any user ID for any period T_t. However, A_I does not know the master secret key MSK, but it is permitted to retrieve the identity key IK_{ID} of any user ID, except the attacking target user ID^*. Additionally, A_I can derive fractional constituent of the master secret key MSK in *Identity key extract* algorithm. A_I can also derive fractional constituent of IK_{ID} in the *Signing* algorithm.
- Type II adversary (A_{II}, revoked user): A_{II} is permitted to retrieve the secret key SK_{ID} and identity key IK_{ID} of any user ID. However, A_{II} does not know the cloud secret key CSK, but it is permitted to retrieve the time update key $TK_{ID,t}$ of any user ID for any period T_t, except TK_{ID^*,t^*} of the target user ID^* at period T_{t^*}. Additionally, A_{II} can derive fractional constituent of CSK in the *Time update key extract* algorithm.
- Type III adversary (A_{III}, honest-but-curious KGC): A_{III} is permitted to retrieve the identity key IK_{ID} and time update key $TK_{ID,t}$ of any user ID for any period T_t. Additionally, it is permitted to retrieve the secret key SK_{ID} of any user ID, except SK_{ID^*} of the attacking target user ID^*. Meanwhile, A_{III} can derive fractional constituent of the secret key SK_{ID^*} in the *Signing* algorithm.

In the continual leakage model, the security notions of LR-ORCLS schemes are defined in the following security game played by both a challenger C and an adversary A (A_I, A_{II} or A_{III}).

Definition 2. In the continual leakage model, a LR-ORCLS scheme is existential unforgeable against adaptive chosen-message attacks (UF-LR-ORCLS-ACMA) if no adversary A (A_I, A_{II} or A_{III}) wins the UF-LR-ORCLS-ACMA game with non-negligible probability in polynomial time. This game includes three phases below:

- *Setup*: A challenger C performs the *Setup* algorithm to produce the master secret key MSK=($MSK_{0,1}$, $MSK_{0,2}$) and cloud secret key CSK=($CSK_{0,1}$, $CSK_{0,2}$), and sets z+1 periods T_0, T_1, ..., T_z and public parameters PP. According to adversary type, C runs the following steps:
  - If A is of A_I, C sends PP and CSK to A.
  - If A is of A_{II}, C sends PP and MSK to A.
  - If A is of A_{III}, C sends PP, MSK and CSK to A.
- *Queries*: In this phase, A can adaptively issue the following queries to C.
  - *Identity key query* (ID): For the i-th execution, C sets the new master secret key ($MSK_{i,1}$, $MSK_{i,2}$) by using ($MSK_{i-1,1}$, $MSK_{i-1,2}$). Afterward, C uses ($MSK_{i,1}$, $MSK_{i,2}$) to generate and return the associated identity key IK_{ID}.
  - *Identity key leak query* ($f_{IKE,i}$, $h_{IKE,i}$, i): For the i-th *Identity key query*, A is permitted to issue this query only once. C returns fractional constituents ($\Lambda f_{IKE,i}$, $\Lambda h_{IKE,i}$).
  - *Time update key query* (ID, T_t): For the j-th execution, C sets the current cloud secret key ($CSK_{j,1}$, $CSK_{j,2}$) by using ($CSK_{j-1,1}$, $CSK_{j-1,2}$). Afterward, C uses ($CSK_{j,1}$, $CSK_{j,2}$), ID and T_t to generate and return the associated time update key $TK_{ID,t}$.
  - *Time update key leak query* ($f_{TKE,j}$, $h_{TKE,j}$, j): For the j-th *Time update key query*, A is permitted to issue this query only once. C returns fractional constituents ($\Lambda f_{TKE,j}$, $\Lambda h_{TKE,j}$).
  - *Public key retrieve query* (ID, T_t): C returns the associated public key tuple (Q_{ID}, $R_{ID,t}$, PK_{ID}).
  - *Public key replace query* (ID, T_t, (Q'_{ID}, $R'_{ID,t}$, PK'_{ID})): C records this public key replacement.
  - *Secret key corrupt query* (ID): C returns the secret key SK_{ID} if *Public key replace query* (ID) is never issued. Otherwise, C returns false.
  - *Signing query* (ID, T_t, msg): For the k-th execution of ID at period T_t, C sets the current identity key ($IK_{ID,k,1}$, $IK_{ID,k,2}$) and secret key ($SK_{ID,k,1}$, $SK_{ID,k,2}$) by using ($IK_{ID,k-1,1}$, $IK_{ID,k-1,2}$) and ($SK_{ID,k-1,1}$, $SK_{ID,k-1,2}$), respectively. Afterward, C uses ($IK_{ID,k,1}$, $IK_{ID,k,2}$), $TK_{ID,t}$ and ($SK_{ID,k,1}$, $SK_{ID,k,2}$) to generate and return a signature σ.
  - *Signing leak query* (ID, T_t, $f_{SIG,k}$, $h_{SIG,k}$, k): For the k-th *Signing query* of ID at period T_t, A is permitted to issue this query only once. C returns fractional constituents ($\Lambda f_{SIG,k}$, $\Lambda h_{SIG,k}$).
- *Forgery*: In this phase, A outputs a tuple (ID^*, T_t^*, msg^*, σ^*). If the following conditions hold, it is said that A wins the game.
  1. *Sign query* (ID^*, T_t^*, msg^*) is never issued.
  2. The response of the *Verify* algorithm on (ID^*, T_t^*, msg^*, σ^*) is "accept".

3 According to the adversary type, C checks the following conditions:
   a If A is of A_I, the *Identity key query* (ID^*) is never requested.
   b If A is of A_{II}, the *Time update key query* (ID^*, T_t^*) is never requested.
   c If A is of A_{III}, the *Public key replace query* (ID^*, T_t^*) or *Secret key corrupt query* (ID^*) is never requested.

4. The Proposed LR-ORCLS Scheme

The proposed LR-ORCLS scheme includes eight algorithms as below.

– *Setup*: The *KGC* chooses the related parameters $\{G, G_T, p, P, \hat{e}\}$ of bilinear pairing groups presented in Section II.A. The *KGC* then performs the following steps:
   1 Choose a random integer $x \in Z_p^*$, and compute the master secret key $MSK=x \cdot P$ and master public key $MPK=\hat{e}(P, MSK)$.
   2 Choose a random integer $y \in Z_p^*$, and compute the cloud secret key $CSK=y \cdot P$ and cloud public key $CPK=\hat{e}(P, CSK)$.
   3 Choose a random integer $a_0 \in Z_p^*$ and compute the primary master secret key $(MSK_{0,1}, MSK_{0,2}) = (a_0 \cdot P, MSK+(a_0) \cdot P)$.
   4 Choose six random integers $r, s, u, v, m, n \in Z_p^*$, and compute $R=r \cdot P, S=s \cdot P, U=u \cdot P, V=v \cdot P, M=m \cdot P, N=n \cdot P$.
   5 Choose z+1 periods $T_t \in \{0, 1\}^*$, for t=0, 1, ..., z.
   6 Publish public parameters $PP=\{G, G_T, p, P, \hat{e}, MPK, CPK, R, S, U, V, M, N\}$.
   7 Securely send CSK to the CRS. The CRS then chooses a random integer $b_0 \in Z_p^*$ and sets the primary cloud secret key $(CSK_{0,1}, CSK_{0,2}) = (b_0 \cdot P, CSK+(-b_0) \cdot P)$.

– *Identity key extract*: For the i-th execution, by taking as input a user ID, the *KGC* runs two sub-algorithms as below:
   • *IKExtract*-1 ($MSK_{i-1,1}$):
      1 Choose a random integer $a_i \in Z_p^*$, and compute $MSK_{i,1}=MSK_{i-1,1}+a_i \cdot P$.
      2 Choose a random integer $\alpha \in Z_p^*$, and compute $Q_{ID}=\alpha \cdot P$ and temporary information $TI_{IKE}=MSK_{i,1}+\alpha \cdot (R+ID \cdot S)$.
   • *IKExtract*-1 ($MSK_{i-1,2}$):
      1 Compute $MSK_{i,2}=MSK_{i-1,2}+(-a_i) \cdot P$ and $IK_{ID}=MSK_{i,2}+TI_{IKE}$.
      2 Send the user's identity key IK_{ID} and partial public key Q_{ID} to the user via a secure channel.

