Leakage-Resilient Revocable Identity-Based Signature with Cloud Revocation Authority

Jui-Di WU, Yuh-Min TSENG*, Sen-Shan HUANG, Tung-Tso TSAI
Department of Mathematics, National Changhua University of Education, Jin-De Campus, Chang-Hua City 500, Taiwan
e-mail: ymtseng@cc.ncue.edu.tw

Received: October 2019; accepted: February 2020

Abstract. Very recently, side-channel attacks have threatened all traditional cryptographic schemes. Typically, in traditional cryptography, private/secret keys are assumed to be completely hidden to adversaries. However, by side-channel attacks, an adversary may extract fractional content of these private/secret keys. To resist side-channel attacks, leakage-resilient cryptography is a countermeasure. Identity-based public-key system (ID-PKS) is an attractive public-key setting. ID-PKS settings not only discard the certificate requirement, but also remove the construction of the public-key infrastructure. For solving the user revocation problem in ID-PKS settings, revocable ID-PKS (RID-PKS) setting has attracted significant attention. Numerous cryptographic schemes based on RID-PKS settings have been proposed. However, under RID-PKS settings, no leakage-resilient signature or encryption scheme is proposed. In this article, we present the first leakage-resilient revocable ID-based signature (LR-RIBS) scheme with cloud revocation authority (CRA) under the continual leakage model. Also, a new adversary model of LR-RIBS schemes with CRA is defined. Under this new adversary model, security analysis is made to demonstrate that our LR-RIBS scheme with CRA is provably secure in the generic bilinear group (GBG) model. Finally, performance analysis is made to demonstrate that our scheme is suitable for mobile devices.

Key words: ID-based signature, leakage resilience, revocation, side-channel attack.

1. Introduction

Identity-based public-key system (ID-PKS) (Shamir, 1984; Boneh and Franklin, 2001) not only discards the certificate requirement, but also removes the construction of the public-key infrastructure. In an ID-PKS setting, there are two roles, namely, users and a private key generator (PKG). A user’s identity information is regarded as the user’s public key. The PKG employs the user’s identity information to generate the user’s associated private key. For public-key settings, user revocation mechanisms are required to revoke the misbehaving or compromised users before the intended expiration date of their public keys. Typically, a conventional public-key setting adopts the certificate revocation list (CRL) (Housley et al., 2002) to manage the revoked users. In such a setting, each user has a public key and the associated certificate. Before employing a user’s public key, one must
validate its associated certificate while looking up the CRL to ensure that the user’s certificate was not revoked. However, ID-PKS settings do not require the usage of certificates so that the CRL mechanism cannot be employed to the ID-PKS settings.

Recently, Tseng and Tsai (2012) proposed a revocable ID-PKS (RID-PKS) setting with a public channel. In the RID-PKS setting, a user’s private key includes two parts, namely, a secret key and a time update key. Initially, the PKG employs a user’s identity information to generate and send the associated secret key to the user using a secure channel. Also, the PKG generates the time update key by time period and the user’s identity information. Namely, for all non-revoked users, the PKG periodically generates and sends the associated time update keys to these users using a public channel. Subsequently, numerous cryptographic primitives based on RID-PKS settings were presented, such as revocable ID-based encryption (RIBE) (Tsai et al., 2012, 2013a) and revocable ID-based signature (RIBS) schemes (Tsai et al., 2013b; Hung et al., 2017). Furthermore, several RIBE and RIBS schemes (Lì et al., 2015; Tseng et al., 2018; Jia et al., 2017) have been proposed to outsource the periodical generations of time update keys to a cloud revocation authority (CRA).