– *Time update key extract*: For the j-th execution, by taking a non-revoked user ID and the current period T_t as input, the CRS runs two sub-algorithms as below:
   • *TKExtract*-1 ($CSK_{j-1,1}$):
      1 Choose a random integer $b_j \in Z_p^*$, and compute $CSK_{j,1}=CSK_{j-1,1}+b_j \cdot P$.
      2 Choose a random integer $\beta \in Z_p^*$, and compute $R_{ID,t}=\beta \cdot P$ and temporary information $TI_{TKE}=CSK_{j,1}+\beta \cdot (U+(ID||T_t) \cdot V)$.
   • *TKExtract*-1 ($CSK_{j-1,2}$):
      1 Compute $CSK_{j,2}=CSK_{j-1,2}+(-b_j) \cdot P$ and $TK_{ID,t}=CSK_{j,2}+TI_{TKE}$.
      2 Send the user's time update key $TK_{ID,t}$ and partial public key $R_{ID,t}$ to the user via a public channel.

– *Set secret key*: A user ID randomly chooses an integer $z \in Z_p^*$ and computes her/his secret key $SK_{ID}=z \cdot P$ and partial public key $PK_{ID}=\hat{e}(P, SK_{ID})$.

– *Set private key*: At period T_t, the signing private key of a user ID consists of the identity key IK_{ID}, the time update key $TK_{ID,t}$ and the secret key SK_{ID}. The user runs the following steps:
   1 Choose a random integer $c_0 \in Z_p^*$ and compute the primary identity key $(IK_{ID,0,1}, IK_{ID,0,2}) = (c_0 \cdot P, IK_{ID}+(c_0) \cdot P)$.
   2 Choose a random integer $d_0 \in Z_p^*$ and compute the primary secret key $(SK_{ID,0,1}, SK_{ID,0,2})=(d_0 \cdot P, SK_{ID}+(d_0) \cdot P)$.
   3 Sets her/his primary private key tuple $((IK_{ID,0,1}, IK_{ID,0,2}), TK_{ID,t}, (SK_{ID,0,1}, SK_{ID,0,2}))$.

– *Set public key*: Upon receiving the partial public keys Q_{ID}, $R_{ID,t}$ and PK_{ID}, the user sets her/his public key tuple $(Q_{ID}, R_{ID,t}, PK_{ID})$ at period T_t.

– *Signing*: For the k-th execution of a user ID at period T_t, by taking a message msg as input, the user runs two sub-algorithms as below:
   • *Signing*-1 ($IK_{ID,k-1,1}$, $TK_{ID,t}$, $SK_{ID,k-1,1}$):
      1 Randomly choose an integer $c_k \in Z_p^*$, and compute $IK_{ID,k,1}=IK_{ID,k-1,1}+c_k \cdot P$.

2 Randomly choose an integer $d_k \in Z_p^*$, and compute $SK_{ID,k,1}=SK_{ID,k-1,1}+d_k \cdot P$.

3 Randomly choose an integer $\gamma \in Z_p^*$, and compute $\sigma_1=\gamma \cdot P$ and temporary information $TI_{SIG}=IK_{ID,k,1}+TK_{ID,t}+SK_{ID,k,1}+\gamma \cdot (M+(ID||T_t||msg) \cdot N)$.

- *Signing-2* ($IK_{ID,k-1,2}$, $SK_{ID,k-1,2}$):

1 Compute $IK_{ID,k,2}=IK_{ID,k-1,2}+(-c_k) \cdot P$ and $SK_{ID,k,2}=SK_{ID,k-1,2}+(-d_k) \cdot P$.

2 Compute $\sigma_2=IK_{ID,k,2}+SK_{ID,k,2}+TI_{SIG}$.

3 Set a signature tuple (ID, T_t, msg, $\sigma=(\sigma_1, \sigma_2)$).

– *Verifying*: Given a signature tuple (ID, T_t, msg, $\sigma=(\sigma_1, \sigma_2)$) and the associated public key tuple (Q_{ID}, $R_{ID,t}$, PK_{ID}), a verifier accepts it if $\hat{e}(P, \sigma_2) = MPK \cdot \hat{e}(Q_{ID}, R + ID \cdot S) \cdot CPK \cdot \hat{e}(R_{ID,t}, U + (ID||T_t) \cdot V) \cdot PK_{ID} \cdot \hat{e}(\sigma_1, M + (ID||T_t||msg) \cdot N)$. Otherwise, the verifier rejects it.

In the following, the signature correctness is demonstrated. By the key blinding (refreshing) technique, we have

$$
\begin{aligned}
&MSK=MSK_{0,1}+MSK_{0,2}=MSK_{1,1}+MSK_{1,2}=\ldots\\
&=MSK_{i,1}+MSK_{i,2};\\
&CSK=CSK_{0,1}+CSK_{0,2}=CSK_{1,1}+CSK_{1,2}=\ldots\\
&=CSK_{j,1}+CSK_{j,2};\\
&IK_{ID}=IK_{ID,0,1}+IK_{ID,0,2}=IK_{ID,1,1}+IK_{ID,1,2}=\ldots\\
&=IK_{ID,k,1}+IK_{ID,k,2};\\
&SK_{ID}=SK_{ID,0,1}+SK_{ID,0,2}=SK_{ID,1,1}+SK_{ID,1,2}=\ldots\\
&=SK_{ID,k,1}+SK_{ID,k,2}.
\end{aligned}
$$

Therefore, the signature correctness is demonstrated below.

$$
\begin{aligned}
&\hat{e}(P, \sigma_2) =\\
&\hat{e}(P, IK_{ID}+TK_{ID,t}+SK_{ID}+\gamma \cdot (M+(ID||T_t||msg) \cdot N))\\
&=\hat{e}(P, MSK+\alpha \cdot (R+ID \cdot S)+CSK+\beta \cdot (U+(ID||T_t) \cdot V)\\
&+SK_{ID}+\gamma \cdot (M+(ID||T_t||msg) \cdot N))\\
&=\hat{e}(P, MSK) \cdot \hat{e}(P, \alpha \cdot (R+ID \cdot S)) \cdot \hat{e}(P, CSK) \cdot \hat{e}(P, \beta \cdot (U+\\
&(ID||T_t) \cdot V)) \cdot \hat{e}(P, SK_{ID}) \cdot \hat{e}(P, \gamma \cdot (M+(ID||T_t||msg) \cdot N))\\
&=MPK \cdot \hat{e}(\alpha \cdot P, R+ID \cdot S) \cdot CPK \cdot \hat{e}(\beta \cdot P, U+(ID||T_t) \cdot V)\\
&\cdot PK_{ID} \cdot \hat{e}(\gamma \cdot P, M+(ID||T_t||msg) \cdot N)\\
&=MPK \cdot \hat{e}(Q_{ID}, R+ID \cdot S) \cdot CPK \cdot \hat{e}(R_{ID,t}, U+(ID||T_t) \cdot V) \cdot\\
&PK_{ID} \cdot \hat{e}(\sigma_1, M+(ID||T_t||msg) \cdot N).
\end{aligned}
$$

5. Security Analysis

In this section, the security analysis of the proposed LR-ORCLS scheme is given. As the UF-LR-ORCLS-ACMA game presented in Definition 2, the adversary model includes three types of adversaries, namely, Type I (A_I, outsider), Type II (A_{II}, revoked user) and Type III (A_{III}, honest-but-curious KGC). In the GBG model, three theorems are, respectively, proved to demonstrate that our scheme is existential unforgeable against all Types I, II and III adversaries in the continual leakage model.

Theorem 1. In the GBG model, our LR-ORCLS scheme is existential unforgeable against Type I adversary (A_I, outsider) in the UF-LR-ORCLS-ACMA game.

Proof. Let A_I be of Type I adversary in the UF-LR-ORCLS-ACMA game played with a challenger C. A_I may issue various queries to C at most q times in the game. In the GBG model, for performing three group operations, an adversary issues three associated queries Q_G, Q_T and Q_p. In the game, there are three phases below:

– *Setup phase*: C first runs the *Setup* algorithm of the proposed LR-ORCLS scheme to generate MSK, CSK, z+1 periods T_0, T_1, ..., T_z and $PP=\{G, G_T, p, P, \hat{e}, MPK, CPK, R, S, U, V, M, N\}$. In the following, five lists L_G, L_{GT}, L_{IKE}, L_U and L_{TKE} are constructed to record both the inputs and outputs of queries issued by A_I.

- L_G and L_{GT} are, respectively, applied to record elements of two groups G and G_T.

1 L_G includes pairs of ($\Xi G_{t,v,r}$, $\Theta G_{t,v,r}$). $\Xi G_{t,v,r}$ is a multivariate polynomial to represent an element in G and $\Theta G_{t,v,r}$ is the corresponding bit-string, where t, v and r, respectively, represent the query type t, the v-th query and r-th element in G. Initially, C stores nine pairs (ΞP, $\Theta G_{I,0,1}$), (ΞMSK, $\Theta G_{I,0,2}$), (ΞCSK, $\Theta G_{I,0,3}$), (ΞR, $\Theta G_{I,0,4}$), (ΞS, $\Theta G_{I,0,5}$), (ΞU, $\Theta G_{I,0,6}$), (ΞV, $\Theta G_{I,0,7}$), (ΞM, $\Theta G_{I,0,8}$) and (ΞN, $\Theta G_{I,0,9}$) in L_G.

2 L_{GT} includes pairs of ($\Xi T_{t,v,r}$, $\Theta T_{t,v,r}$). $\Xi T_{t,v,r}$ is a multivariate polynomial to represent an element in G_T and $\Theta G_{t,v,r}$ is the corresponding bit-string, where t, v and r have the same meanings in L_G. Initially, C stores two pairs (ΞMPK, $\Theta T_{I,0,1}$) and (ΞCPK, $\Theta T_{I,0,2}$) in L_{GT}, where $\Xi MPK=\Xi P \cdot \Xi MSK$ and $\Xi CPK=\Xi P \cdot \Xi CSK$.

Note that two transformation rules for L_G/L_{GT} are given below.

1 On receiving a polynomial $\Xi G_{t,v,r}/\Xi T_{t,v,r}$, C looks for $(\Xi G_{t,v,r}, \Theta G_{t,v,r})/(\Xi T_{t,v,r}, \Theta T_{t,v,r})$ in L_G/L_{GT}. If so, C returns the bit-string $\Theta G_{t,v,r}/\Theta T_{t,v,r}$. Otherwise, C randomly selects and returns a distinct bit-string $\Theta G_{t,v,r}/\Theta T_{t,v,r}$. Additionally, C adds $(\Xi G_{t,v,r}, \Theta G_{t,v,r})$ / $(\Xi T_{t,v,r}, \Theta T_{t,v,r})$ in L_G/L_{GT}.

2 On receiving an encoded bit-string $\Theta G_{t,v,r}/\Theta T_{t,v,r}$, C looks for $(\Xi G_{t,v,r}, \Theta G_{t,v,r})/(\Xi T_{t,v,r}, \Theta T_{t,v,r})$ in L_G/L_{GT}. If it is found, C returns the associated multivariate polynomial $\Xi G_{t,v,r}/\Xi T_{t,v,r}$. Otherwise, C terminates the game.

- L_{IKE} includes tuples of $(ID, \Xi IK_{ID}, \Xi Q_{ID})$, where ΞIK_{ID} and ΞQ_{ID}, respectively, denote the user's IK_{ID} and Q_{ID} in the *Identity key extract* algorithm.
- L_U includes tuples of $(ID, \Xi SK_{ID}, \Xi PK_{ID})$, where ΞSK_{ID} and ΞPK_{ID}, respectively, denote the user's SK_{ID} and PK_{ID} in the *Set secret key* algorithm.
- L_{TKE} includes tuples of $(ID, T_t, \Xi TK_{ID,t}, \Xi R_{ID,t})$, where $\Xi TK_{ID,t}$ and $\Xi R_{ID,t}$, respectively, denote the user's $TK_{ID,t}$ and $R_{ID,t}$ in the *Time update key extract* algorithm.

Finally, C sends these public parameters ΞP, ΞR, ΞS, ΞU, ΞV, ΞM, ΞN, ΞMPK and ΞCPK to A_I. Meanwhile, C sends the cloud secret key ΞCSK to A_I.

– *Query phase*: A_I can adaptively request various queries to C at most q times. Note that since A_I is permitted to get the secret key SK_{ID} and time update key $TK_{ID,t}$ of any user ID for any period T_t, A_I has no need to request the *Public key replace query* and *Time update key leak query*.

- Q_G *query* $(\Theta G_{Q,l,1}, \Theta G_{Q,l,2}, Operation)$: For the l-th Q_G *query*, C runs the following steps.
  1 Transform a pair of bit-strings $(\Theta G_{Q,l,1}, \Theta G_{Q,l,2})$ to get a pair of polynomials $(\Xi G_{Q,l,1}, \Xi G_{Q,l,2})$ in L_G.
  2 Compute the resulting polynomial $\Xi G_{Q,l,3}= \Xi G_{Q,l,1} + \Xi G_{Q,l,2}$ if $Operation$= "addition", and $\Xi G_{Q,l,3}= \Xi G_{Q,l,1}- \Xi G_{Q,l,2}$ if $Operation$="subtraction".
  3 Transform $\Xi G_{Q,l,3}$ to return the encoded bit-string $\Theta G_{Q,l,3}$.
- Q_T *query* $(\Theta T_{Q,l,1}, \Theta T_{Q,l,2}, Operation)$: For the l-th Q_T *query*, C runs the following steps.
  1 Transform a pair of bit-strings $(\Theta T_{Q,l,1}, \Theta T_{Q,l,2})$ to get a pair of polynomials $(\Xi T_{Q,l,1}, \Xi T_{Q,l,2})$ in L_{GT}.
  2 Compute the resulting polynomial $\Xi T_{Q,l,3}= \Xi T_{Q,l,1}+ \Xi T_{Q,l,2}$ if $Operation$="multiplication", and $\Xi T_{Q,l,3}= \Xi T_{Q,l,1}- \Xi T_{Q,l,2}$ if $Operation$="division".
  3 Transform $\Xi T_{Q,l,3}$ to return the encoded bit-string $\Theta T_{Q,l,3}$.
- Q_P *query* $(\Theta G_{P,l,1}, \Theta G_{P,l,2})$: For the l-th Q_P *query*, C runs the following steps.
  1 Transform a pair of bit-strings $(\Theta G_{P,l,1}, \Theta G_{P,l,2})$ to get a pair of polynomials $(\Xi G_{P,l,1}, \Xi G_{P,l,2})$.
  2 Compute the resulting polynomial $\Xi T_{P,l,1}= |\Xi G_{P,l,1}\cdot\Xi G_{P,l,2}$.
  3 Transform $\Xi T_{P,l,1}$ to return the encoded bit-string $\Theta T_{P,l,1}$.
- *Identity key query* (ID): For the i-th execution, C searches $(ID, \Xi IK_{ID}, \Xi Q_{ID})$ in L_{IKE}. If it is found, C transforms ΞIK_{ID} and ΞQ_{ID} to return two encoded bit-strings ΘIK_{ID} and ΘQ_{ID} to A_I. Otherwise, C adds a record in L_{IKE} as below.
  1 Choose a new variate $\Xi TG_{IK,i,1}$ in G.
  2 Set a polynomial $\Xi Q_{ID}= \Xi TG_{IK,i,1}$ and $\Xi TID=ID$.
  3 Compute the user's identity key $\Xi IK_{ID}= \Xi MSK+ \Xi TG_{IK,i,1}\cdot(\Xi R + \Xi S\cdot\Xi TID)$ while adding $(ID, \Xi IK_{ID}, \Xi Q_{ID})$ in L_{IKE}.
  4 Transform and return two encoded bit-strings ΘIK_{ID} and ΘQ_{ID} to A_I.
- *Identity key leak query* $(f_{IKE,i}, h_{IKE,i}, i)$: For the i-th *Identity key query*, A_I is permitted to issue this query to C only once. C returns two outputs $\Lambda f_{IKE,i}$ and $\Lambda h_{IKE,i}$ to A_I, where $\Lambda f_{IKE,i} = f_{IKE,i}(MSK_{i,1}, a_i, \alpha)$ and $\Lambda h_{IKE,i} = h_{IKE,i}(MSK_{i,2}, a_i, TI_{IKE})$.
- *Time update key query* (ID, T_t): For the j-th execution, C searches $(ID, T_t, \Xi TK_{ID,t}, \Xi R_{ID,t})$ in L_{TKE}. If it is found, C transforms $\Xi TK_{ID,t}$ and $\Xi R_{ID,t}$ to return two bit-strings $\Theta TK_{ID,t}$ and $\Theta R_{ID,t}$ to A_I. Otherwise, C adds a record in L_{TKE} as below.
  1 Choose a new variate $\Xi TG_{TK,ID,j,1}$ in G.
  2 Set a polynomial $\Xi R_{ID,t}= \Xi TG_{TK,ID,j,1}$ and $\Xi TTD = ID||T_t$.
  3 Set the user's time update key $\Xi TK_{ID,t}= \Xi CSK + \Xi TG_{TK,ID,j,1}\cdot(\Xi U+\Xi V\cdot\Xi TTD)$ while adding $(ID, \Xi SK_{ID}, \Xi PK_{ID})$ in L_{TKE}.