Quite recently, side-channel attacks have threatened all traditional cryptographic schemes because private/secret keys are assumed to be completely hidden to adversaries in traditional cryptography. By various kinds of side-channel attacks (Boneh et al., 1997; Kocher et al., 1999; Brumley and Boneh, 2005; Biham et al., 2008), an adversary can extract fractional content of private/secret keys participated in computation rounds. To resist side-channel attacks, leakage-resilient cryptography is a countermeasure while the design of leakage-resilient cryptographic schemes has attracted significant attention from researchers. For leakage-resilient cryptographic schemes, adversaries are allowed to extract fractional content of private/secret keys while these schemes still retain secure. However, no leakage-resilient RIBS scheme based on RID-PKS settings is proposed. In the article, our goal is to propose the first leakage-resilient RIBS (LR-RIBS) scheme.

1.1. Related Work

Here, let us briefly review some leakage-resilient encryption and signature schemes based on conventional and ID-PKS settings.

According to the amount of leaked content of private/secret keys during the life time, the leakage model has two kinds, namely, bounded leakage model (Alwen et al., 2009) and continual leakage model (Brakerski et al., 2010). In a leakage-resilient cryptographic scheme under the bounded leakage model, the overall amount of leaked content has to be limited to a ratio or a fixed bit-length of private/secret keys. On the contrary, a leakage-resilient cryptographic scheme under the continual leakage model allows adversaries to continuously extract fractional content of private/secret keys so that its overall amount of leaked content is unlimited. For security robustness, a cryptographic scheme under the continual leakage model is stronger than that under the bounded leakage model. The properties of continual leakage model have four properties as below:

– **Bounded leakage of single observation:** A cryptographic scheme typically includes several computation rounds (i.e. observations). In each computation round, an adversary
can extract fractional content of private/secret keys. Namely, adversaries can select a leakage function \( f \) for each computation round and then obtain the leakage information \( f(SK) \), where \( SK \) denotes the involved private/secret keys and the output information of \( f(SK) \) is bounded to \( \lambda \) bits.

- **Only computation leakage**: Adversaries are only allowed to extract fractional content of private/secret keys involved in the current computation round.
- **Independent leakage**: Any two leaked fractional contents of private/secret keys in various computation rounds are mutually independent. For achieving this property, a private/secret key must be updated before (or after) running each computation round.
- **Overall unbounded leakage**: The total amount of leakage information is overall unbounded. Indeed, by the independent leakage property, the total leakage amount of private/secret keys is not strict.

Under the continual leakage model, there are several leakage-resilient encryption and signature schemes based on the conventional public-key settings. In the generic bilinear group (GBG) model (Boneh et al., 2005), Kiltz and Pietrzak (2010) presented a leakage-resilient encryption scheme that allows adversaries to continually extract fractional content of secret/private keys. In Kiltz and Pietrzak’s scheme, each user’s secret key is divided into two components. After/before performing the decryption procedure, a receiver (user) must refresh two components of her/his secret key. The key idea of refreshing employs the multiplicative blinding technique which appeared in Kiltz and Pietrzak (2010). Based on this key idea, Galindo et al. (2016) presented an efficient leakage-resilient ElGamal public-key encryption scheme. Also, Galindo and Virek (2013) proposed the first leakage-resilient signature scheme under the continual leakage model. To improve the performance of their scheme, Tang et al. (2014) presented a modified leakage-resilient signature scheme by employing Boneh et al.’s short signature (Boneh et al., 2001).

Based on an ID-PKS setting, Brakerski et al. (2010) presented the first leakage-resilient ID-based encryption (LR-IBE) scheme under the continual leakage model. Subsequently, Yuen et al. (2012) presented an improvement on Brakerski et al.’s scheme in terms of computational costs. Under the continual leakage model, Wu et al. (2016) proposed the first leakage-resilient ID-based signature (LR-IBS) scheme.