4 Transform and return two encoded bit-strings $\Theta TK_{ID,t}$ and $\Theta R_{ID,t}$ to A_I.

- *Time update key leak query* ($f_{TKE,j}$, $h_{TKE,j}$, j): For the j-th *Time update key query*, A_I is permitted to issue this query to C only once. C returns two leakage outputs $\Lambda f_{TKE,j}$ and $\Lambda h_{TKE,j}$ to A_{II}, where $\Lambda f_{TKE,j}=f_{TKE,j}(CSK_{j,1}, b_j, \beta)$ and $\Lambda h_{TKE,j}=h_{TKE,j}(CSK_{j,2}, b_j, TI_{TKE})$.
- *Public key retrieve query* (ID, T_t): C applies ID and T_t to search L_{KE}, L_U and L_{TKE} and then obtains the corresponding public key tuple (ΞQ_{ID}, $\Xi R_{ID,t}$, ΞPK_{ID}). C then transforms and returns a tuple of bit-strings (ΘQ_{ID}, $\Theta R_{ID,t}$, ΘPK_{ID}) to A_I.
- *Public key replace query* (ID, T_t, ($\Theta Q'_{ID}$, $\Theta R'_{ID,t}$, $\Theta PK'_{ID}$)): C first transforms a tuple of bit-strings ($\Theta Q'_{ID}$, $\Theta R'_{ID,t}$, $\Theta PK'_{ID}$) to obtain the corresponding tuple of polynomials ($\Xi Q'_{ID}$, $\Xi R'_{ID,t}$, $\Xi PK'_{ID}$). C replaces the related tuples with (ID, -, $\Xi Q'_{ID}$) in L_{KE}, (ID, -, $\Xi PK'_{ID}$) in L_U and (ID, T_t, -, $\Xi R_{ID,t}$) in L_{TKE}.
- *Secret key corrupt query* (ID): If *Public key replace query* (ID) is never issued, C uses ID to search (ID, ΞSK_{ID}, ΞPK_{ID}) in L_U. C transforms the secret key ΞSK_{ID} to return the bit-string ΘSK_{ID}. Otherwise, C runs the following steps.
  1 Choose a new variate $\Xi TG_{SK,ID,1}$ in G.
  2 Set two polynomials $\Xi SK_{ID}=\Xi TG_{SK,ID,1}$ and $\Xi PK_{ID}=\Xi P\cdot\Xi SK_{ID}$, and store ($ID$, ΞSK_{ID}, ΞPK_{ID}) in L_U.
  3 Transform ΞSK_{ID} and ΞPK_{ID} to obtain two encoded bit-strings ΘSK_{ID} and ΘPK_{ID}.
  4 Return the bit-string ΘSK_{ID} to A_I.
- *Singing query* (ID, T_t, msg): For the k-th execution of the user ID at period T_t, by taking as input msg, C runs the following steps.
  1 By ID, search (ID, ΞIK_{ID}, ΞQ_{ID}) in L_{IKE}.
  2 By ID, search (ID, ΞSK_{ID}, ΞPK_{ID}) in L_U.
  3 By ID and T_t, search (ID, T_t, $\Xi TK_{ID,t}$, $\Xi R_{ID,t}$) in L_{TKE}.
  4 Choose a new variate $\Xi TG_{S,k,1}$ in G and set $\Xi\sigma_1=\Xi TG_{S,k,1}$.
  5 Set $\Xi\sigma_2=\Xi IK_{ID}+\Xi TK_{ID,t}+\Xi SK_{ID}+\Xi TG_{S,k,1}\cdot(\Xi M+(ID||T_t||msg)\cdot\Xi N)$.
  6 Transform ($\Xi\sigma_1$, $\Xi\sigma_2$) to gain and return the encoded bit-strings ($\Theta\sigma_1$, $\Theta\sigma_2$) to A_I.
- *Signing leak query* (ID, T_t, $f_{SIG,k}$, $h_{SIG,k}$, k): For the k-th *Signing query* of the user ID at period T_t, by taking as input two leakage functions $f_{SIG,k}$ and $h_{SIG,k}$, C returns $\Lambda f_{SIG,k}$ and $\Lambda h_{SIG,k}$ to A_I, where $\Lambda f_{SIG,k}=f_{SIG,k}(IK_{ID,k,1}, SK_{ID,k,1}, c_k, d_k, \gamma)$ and $\Lambda h_{SIG,k}=h_{SIG,k}(IK_{ID,k,2}, SK_{ID,k,1}, c_k, d_k, TI_{SIG})$. Note that A_I is permitted to issue this query only once.

– *Forgery phase*: A_I outputs (ID^*, T_t^*, msg^*, ($\Theta\sigma^*_1$, $\Theta\sigma^*_2$)). A_I is not permitted to issue the *Signing query* (ID^*, T_t^*, msg^*) or *Identity key query* (ID^*). C transforms ($\Theta\sigma^*_1$, $\Theta\sigma^*_2$) to gain ($\Xi\sigma^*_1$, and $\Xi\sigma^*_2$), and sets $TID^*=ID^*$ and $TTD^*=ID^*||T_t^*$. If the equality $\Xi P\cdot\Xi\sigma^*_2=\Xi MPK+\Xi Q_{ID}\cdot(\Xi R+TID^*\cdot\Xi S)+\Xi CPK+\Xi R_{ID,t}\cdot(\Xi U+TTD^*\cdot\Xi V)+PK_{ID}+\Xi\sigma^*_1\cdot(\Xi M+(ID^*||T_t^*||msg^*)\cdot\Xi N)$ holds, we say that A_I wins the game.

In the following, let us evaluate the probability that A_I wins the game. Firstly, the amounts of group elements in L_G and L_{GT} are counted as given below:

1 In the *Setup phase*, 9 and 2 elements are, respectively, added in L_G and L_{GT}.
2 In the *Query phase*, the added amounts of L_G and L_{GT} for each query are discussed as follows.
- For each Q_G, Q_T or Q_P *query*, 3 elements could be added in L_G or L_{GT}.
- For each *Identity key query*, 2 elements could be added in L_G.
- For each *Time update key query*, 2 elements could be added in L_G.
- For each *Signing query*, 8 elements could be added in L_G.

Let q_O denote the total number of Q_G, Q_T and Q_P queries. Let q_{IK}, q_{TK} and q_S, respectively, be the query numbers of the *Identity key query*, *Time update key query* and *Signing query*. Since A_I is permitted to request all queries at most q times, we have $|L_G|+|L_{GT}|\leqq 11+3q_O+2q_{IK}+2q_{TK}+8q_S\leqq 8q$.

Secondly, let us evaluate the maximal degrees of polynomials in L_G and L_{GT}, respectively.

1 In L_G, the maximal degree of polynomials is 3 by the following discussions.
- In the *Setup* phase, nine new variates (polynomials) ΞP, ΞMSK, ΞCSK, ΞR, ΞS, ΞU, ΞV, ΞM and ΞN are initially added in L_G. All these polynomials have degree 1.
- For the Q_G *query*, $\Xi G_{Q,l,3}$ has the maximal degree of

$\Xi G_{Q,l,1}$ or $\Xi G_{Q,l,2}$.

- For the *Identity key query*, three polynomials $\Xi TG_{IK,i,1}$, ΞTID and ΞIK_{ID} have degrees 1, 1 and 3, respectively.
- For the *Time update key query*, three polynomials $\Xi TG_{TK,ID,j,1}$, ΞTTD and $\Xi TK_{ID,t}$ have degrees 1, 1 and 3, respectively.
- For the *Signing query*, two polynomials $\Xi\sigma_1$ and $\Xi\sigma_2$ have degrees 1 and 3, respectively.

2 In L_{GT}, the maximal degree of polynomials is 6 by the following discussions.

- In the *Setup* phase, two polynomials ΞMPK and ΞCPK have degree 2.
- For Q_T *query*, $\Xi T_{Q,l,3}$ has the maximal degree of $\Xi T_{Q,l,1}$ or $\Xi T_{Q,l,2}$.
- For Q_P *query*, since the maximal degree of polynomials in L_G is 3 and $\Xi T_{P,l,1}= \Xi G_{P,l,1}\cdot \Xi G_{P,l,2}$, the polynomial $\Xi T_{P,l,1}$ has degree 6.