### 1.2. Contribution and Organization

Up to date, no work has been published on leakage-resilient revocable ID-based signature (LR-RIBS) scheme. In the article, we present a new adversary model of LR-RIBS schemes with a cloud revocation authority (CRA) under the continual leakage model. In the adversary model, there are two types of adversaries, namely, Type I adversary (a curious CRA or an outsider) and Type II adversary (a revoked user). As compared with the adversary models of RIBS schemes presented in Tsai et al. (2013b), Hung et al. (2017), Jia et al. (2017), three new key leakage queries, namely, the key extract leak query, time key update leak query and signing leak query are added to our new adversary model. These added leak queries allow an adversary to continuously extract fractional content of private/secret keys participated in the computation rounds.
The first LR-RIBS scheme with CRA is proposed while the revocation functionality is outsourced to the CRA. By employing Kiltz and Pietrzak’s key refreshing idea (Kiltz and Pietrzak, 2010), the proposed LR-RIBS scheme with CRA allows adversaries to continuously gain fractional content of private/secret keys so that its overall amount of leaked content is unbounded and it possesses overall unbounded leakage property. Under the new adversary model and generic bilinear group (GBG) model (Boneh et al., 2005), security analysis is given to show that our LR-RIBS scheme is existential unforgeability against adaptive chosen-message (UF-LR-RIBS-ACMA) attacks of both Types I and II adversaries. Finally, performance analysis and comparisons are made to demonstrate that the proposed LR-RIBS scheme requires some additional computation costs than the previously proposed RIBS schemes. The point is that the proposed LR-RIBS scheme with CRA can resist side-channel attacks. By the simulation experiences (Lynn, 2015) on a smartphone, the proposed LR-RIBS scheme with CRA is still suitable for mobile devices.

The rest of the paper is organized as follows. In Section 2, preliminaries are given. In Section 3, we define the framework and adversary model of LR-RIBS schemes with CRA. In Section 4, we propose a secure LR-RIBS scheme with CRA under the continual leakage model. Section 5 demonstrates the security analysis of the proposed LR-RIBS scheme. In Section 6, we present the performance analysis and comparisons with the previously proposed RIBS schemes. Finally, conclusions are given in Section 7.

2. Preliminaries

Several preliminaries are introduced in this section.

2.1. Bilinear groups

Let $G$ and $G_T$ be two multiplicative cyclic groups with (large) prime order $p$. Let $g$ be a generator of $G$. An admissible bilinear map $\hat{e} : G \times G \rightarrow G_T$ possesses the following three properties:

1. Non-degeneracy: $\hat{e}(g, g) \neq 1$.
2. Bilinearity: for all $r, s \in \mathbb{Z}_p^*$, $\hat{e}(g^r, g^s) = \hat{e}(g, g)^{rs}$.
3. Computability: $\hat{e}(g, g)^{rs}$ can be computed efficiently for any $r, s \in \mathbb{Z}_p^*$.

For the detailed properties and settings with regard to bilinear groups, please refer to Boneh and Franklin (2001), Tsai et al. (2013b), Lynn (2015), Scott (2011).

2.2. Generic Bilinear Group Model

By extending the generic group model presented by Shoup (1997), Boneh et al. (2005) introduced the generic bilinear group (GBG) model. Their GBG model is an adversary model played by adversaries and a challenger. In the GBG model, to perform various kinds of group operations, adversaries have to request the associated group oracles/queries to the challenger. Also, the challenger uses bit strings to denote group elements of $G$ and $G_T$. 
More precisely, the challenger employs two random injective functions $\varepsilon : Z_p \rightarrow \xi$ and $\varepsilon_T : Z_p \rightarrow \xi_T$, respectively, to transform the elements of $G$ and $G_T$ into bit strings in $\xi$ and $\xi_T$. In addition, both $\xi$ and $\xi_T$ have $p$ elements and are disjoint, namely $|\xi| = |\xi_T| = p$ and $\xi \cap \xi_T = \phi$. The discrete logarithm problem on $G$ or $G_T$ will be solved if the adversary discovers a collision encoding element of $G$ or $G_T$.

- **Discrete logarithm problem:** Let $G$ and $G_T$ be two multiplicative cyclic groups of a large prime order $p$. Let $g$ and $\hat{e}(g, g)$ denote the generators of $G$ and $G_T$, respectively. Given a group element $g^z \in G$ or $\hat{e}(g, g)^z \in G_T$ with unknown $z \in Z_p^*$, the discrete logarithm problem in $G$ and $G_T$ is that no probabilistic polynomial time (PPT) algorithm $A$ may obtain $z$ with a non-negligible probability (Boneh et al., 2005).

In the GBG model, there are three group operations, namely, the multiplication $Q_G$ on $G$, the multiplication $Q_T$ on $G_T$, and the bilinear map $Q_P : G \times G \rightarrow G_T$, which is denoted by $\hat{e}$ above. For any $r, s \in Z_p^*$, we have the following properties:

- $Q_G(\varepsilon(r), \varepsilon(s)) \rightarrow \varepsilon(r + s \text{ mod } p)$.
- $Q_T(\varepsilon_T(r), \varepsilon_T(s)) \rightarrow \varepsilon_T(r + s \text{ mod } p)$.
- $Q_P(\varepsilon(r), \varepsilon(s)) \rightarrow \varepsilon_T(rs \text{ mod } p)$.

### 2.3. Entropy of Leakage Content

In information theory, entropy is usually employed to measure the uncertainty of unknown private/secret values. Assume that $W$ is a discrete random variable (i.e. secret value) and $\Pr[W = w]$ denotes the probability of $W = w$. The min-entropy of $W$ is the estimation of $W = w$ with the largest probability, namely, the worst-case predictability of $W$. Two types of min-entropies are defined as below:

- **Min-entropy of $W$:**

  $$H_\infty(X) = -\log_2 \left( \max_w \Pr[W = w] \right).$$

- **Average conditional min-entropy of $W$ under the condition $Z = z$:**

  $$\tilde{H}_\infty(W|Z) = -\log_2 \left( \mathbb{E}_{w \leftarrow Z} \left[ \max_w \Pr[W = w|Z = z] \right] \right).$$

To measure the entropy of a finite discrete random variable (secret value) with fractional leakage content, Dodis et al. (2008) derived the following consequence.

**Lemma 1** (See Dodis et al., 2008). Assume that a leakage function $f : W \rightarrow \{0, 1\}^\lambda$ takes as input a discrete random variable $W$ and the maximal output bit-length is $\lambda$. Under the event $f(W)$, the average conditional min-entropy of $W$ is $\tilde{H}_\infty(W|f(W)) \geq H_\infty(W) - \lambda$.

By Lemma 1, Galindo and Virek (2013) derived the following consequence to measure the probability distribution of a polynomial with multiple random variables and leakage content.
Lemma 2 (See Galindo and Virek, 2013). Let $F \in \mathbb{Z}_p[W_1, W_2, \ldots, W_n]$ be a non-zero polynomial. Its maximal output bit-length of fraction leakage content and degree are $\lambda$ ($0 \leq \lambda \leq \log p$), and at most $d$, respectively. Let $P_i$ be the associated probability distributions of $W_i = w_i$, for $i = 1, 2, \ldots, n$, that satisfy $H_\infty(P_i) \geq \log p - \lambda$. Thus, we have $\Pr[F(w_1, w_2, \ldots, w_n) = 0] \leq \frac{d}{p^{2\lambda}}$ if $w_i \overset{\text{i.i.d.}}{\leftarrow} \mathbb{Z}_p$ (for $i = 1, 2, \ldots, n$) are mutually independent. Therefore, $\Pr[F(w_1, w_2, \ldots, w_n) = 0]$ is negligible if $\lambda < \log p - \omega(\log \log p)$.