Let us evaluate the advantage that A_I wins the game without requesting the *Identity key leak query* and *Signing leak query*. Subsequently, the advantage of A_I with requesting two kinds of leak queries is evaluated.

1 **The advantage of A_I without requesting two kinds of leak queries:** It is said that A_I wins the game if anyone of two cases occurs.

Case 1: A_I discovers a collision of any two elements in L_G or L_{GT}. Firstly, let us evaluate the collision probability in L_G. Let n denote the total number of all variates in L_G. The challenger C selects n random values $v_i \in Z_p^*$ for l=1, 2,...,n. Let ΞG_i and ΞG_j denote any two distinct polynomials in L_G. C then computes $\Xi G_C= \Xi G_i - \Xi G_j$ and $\Xi G_C(v_1, v_2, ..., v_n)$. If $\Xi G_C(v_1, v_2, ..., v_n)$=0, it is said that the collision occurs. By Lemma 2, the probability of $\Xi G_C(v_1, v_2, ..., v_n)$=0 is at most $3/p$ because the maximal polynomial degree in L_G is 3 and no fractional constituent (λ=0) is leaked. Since there are $\binom{|L_G|}{2}$ distinct pairs $(\Xi G_i, \Xi G_j)$ in L_G, the collision probability is $(3/p)\binom{|L_G|}{2}$. For the collision probability in L_{GT}, by similar evaluations, it is $(6/p)\binom{|L_{GT}|}{2}$. As mentioned earlier, we have $|L_G|+|L_{GT}| \leqq 8q$. Let the probability of *Case* 1 is denoted by Pr[Case 1], we have

$$\Pr[\text{Case 1}] \leqq (3/p)\binom{|L_G|}{2}+(6/p)\binom{|L_{GT}|}{2} \leqq (6/p)(|L_G|+|L_{GT}|)^2 \leqq 384q^2/p.$$

Case 2: Let us evaluate the probability that A_I outputs a valid signature $(ID^*, T_t^*, msg^*, (\Theta\sigma^*_1, \Theta\sigma^*_2))$ that satisfies $\Xi f = \Xi MPK + \Xi Q_{ID}\cdot(\Xi R+TID^*\cdot \Xi S)+ \Xi CPK + \Xi R_{ID,t}\cdot\Xi U + TTD^*\cdot \Xi V)+PK_{ID}+ \Xi\sigma^*_1\cdot(\Xi M+(ID^*||T_t^*||msg^*)\cdot\Xi N) - \Xi P\cdot\Xi\sigma^*_2=0$. Obviously, the degree of Ξf is at most 5. By Lemma 2, the probability is $5/p$. Let the probability of *Case* 2 is denoted by Pr[Case 2], we have $\Pr[\text{Case 2}] \leqq 5/p$.

Let $Adv_{AI\text{-}W}$ is the advantage that A_I wins the game without requesting two kinds of leak queries. By Pr[Case 1] and Pr[Case 2], we have

$$Adv_{AI\text{-}W} \leqq \Pr[\text{Case 1}]+\Pr[\text{Case 2}] \leqq 384q^2/p + 5/p = O(q^2/p).$$

Hence, $Adv_{AI\text{-}W}$ is negligible if $q=poly(\log p)$.

2 **The advantage of A_I with requesting two kinds of leak queries:** Firstly, let us discuss the fractional constituents of the private (or secret) keys involved in the associated leak queries.

1 *Identity key leak query* $(f_{IKE,i}, h_{IKE,i}, i)$: As mentioned earlier, we have the conditions $|f_{IKE,i}| \leq \lambda$ and $|h_{IKE,i}| \leq \lambda$. By this query, A_I derives fractional constituents $\Lambda f_{IKE,i}=f_{IKE,i}(MSK_{i,1}, a_i, \alpha)$ and $\Lambda h_{IKE,i}=h_{IKE,i}(MSK_{i,2}, a_i, TI_{IKE})$ that are discussed as below.

- a_i, α: Since a_i and α are randomly selected in each *Identity key query*, the leakage information of a_i or α is no help to learn the master secret key MSK.
- $(MSK_{i,1}, MSK_{i,2})$: Indeed, the master secret key MSK satisfies $MSK = MSK_{0,1}+ MSK_{0,2}= MSK_{1,1} + MSK_{1,2} =...= MSK_{i,1} + MSK_{i,2}$. By the blinding technique, fractional constituent of $MSK_{i-1,1}/MSK_{i-1,2}$ is independent of that of $MSK_{i,1}/MSK_{i,2}$. Hence, A_I derives at most 2λ bits of MSK.
- TI_{IKE}: TI_{IKE} is a temporary value and applied to compute the user's identity key IK_{ID}. Since A_I can obtain the whole IK_{ID} except for ID^*, TI_{IKE} is helpless for A_I.

2 *Signing leak query* $(ID^*, T_t, f_{SIG,k}, h_{SIG,k}, k)$: As mentioned earlier, we have the conditions $|f_{SIG,k}| \leq \lambda$ and $|h_{SIG,k}| \leq \lambda$. And A_I is permitted to get the secret key SK_{ID} and time update key $TK_{ID,t}$ of any user ID for any period T_t. By this query, A_I derives fractional constituents $\Lambda f_{SIG,k}=$

$f_{SIG,k}(IK_{ID^*,k,1}, c_k, d_k, \gamma)$ and $\Lambda h_{SIG,k}=h_{SIG,k}(IK_{ID^*,k,2}, c_k, d_k, TI_{SIG})$ that are discussed as below.

- c_k, d_k, γ: Since c_k, d_k and γ are randomly selected in each *signing query*, their leakages are no help to learn the master secret key IK_{ID^*}.
- $(IK_{ID^*,k,1}, IK_{ID^*,k-1,2})$: Indeed, the identity key IK_{ID^*} satisfies $IK_{ID^*}= IK_{ID^*,0,1}+IK_{ID^*,0,2}=IK_{ID^*,1,1}+IK_{ID^*,1,2}=...=IK_{ID^*,k,1}+IK_{ID^*,k,2}$. By the blinding technique, fractional constituent of $IK_{ID^*,k-1,1}/IK_{ID^*,k-1,2}$ is independent of that of $IK_{ID^*,k,1}/IK_{ID^*,k,2}$. Hence, A_I derives at most 2λ bits of IK_{ID^*}.
- TI_{SIG}: TI_{SIG} is a temporary value and applied to compute the signature σ_2. Since A_I can obtain the entire σ_2 by the *Sign query*, TI_{SIG} is helpless for A_I.

Let Adv_{AI} be the advantage that A_I wins the game with requesting the *Identity key leak query* and *Signing leak query*. If A_I can know the master secret key MSK or the target user's identity key IK_{ID^*}, A_I may forge a legal signature. Two events are defined as below.

1 Let the event $EMSK$ denote that A_I knows the whole MSK by $\Lambda f_{IKE,i}$ and $\Lambda h_{IKE,i}$ while $\overline{EMSK}$ is the corresponding complement event.

2 Let the event EIK denote that A_I knows the whole IK_{ID^*} by $\Lambda f_{SIG,k}$ and $\Lambda h_{SIG,k}$ while $\overline{EIK}$ is the corresponding complement event.

Let the event EFS denote that A_I can forge a legal signature. Hence, the advantage Adv_{AI} is Pr[EFS] such that the following inequality

$$\begin{aligned}
Adv_{AI} &= \Pr[EFS] \\
&= \Pr[EFS\wedge(EMSK\vee EIK)] \\
&\quad +\Pr[EFS\wedge(\overline{EMSK}\wedge\overline{EIK})] \\
&=\Pr[EFS\wedge EMSK]+\Pr[EFS\wedge EIK] \\
&\quad +\Pr[EFS\wedge\overline{EMSK}\wedge\overline{EIK}] \\
&\leqq \Pr[EMSK]+\Pr[EFS\wedge EIK] \\
&\quad +\Pr[EFS\wedge\overline{EMSK}\wedge\overline{EIK}] \\
&\leqq \Pr[EMSK]+\Pr[EIK] \\
&\quad +\Pr[EFS\wedge\overline{EMSK}\wedge\overline{EIK}].
\end{aligned}$$

In *Case* 1 of A_I without requesting two kinds of leak queries, the advantage is Pr[Case 1] $\leqq 384q^2/p = O(q^2/p)$. By the *Identity key leak query*, A_I derives at most 2λ bits of MSK. By Lemma 2, we have $\Pr[EMSK]\leqq O((q^2/p)^*2^{2\lambda})$. By the similar reason, we have $\Pr[EIK]\leqq O((q^2/p)^*2^{2\lambda})$. Finally, the event $\overline{EMSK}\wedge\overline{EIK}$ is that A_I can get fractional constituents of MSK and IK_{ID^*}. Since $Adv_{AI\text{-}W}\leqq O(q^2/p)$ and A_I can gain at most 2λ bits about MSK and IK_{ID^*}, we have $\Pr[EFS\wedge\overline{EMSK}\wedge\overline{EIK}]\leqq O((q^2/p)^*2^{2\lambda})$. According to the discussions above, we have

$$\begin{aligned}
Adv_{AI} &= \Pr[EFS] \\
&\leqq \Pr[EMSK]+\Pr[EIK]+\Pr[EFS\wedge\overline{EMSK}\wedge\overline{EIK}] \\
&\leqq O((q^2/p)^*2^{2\lambda}).
\end{aligned}$$

By Corollary 1, Adv_{AI} is negligible if $\lambda<\log p-\omega(\log\log p)$. Q.E.D.