3. System Architecture, Framework and Adversary Model

Here, let us present the system architecture, framework and adversary model of LR-RIBS schemes with CRA. In an LR-RIBS scheme with CRA, there are three roles, namely, PKG, CRA and users. Several notations are defined as below:

- $SPK$: the PKG’s system public key.
- $SSK$: the PKG’s system secret key.
- $TPK$: the CRA’s time public key.
- $TSK$: the CRA’s time secret key.
- $ID$: a user’s identity, where $ID \in \{0, 1\}^*$.
- $T_t$: a time period, for $t = 1, 2, \ldots, z$, where $z$ represents the number of periods.
- $UTKID_{ID, t}$: the time update key of the user with identity $ID$ at period $T_t$.
- $USKID$: the secret key of the user with identity $ID$.
- $msg$: a message.
- $\sigma$: a signature value.

3.1. System Architecture

The system architecture of LR-RIBS schemes with CRA is depicted in Fig. 1. Firstly, the PKG sets the system secret key $SSK$, the time secret key $TSK$ and a total number $z$ of periods $T_0, T_1, \ldots, T_z$ while computing public parameters $PP$ and sending $TSK$ to the CRA. The PKG employs $SSK$ to generate the secret key $USKID$ of the user with identity $ID$. By a secure channel, the PKG sends $USKID$ to the user. For non-revoked user $ID$ at time period $T_t$, the CRA employs $TSK$ to generate the time update key $UTKID_{ID, t}$. By a public channel (e.g. e-mail), the CRA sends $UTKID_{ID, t}$ to the user. Hence, a user’s private key consists of two parts, namely, $USKID$ and $UTKID_{ID, t}$. Suppose that the user (signer) with identity $ID$ at period $T_t$ would like to sign a message $msg$, the signer employs $USKID$ and $UTKID_{ID, t}$ to generate a signature value $\sigma$ and sends it to a verifier.

3.2. Framework

To achieve overall unbounded leakage property (Galindo and Virek, 2013; Wu et al., 2016, 2018, 2019), a private/secret key must be split into two components. Additionally, each private/secret key participated in the associated algorithm is also refreshed before/after
each algorithm invocation. In such a case, the PKG’s system secret key $SSK$, the CRA’s time secret key $TSK$ and a user’s secret key $USK_{ID}$ must, respectively, be split into two components. In the meantime, the PKG’s $SSK$ must be refreshed before/after performing the key extract algorithm. Also, the CRA’s $TSK$ and a user’s secret key $USK_{ID}$ must be refreshed before/after performing the time key update and signing algorithms, respectively.

In the following, we define the framework (syntax) of LR-RIBS schemes with CRA.

**Definition 1.** An LR-RIBS scheme with CRA includes five algorithms as follows:

- **System setup:** The PKG first sets the system secret key $SSK = (SSK_{0,1}, SSK_{0,2})$, a time secret key $TSK = (TSK_{0,1}, TSK_{0,2})$ and $z$ periods $T_0, T_1, \ldots, T_z$ while generating public parameters $PP$ and sending $TSK$ to the CRA using a secure channel. The PKG holds $SSK = (SSK_{0,1}, SSK_{0,2})$ and publishes $PP$.
- **Key extract:** In the $i$-th invocation of the $Key$ $extract$ algorithm, the PKG refreshes $(SSK_{i-1,1}, SSK_{i-1,2})$ to set the current system secret key $(SSK_{i,1}, SSK_{i,2})$. The PKG takes as input a user’s identity ID and generates the user’s associated secret key $USK_{ID}$. The PKG returns $USK_{ID}$ to the user via a secure channel. Afterwards, the user sets her/his initial secret key $USK_{ID} = (USK_{ID,0,1}, USK_{ID,0,2})$.
- **Time key update:** In the $j$-th invocation of the $Time$ $key$ $update$ algorithm, the CRA refreshes $(TSK_{j-1,1}, TSK_{j-1,2})$ to set the current time secret key $(TSK_{j,1}, TSK_{j,2})$. The CRA takes as input a user’s identity ID and a period $T_i$, and generates the user’s time update key $UTK_{ID,i}$. The CRA sends $UTK_{ID,i}$ to the user via a public channel.
• **Signing:** In the $k$-th invocation of the *Signing* algorithm, the user $ID$ refreshes $(USK_{ID,k-1,1}, USK_{ID,k-1,2})$ to set her/his current secret key $(USK_{ID,k,1}, USK_{ID,k,2})$. At period $T_t$, the user $ID$ employs her/his current secret key $(USK_{ID,k,1}, USK_{ID,k,2})$ and time update key $UTK_{ID,t}$ to generate a signature value $\sigma$ on a message $msg$. The user outputs a signature tuple $(ID, T_t, msg, \sigma)$.

• **Verifying:** Upon receiving $(ID, T_t, msg, \sigma)$, the verifier returns either “accept” or “reject”.

3.3. **Adversary Model (Security Notions)**

By extending the adversary model (security notions) presented in the RIBS schemes (Tsai et al., 2013b; Hung et al., 2017; Jia et al., 2017), we present an adversary model of LR-RIBS schemes with CRA, which allows an adversary to extract fractional content of the private/secret keys. According to our framework, an adversary can extract fractional content of the PKG’s system secret key $(SSK_{i,1}, SSK_{i,2})$ in the $i$-th invocation of the *Key extract* algorithm. Also, an adversary can extract fractional content of the CRA’s time secret key $(TSK_{j,1}, TSK_{j,2})$ in the $j$-th invocation of the *Time key update* algorithm. In the $k$-th invocation of the *Signing* algorithm by the user $ID$ at period $T_t$, an adversary can extract fractional content of the user’s secret key $(USK_{ID,k,1}, USK_{ID,k,2})$. Six leakage functions $f_{KE,i}, h_{KE,i}, f_{TKU,j}, h_{TKU,j}, f_{S,k}, h_{S,k}$ are employed to model the leakage abilities. More precisely, the output is in the form of three pairs $(f_{KE,i}(SSK_{i,1}), h_{KE,i}(SSK_{i,2})), (f_{TKU,j}(TSK_{j,1}), h_{TKU,j}(TSK_{j,2}))$ and $(f_{S,k}(USK_{ID,k,1}), h_{S,k}(USK_{ID,k,2}))$ to indicate, respectively, the fractional content of $(SSK_{i,1}, SSK_{i,2}), (TSK_{j,1}, TSK_{j,2})$ and $(USK_{ID,k,1}, USK_{ID,k,2})$. Also, we require that the output bit-string lengths of the six leakage functions are at most $\lambda$. For brevity, we introduce the following notation which will be used in the sequel:

- $Af_{KE,i} = f_{KE,i}(SSK_{i,1})$.
- $Ah_{KE,i} = h_{KE,i}(SSK_{i,2})$.
- $Af_{TKU,j} = f_{TKU,j}(TSK_{j,1})$.
- $Ah_{TKU,j} = h_{TKU,j}(TSK_{j,2})$.
- $Af_{S,k} = f_{S,k}(USK_{ID,k,1})$.
- $Ah_{S,k} = h_{S,k}(USK_{ID,k,2})$.

In the adversary model of LR-RIBS schemes with CRA, there are two types of adversaries:

• **Type I adversary** $A_I$ (a curious CRA or an outsider): $A_I$ denotes a curious CRA or an outsider. $A_I$ is allowed to acquire the time update key $UTK_{ID,t}$ for any user $ID$ and period $T_t$. Meanwhile, $A_I$ can acquire the secret key $USK_{ID}$ for any $ID$, except for the target identity $ID^*$. In addition, $A_I$ can extract fractional content of the target user’s secret key $USK_{ID^*}$ in the *Signing* algorithm and the PKG’s system secret key $SSK$ in the *Key extract* algorithm.

• **Type II adversary** $A_{II}$ (a revoked user): $A_{II}$ denotes the adversary who was a legal user with identity $ID^*$ and has been revoked at period $T_t^*$. In such a case, $A_{II}$ is allowed
to acquire the secret key $USK_{ID}$ and time update key $UTK_{ID,t}$ for any $ID$ and $T_t$. But, $A_{II}$ is disallowed to acquire the time update key $UTK_{ID,t^*}$ for the target identity $ID^*$ at period $T_t^*$. In addition, $A_{II}$ can extract fractional content of the CRA’s time secret key $TSK$ in the Time key update algorithm.