Theorem 2. In the GBG model, our LR-ORCLS scheme possesses existential unforgeability against Type II adversary (A_{II}, revoked user) in the UF-LR-ORCLS-ACMA game.

Proof. Let A_{II} be of Type II adversary in the UF-LR-ORCLS-ACMA game played with a challenger C. A_{II} may issue various queries to C at most q times in the game. This game consists of three phases as follows:

– *Setup Phase*: The phase is the same with that of the proof in Theorem 1. C sends the public parameters $\Xi P, \Xi R, \Xi S, \Xi U, \Xi V, \Xi M, \Xi N, \Xi MPK$ and ΞCPK to A_I. Additionally, C also sends the master secret key ΞMSK to A_{II}.

– *Query phase*: In this phase, A_{II} can adaptively issue various queries to C at most q times. All queries are identical to those queries in Theorem 1. Note that since A_{II} is permitted to get both the identity key IK_{ID} and secret key USK_{ID} of any user ID, A_{II} has no need to issue the *Identity key leak query* and *Public key replace query*. Indeed, a revoked user's time update key $UTK_{ID,t}$ is never generated so that the *Signing leak query* does not leak any content. Additionally, A_{II} can derive fractional constituents of the cloud secret key CSK by the *Time update key leak query*.

– *Forgery phase*: A_{II} outputs $(ID^*, T_t^*, msg^*, (\Theta\sigma^*_1, \Theta\sigma^*_2))$. A_{II} is not permitted to issue the *Signing query* (ID^*, T_t^*, msg^*) or *Time update key query* (ID^*, T_t^*). C transforms $(\Theta\sigma^*_1, \Theta\sigma^*_2)$ to gain $(\Xi\sigma^*_1$, and $\Xi\sigma^*_2)$, and sets $TID^*=ID^*$ and $TTD^*=ID^*||T_t^*$. If the equality $\Xi P\cdot\Xi\sigma^*_2= \Xi MPK+\Xi Q_{ID}\cdot(\Xi R + TID^*\cdot\Xi S) + \Xi CPK + \Xi R_{ID,t}\cdot(\Xi U +TTD^*\cdot\Xi V)+PK_{ID}+\Xi\sigma^*_1\cdot(\Xi M+(ID^*||T_t^*||msg^*)\cdot\Xi N)$ holds, we say that A_{II} wins the game.

In the following, let us evaluate the probability that A_{II} wins the game. Let us first evaluate the advantage that A_{II} wins the game without requesting the *Time update key leak query*. Subsequently, the advantage of A_{II} with requesting the *Time update key leak query* is evaluated.

1 **The advantage of A_{II} without requesting the *Time update key leak query*:** Let $Adv_{AII\text{-}W}$ be the advantage that A_{II} wins the game without requesting the *Time update key leak query*. As the similar evaluations in Theorem 1, we have $Adv_{AII\text{-}W} = O(q^2/p)$.

2 **The advantage of A_{II} with requesting the *Time update key leak query*:** For the j-th *Time update key leak query* with $f_{TKE,j}$ and $h_{TKE,j}$ such that $|f_{TKE,j}| \leq \lambda$ and $|h_{TKE,j}| \leq \lambda$, A_{II} can gain fractional constituents $\Lambda f_{TKE,j} = f_{TKE,j}$ $(CSK_{j,1}, b_j, \beta)$ and $\Lambda h_{TKE,j} = h_{TKE,j}(CSK_{j,2}, b_j, TI_{TKE})$. Indeed, the cloud secret key CSK satisfies $CSK = CSK_{0,1} + CSK_{0,2} = CSK_{1,1} + CSK_{1,2} = \ldots = CSK_{j,1} + CSK_{j,2}$. By the blinding technique, fractional constituent of $CSK_{j-1,1}/CSK_{j-1,2}$ is independent of that of $CSK_{j,1}/CSK_{j,2}$. In such a case, A_{II} derives at most 2λ bits of CSK.

Let Adv_{AII} be the advantage that A_{II} wins the game with requesting the *Time update key leak query*. If A_{II} knows the whole cloud secret key CSK, A_{II} can get the time update key TK_{ID^*,t^*} of the target user ID^* at period T_{t^*}. Thus, A_{II} may forge a legal signature. Let the event $ECSK$ denote that A_{II} knows the whole CSK by $f_{TKE,j}$ and $h_{TKE,j}$ while $\overline{ECSK}$ is the corresponding complement event. Let the event EFS denote that A_{II} can forge a legal signature. Hence, we have $Adv_{AII} = \Pr[EFS]$ that satisfies the following inequality

$$\begin{aligned} Adv_{AII} &= \Pr[EFS] \\ &= \Pr[EFS \wedge ECSK] + \Pr[EFS \wedge \overline{ECSK}] \\ &\leqq \Pr[ECSK] + \Pr[EFS \wedge \overline{ECSK}]. \end{aligned}$$

By the *Time update key leak query*, A_{II} derives at most 2λ bits of CSK. By Pr[Case 1] $\leqq O((q^2/p)$ in Theorem 1 and Lemma 2, we have $\Pr[ECSK] \leqq O((q^2/p)^* 2^{2\lambda})$. Finally, the event $\overline{ECSK}$ is that A_{II} can get fractional constituents of $(CSK_{j,1}, CSK_{j,2})$ by $\Lambda f_{TKE,j}$ and $\Lambda h_{TKE,j}$. Since $Adv_{AII\text{-}W} \leqq O(q^2/p)$ and A_{II} can gain at most 2λ bits about CSK, we have $\Pr[EFS \wedge \overline{ECSK}] \leqq O((q^2/p)^* 2^{2\lambda})$. According to the discussions above, we have

$$Adv_{AII} \leqq \Pr[ECSK] + \Pr[EFS \wedge \overline{ECSK}] \leqq O((q^2/p)^* 2^{2\lambda}).$$

By Corollary 1, Adv_{AII} is negligible if $\lambda < \log p - \omega(\log\log p)$. Q.E.D.

Theorem 3. In the GBG model, our LR-ORCLS scheme is existential unforgeable against Type III adversary (A_{III}, honest-but-curious *KGC*) in the UF-LR-ORCLS-ACMA game.

Proof. Let A_{III} be of Type III adversary in the UF-LR-ORCLS-ACMA game played with a challenger C. A_{III} may issue various queries to the challenger C at most q times in the game. This game consists of three phases as follows:

- *Setup Phase*: The phase is the same with that of the proof in Theorem 1. C sends public parameters $\Xi P, \Xi R, \Xi S, \Xi U, \Xi V, \Xi M, \Xi N, \Xi MPK$ and ΞCPK to A_I. Additionally, C also sends the master secret key ΞMSK and the cloud secret key ΞCSK to A_{III}.
- *Query phase*: In this phase, A_{III} can adaptively issue various queries to C at most q times. Note that since A_{III} is permitted to get the identity key IK_{ID} and time update key $TK_{ID,t}$ of any user ID for any period T_t, A_{III} has no need to issue the *Identity key leak query* and *Time update key leak query*. Additionally, A_{III} is permitted to get the secret key SK_{ID} of any user ID, except SK_{ID^*} of the attacking target user ID^*. Meanwhile, A_{III} can derive fractional constituent of the secret key SK_{ID^*} by the *Signing leak query*.
- *Forgery phase*: A_{III} outputs $(ID^*, T_t^*, msg^*, (\Theta\sigma^*_1, \Theta\sigma^*_2))$. A_{III} is not permitted to issue the *Signing query* (ID^*, T_t^*, msg^*), *Public key replace query* (ID^*, T_t^*) or *Secret key corrupt query* (ID^*). C transforms $(\Theta\sigma^*_1, \Theta\sigma^*_2)$ to gain $(\Xi\sigma^*_1$, and $\Xi\sigma^*_2)$, and sets $TID^* = ID^*$ and $TTD^* = ID^* || T_t^*$. If the equality $\Xi P \cdot \Xi\sigma^*_2 = \Xi MPK + \Xi Q_{ID} \cdot (\Xi R + TID^* \cdot \Xi S) + \Xi CPK + \Xi R_{ID,t} \cdot (\Xi U + TTD^* \cdot \Xi V) + PK_{ID} + \Xi\sigma^*_1 \cdot (\Xi M + (ID^* || T_t^* || msg^*) \cdot \Xi N)$ holds, it is said that A_{III} wins the game.