The following security game $G_{LR-RIBS}$ is used to model the adversary model (security notions) of LR-RIBS schemes with CRA.

**Definition 2 ($G_{LR-RIBS}$).** For the LR-RIBS scheme with CRA, the game $G_{LR-RIBS}$ is used to model the interactions between an adversary $A$ ($A_I$ or $A_{II}$) and a challenger $C$. It is said that the LR-RIBS scheme with CRA is existential unforgeability against adaptive chosen-message attacks (UF-LR-RIBS-ACMA) if no probabilistic polynomial-time (PPT) adversary may win $G_{LR-RIBS}$ with a non-negligible probability. Three phases of $G_{LR-RIBS}$ are presented as below:

- **Setup phase.** The challenger $C$ performs the System setup algorithm in Definition 1 to set a system secret key $SSK = (SSK_{0,1}, SSK_{0,2})$, a time secret key $TSK = (TSK_{0,1}, TSK_{0,2})$ and a total number $z$ of periods $T_0, T_1, \ldots, T_z$. Meanwhile, $C$ sets and publishes public parameters $PP$. In addition, by a secure channel, $C$ sends the time secret key $TSK$ to the CRA. Also, if $A$ is a Type I adversary, the time secret key $TSK$ is sent to $A$.

- **Query phase.** The adversary $A$ can request the following queries to $C$ adaptively.
  - **Key extract query ($ID$):** Upon receiving a user’s $ID$, $C$ generates and sends the user’s corresponding secret key to $A$.
  - **Key extract leak query ($i$, $f_{KE,i}$, $h_{KE,i}$):** Upon receiving two leakage functions $f_{KE,i}$ and $h_{KE,i}$, $C$ computes the fractional leakage content $\Lambda f_{KE,i}$ and $\Lambda h_{KE,i}$ of the PKG’s system secret key $(SSK_{i,1}, SSK_{i,2})$. Afterwards, $C$ sends $\Lambda f_{KE,i}$ and $\Lambda h_{KE,i}$ to $A$. For the $i$-th Key extract query, $A$ is allowed to request the Key extract leak query only once.
  - **Time key update query ($ID$, $T_t$):** Upon receiving a user’s $ID$ and a period $T_t$, $C$ generates and sends the user’s time update key $UTK_{ID,t}$ to $A$.
  - **Time key update leak query ($j$, $f_{TKU,j}$, $h_{TKU,j}$):** Upon receiving two leakage functions $f_{TKU,j}$ and $h_{TKU,j}$, $C$ computes the fractional leakage content $\Lambda f_{TKU,j}$ and $\Lambda h_{TKU,j}$ of the CRA’s time secret key $(TSK_{j,1}, TSK_{j,2})$. Afterwards, $C$ sends $\Lambda f_{TKU,j}$ and $\Lambda h_{TKU,j}$ to $A$. For the $j$-th Time key update query, $A$ is allowed to request the Time key update leak query only once.
  - **Signing query ($ID$, $T_t$, $msg$):** Upon receiving a message $msg$ and a user’s $ID$ at period $T_t$, $C$ generates a signature value $\sigma$ and returns $(ID, T_t, msg, \sigma)$ to $A$.
  - **Signing leak query ($ID$, $k$, $f_{S,k}$, $h_{S,k}$):** Upon receiving two leakage functions $f_{S,k}$ and $h_{S,k}$, $C$ computes the fraction leakage content $\Lambda f_{S,k}$ and $\Lambda h_{S,k}$ of the signer’s secret key $(USK_{ID,k,1}, USK_{ID,k,2})$, and returns $\Lambda f_{S,k}$ and $\Lambda h_{S,k}$ to $A$. In the $k$-th Signing query requested by the user $ID$, $A$ is allowed to issue the Signing leak query only once.