In the following, let us evaluate the probability that A_{III} wins the game. Let us evaluate the advantage that A_{III} wins the game without requesting the *Signing leak query*. Subsequently, the advantage of A_{III} with requesting the *Signing leak query* is evaluated.

1 **The advantage of A_{III} without requesting the *Signing leak query:*** Let $Adv_{AIII\text{-}W}$ is the advantage

that A_{III} wins the game without requesting the *Signing leak query*. As the similar evaluations in Theorem 1, we have $Adv_{AIII\text{-}W}=O(q^2/p)$.

2 **The advantage of A_{III} with requesting the *Signing leak query:*** For the k-th *Signing leak query* with $f_{SIG,k}$ and $h_{SIG,k}$ such that $|f_{SIG,k}|\leq\lambda$ and $|h_{SIG,k}|\leq\lambda$, A_{III} can get fractional constituents $\Lambda f_{SIG,k}=f_{SIG,k}(SK_{ID^*,k,1}, c_k, d_k, \gamma)$ and $\Lambda h_{SIG,k}=h_{SIG,k}(SK_{ID^*,k,1}, c_k, d_k, TI_{SIG})$. Indeed, the user's secret key SK_{ID^*} satisfies $SK_{ID^*}=SK_{ID^*,0,1}+SK_{ID^*,0,2}=SK_{ID^*,1,1}+SK_{ID^*,1,2}=\ldots=SK_{ID^*,k,1}+SK_{ID^*,k,2}$. By the blinding technique, fractional constituent of $SK_{ID^*,k\text{-}1,1}/SK_{ID^*,k\text{-}1,2}$ is independent of that of $SK_{ID^*,k,1}/SK_{ID^*,k,2}$. In such a case, A_{III} derives at most 2λ bits of SK_{ID^*}.

Let Adv_{AIII} be the advantage that A_{III} wins the game with requesting the *Signing leak query*. If A_{III} knows the secret key SK_{ID^*}, A_{III} can forge a legal signature. Let the event ESK denote that A_{III} knows the whole SK_{ID^*} while $\overline{ESK}$ is the corresponding complement event. Let the event EFS denote that A_{III} can forge a legal signature. Hence, we have $Adv_{AIII}=\Pr[EFS]$ that satisfies the following inequality

$$\begin{aligned} Adv_{AIII} &= \Pr[EFS] \\ &= \Pr[EFS\wedge ESK]+\Pr[EFS\wedge\overline{ESK}] \\ &\leqq \Pr[ESK]+\Pr[EFS\wedge\overline{ESK}]. \end{aligned}$$

By the *Signing leak query*, A_{III} derives at most 2λ bits of SK_{ID^*}. By Pr[Case 1] $\leqq O((q^2/p)$ in Theorem 1 and Lemma 2, we have $\Pr[ESK]\leqq O((q^2/p)^*2^{2\lambda})$. Finally, the event $\overline{ESK}$ is that A_{III} can get fractional constituents of $(SK_{ID^*,k,1}, SK_{ID^*,k,2})$ by $\Lambda f_{TKE,j}$ and $\Lambda h_{TKE,j}$. Since $Adv_{AIII\text{-}W}\leqq O(q^2/p)$ and A_{III} can gain at most 2λ bits about SK_{ID^*}, we also have $\Pr[EFS\wedge\overline{ESK}]\leqq O((q^2/p)^*2^{2\lambda})$. According to the discussions above, we have

$$\begin{aligned} Adv_{AIII} &= \Pr[EFS]\leqq \Pr[ESK]+\Pr[EFS\wedge\overline{ESK}] \\ &\leqq O((q^2/p)^*2^{2\lambda}). \end{aligned}$$

By Corollary 1, Adv_{AIII} is negligible if $\lambda<\log p-\omega(\log\log p)$. Q.E.D.

6. Comparisons

In this section, the comparisons between several previous RCLS and ORCLS schemes [7, 12, 21] and our LR-ORCLS scheme are given. Firstly, let us define several computation notations. By the simulation experiences in [10], the corresponding computational costs (in millisecond) are given in Table 2. Note that we omit the computational costs of both the addition on G and the multiplication on G_T because they are small and negligible. For the simulation experiences in [10], the platform is equipped with a 3-GHz Pentium processor while running under a Microsoft window operation system. The security of the simulation results adopts 1024-bit RSA security level to measure the computational costs.

Table 2
Computational costs (in millisecond) of several operations

Notations	Operations	Computational costs
T_{bp}	a bilinear pairing $\hat{e}$: $G'G^{\circledR}G_T$	20.01 ms
T_{sm}	a scalar multiplication on G	6.38 ms
T_{mp}	a map-to-point hash function on G	3.04 ms
T_{cm}	a scalar multiplication on an elliptic curve group G_{ECC}	0.83 ms

Table 3 demonstrates the comparisons between our LR-ORCLS scheme and several RCLS and ORCLS schemes [7, 12, 21] in terms of signing cost (ms), verifying cost (ms), outsourced revocation, resisting side-channel attacks and overall leakage property. To provide leakage-resilient property (i.e., resisting side-channel attacks), our scheme requires some extra computation costs. It is obvious that the performance of our scheme is worse than the previously proposed RCLS and ORCLS schemes. Both Du *et al.*'s scheme and ours apply a CRS to offer outsourced revocation functionality to reduce the computational burden of the KGC. We emphasize that our scheme is the first LR-ORCLS scheme resistant against side-channel attacks while possessing overall unbounded leakage property. It is worth mentioning that the proposed scheme is not suited for unsuitable for some environments with resource-constrained devices (i.e. IoT devices) because it requires time-consuming bilinear pairing operations [33].

Table 3
Comparisons between our scheme and several previous RCLS or ORCLS schemes

	Sun *et al.*'s RCLS scheme [21]	Hung *et al.*'s RCLS scheme [12]	Du *et al.*'s ORCLS scheme [7]	Our LR-ORCLS scheme
Signing cost (ms)	$2T_{sm}+2T_{mp}$ (18.84ms)	$2T_{sm}+2T_{mp}$ (18.84ms)	T_{cm} (0.83ms)	$5T_{sm}$ (31.9ms)
Verifying cost (ms)	$3T_{bp}+2T_{mp}$ (66.11ms)	$4T_{bp}+T_{sm}+3T_{mp}$ (95.9ms)	$5T_{cm}$ (4.15ms)	$4T_{bp}+3T_{sm}$ (99.16ms)
Outsourced revocation	No	No	Yes	Yes
Resisting side-channel attacks	No	No	No	Yes
Overall leakage property	Not provided	Not provided	Not provided	Unbounded

7. Conclusions and Future Work

In this article, the first LR-ORCLS scheme has been proposed. As compared to previous RCLS and ORCLS schemes, our scheme has the following merits: (1) The revocation functionality is outsourced to the CRS to reduce the computational burden of the KGC; (2) It can resist side-channel attacks and permits adversaries to continually derive fractional constituents of private (or secret) keys; (3) It possesses the overall unbounded leakage property. Meanwhile, the novel adversary model was defined. By extending the adversary model of the ORCLS scheme, three kinds of leak queries are added, namely, *Identity key leak query*, *Time update key leak query* and *Signing leak query*. By three kinds of leak queries, adversaries are permitted to continually derive fractional constituents of the KGC's master secret key, the CRS's cloud secret key and a signer's secret key involved in the associated algorithms. In the GBG model, the security of the proposed scheme is shown to be existential unforgeable against Types I, II and III adversaries. By the comparisons mentioned in Table 3, indeed, our protocol still requires bilinear pairing operations and its performance is worse than the previous RCLS and ORCLS scheme. Hence, our protocol is unsuitable for some environments with resource-constrained devices (i.e. IoT devices). In the future, it is interesting to propose a lightweight LR-ORCLS protocol without requiring bilinear pairing operations.

Acknowledgement

This research was partially supported by Ministry of Science and Technology, Taiwan, under contract no. MOST108-2221-E-018-04-MY2.

References

1. Al-Riyami, S. S., Paterson, K. G. Certificateless Public Key Cryptography. In: ASIACRYPT'03, LNCS, 2894, Springer, Belin-Heidelberg, 2003, 452-473. https://doi.org/10.1007/978-3-540-40061-5_29
2. Alwen, J., Dodis, Y., Wichs, D. Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model. In: CRYPTO'09, LNCS, 5677, Springer, Belin-Heidelberg, 2009, 36-54. https://doi.org/10.1007/978-3-642-03356-8_3
3. Boneh, D., Boyen, X., Goh, E. J. Hierarchical Identity-Based Encryption with Constant Size Ciphertext. In: EUROCRYPT'05, LNCS, 3494, Spring-

er, Belin-Heidelberg, 2005, 440-456. https://doi.org/10.1007/11426639_26

4. Boneh, D., Franklin, M. Identity-Based Encryption from the Weil Pairing. In: CRYPTO'01, LNCS, 2139, Springer, Belin-Heidelberg, 2001, 213-229. https://doi.org/10.1007/3-540-44647-8_13
5. Brumley, D., Boneh, D. Remote Timing Attacks Are Practical. Computer Networks, 2005, 48(5), 701-716. https://doi.org/10.1016/j.comnet.2005.01.010
6. Dodis, Y., Haralambiev, K., Lopez-Alt, A., Wichs, D. Efficient Public-Key Cryptography in the Presence of Key Leakage. In: ASIACRYPT'10, LNCS, 6477, Springer, Belin-Heidelberg, 2010, 613-631. https://doi.org/10.1007/978-3-642-17373-8_35
7. Du, H., Wen, Q., Zhang, S. A Provably-Secure Outsourced Revocable Certificateless Signature Scheme without Bilinear Pairings. IEEE Access, 2018, 6, 73846-73855. https://doi.org/10.1109/ACCESS.2018.2880875
8. ElGamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory, 1985, 31(4), 469-472. https://doi.org/10.1109/TIT.1985.1057074
9. Galindo, D., Grobschadl, J., Liu, Z., Vadnala, P. K., Vivek, S. Implementation of a Leakage-Resilient ElGamal Key Encapsulation Mechanism. Journal of Cryptographic Engineering, 2016, 6(3), 229-238. https://doi.org/10.1007/s13389-016-0121-x
10. He, D., Chen, Y., Chen, J. An Efficient Certifcateless Proxy Signature Scheme without Pairing. Mathematical and Computer Modelling, 2013, 57, 2510-2518. https://doi.org/10.1016/j.mcm.2012.12.037
11. Housley, R., Polk, W., Ford, W., Solo, D. Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF, RFC 3280, 2002. https://doi.org/10.17487/rfc3280
12. Hung, Y. H., Tseng, Y. M., Huang, S. S. A Revocable Certificateless Short Signature Scheme and Its Authentication Application. Informatica, 2016, 27(3), 549-572. https://doi.org/10.15388/Informatica.2016.99
13. Katz, J., Vaikuntanathan, V. Signature Schemes with Bounded Leakage Resilience. In: ASIACRYPT'09, LNCS, 5912, Springer, Belin-Heidelberg, 2009, 703-720. https://doi.org/10.1007/978-3-642-10366-7_41
14. Kocher, P. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: CRYPTO'96, LNCS, 1163, Springer, Belin-Heidelberg, 1996, 104-113. https://doi.org/10.1007/3-540-68697-5_9
15. Kocher, P., Jaffe, J., Jun, B. Differential Power Analysis. In: CRYPTO'99, LNCS, 1666, Springer, Belin-Heidelberg, 1999, 388-397. https://doi.org/10.1007/3-540-48405-1_25
16. Rivest, R. L., Shamir, A., Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of ACM, 1978, 21(2), 120-126. https://doi.org/10.1145/359340.359342
17. Scott, M. On the Efficient Implementation of Pairing-Based Protocols. In: Cryptography and Coding, LNCS, 7089, Springer, Belin-Heidelberg, 2011, 296-308. https://doi.org/10.1007/978-3-642-25516-8_18
18. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In: CRYPTO'84, LNCS, 196, Springer, Belin-Heidelberg, 1984, 47-53. https://doi.org/10.1007/3-540-39568-7_5
19. Shen, L., Zhang, F., Sun, Y. Efficient Revocable Certificateless Encryption Secure in the Standard Model. Computer Journal, 2014, 57(4), 592-601. https://doi.org/10.1093/comjnl/bxt040
20. Shoup, V. Lower Bounds for Discrete Logarithms and Related Problems. In: EUROCRYPT'97, LNCS, 1233, Springer, Belin-Heidelberg, 1997, 256-266. https://doi.org/10.1007/3-540-69053-0_18
21. Sun, Y., Zhang, F., Shen, L. A Revocable Certificateless Signature Scheme. Journal of Computer, 2014, 9(8), 1843-1850. https://doi.org/10.4304/jcp.9.8.1843-1850
22. Tang, F., Li, H., Niu, Q., Liang, B. Efficient Leakage-Resilient Signature Schemes in the Generic Bilinear Group Model. In: Information Security Practice and Experience, LNCS, 8434, Springer, Belin-Heidelberg, 2014, 418-432. https://doi.org/10.1007/978-3-319-06320-1_31
23. Tsai, T. T., Tseng, Y. M. Revocable Certificateless Public Key Encryption. IEEE Systems Journal, 2015, 9(3), 824-833. https://doi.org/10.1109/JSYST.2013.2289271
24. Tsai, T. T., Huang, S. S., Tseng, Y. M. Secure Certificateless Signature with Revocation in the Standard Model. Mathematical Problems in Engineering, 2014, Article ID 728591. https://doi.org/10.1155/2014/728591
25. Tseng, Y. M., Tsai, T. T. Efficient Revocable ID-Based Encryption with a Public Channel. Computer Journal, 2012, 55(4), 475-486. https://doi.org/10.1093/comjnl/bxr098
26. Wu, J. D., Tseng, Y. M., Huang, S. S. Leakage-Resilient ID-Based Signature Scheme in the Generic Bilinear Group Model. Security and Communication Networks, 2016, 9(17), 3987-4001. https://doi.org/10.1002/sec.1580

27. Wu, J. D., Tseng, Y. M., Huang, S. S. Leakage-Resilient Certificateless Signature under Continual Leakage Model. Information Technology and Control, 2018, 47(2), 363-386. https://doi.org/10.5755/j01.itc.47.2.17847

28. Wu, J. D., Tseng, Y. M., Huang, S. S. Efficient Leakage-Resilient Authenticated Key Agreement Protocol in the Continual Leakage eCK Model. IEEE Access, 2018, 6(1), 17130-17142. https://doi.org/10.1109/ACCESS.2018.2799298

29. Wu, J. D., Tseng, Y. M., Huang, S. S. An Identity-Based Authenticated Key Exchange Protocol Resilient to Continuous Key Leakage. IEEE Systems Journal, 2019, 13(4), 3968-3979. https://doi.org/10.1109/JSYST.2019.2896132

30. Wu, J. D., Tseng, Y. M., Huang, S. S., Tsai, T. T. Leakage-Resilient Certificate-Based Signature Resistant to Side-Channel Attacks. IEEE Access, 2019, 7(1), 19041-19053. https://doi.org/10.1109/ACCESS.2019.2896773

31. Xiong, H., Yuen, T. H., Zhang, C., Yiu, S. M., He, Y. J. Leakage-Resilient Certificateless Public Key Encryption. In: Proceedings of the first ACM workshop on Asia public-key cryptography, 2013, 13-22. https://doi.org/10.1145/2484389.2484394

32. Zhou, Y., Yang, B., Zhang, W. Provably Secure and Efficient Leakage-Resilient Certificateless Signcryption Scheme without Bilinear Pairing. Discrete Applied Mathematics, 2016, 204, 185-202. https://doi.org/10.1016/j.dam.2015.10.018

33. Zhou, L., Su, C., Yeh, K. A Lightweight Cryptographic Protocol with Certificateless Signature for the Internet of Things. ACM Transactions on Embedded Computing Systems, 2019, 18(3), Article 28. https://doi.org/10.1145/3301306