- **Forgery phase.** The adversary $A$ outputs a signature tuple $(ID^*, T_t^*, msg^*, \sigma^*)$ and $A$ wins $G_{LR-RIBS}$ if the following conditions hold.
(1) If $A$ is a Type I adversary (a curious CRA or an outsider), the Key extract query on $ID^*$ cannot be requested.

(2) If $A$ is a Type II adversary (a revoked user), the Time key update query on $(ID^*, T_i^*)$ cannot be requested.

(3) The Signing query on $(ID^*, T_i^*, msg^*)$ cannot be requested.

(4) The output of the Verifying algorithm on $(ID^*, T_i^*, msg^*, \sigma^*)$ is “accept”.

4. The Proposed LR-RIBS Scheme with CRA

Here, let us present the first LR-RIBS scheme with CRA that consists of five algorithms as below:

- **System setup:** The PKG runs the System setup algorithm to choose two groups $G = \langle g \rangle$ and $G_T = \langle \hat{e}(g, g) \rangle$ of a large prime order $p$. The algorithm sets a total number $z$ of periods $T_0, T_1, \ldots, T_z$. Moreover, the algorithm performs the following steps to compute the system secret key $SSK = (SSK_{0,1}, SSK_{0,2})$, the time secret key $TSK = (TSK_{0,1}, TSK_{0,2})$ and public parameters $PP$.

  (1) Choose a random integer $\alpha \in Z_p^*$, and compute the system secret key $SSK = g^\alpha$ and the system public key $SPK = \hat{e}(g, g^\alpha)$. Also, choose an integer $\alpha \in Z_p^*$ at random, and compute the initial system secret key $(SSK_{0,1}, SSK_{0,2}) = (g^\alpha, SSK : g^{-\alpha})$.

  (2) Choose a random integer $\beta \in Z_p^*$, and set the time secret key $TSK = g^\beta$ and time public key $TPK = \hat{e}(g, g^\beta)$.

  (3) Choose six random integers $a, b, c, \gamma, x, y, z \in Z_p^*$, and set $U = g^a$, $V = g^b$, $W = g^c$, $X = g^x$, $Y = g^y$ and $Z = g^z$.

  (4) Set $PP = (p, G, G_T, \hat{e}, g, SPK, TPK, U, V, W, X, Y, Z)$.

  (5) Finally, the PKG holds $(SSK_{0,1}, SSK_{0,2})$ and publishes $PP$ while sending $TSK$ to the CRA via a secure channel. Afterwards, the CRA selects a random integer $b \in Z_p^*$, and uses $TSK$ to set the initial time secret key $(TSK_{0,1}, TSK_{0,2}) = (g^b, TSK \cdot g^{-b})$.

- **Key extract:** In the $i$-th invocation of the Key extract algorithm, the PKG sets the current system secret key $(SSK_{i,1}, SSK_{i,2})$ by refreshing $(SSK_{i-1,1}, SSK_{i-1,2})$. Afterwards, the PKG takes as input a user’s identity $ID$ and carries out the following steps:

  (1) Choose a random integer $a \in Z_p^*$, and update the PKG’s system secret key $(SSK_{i,1}, SSK_{i,2}) = (SSK_{i-1,1} \cdot g^a, SSK_{i-1,2} \cdot g^{-a})$.

  (2) Choose a random integer $\gamma \in Z_p^*$, and compute $QK_{ID} = g^\gamma$.

  (3) Compute $TID = STID \text{ (mod } p\text{)}$, where $STID$ is the integer value of the bit string $IDQK_{ID}$. And compute the temporary information $TI_{KE} = SSK_{i,1} \cdot (U \cdot V^{TID})^\gamma$ and the user’s secret key $USK_{ID} = SSK_{i,2} \cdot TI_{KE}$.

  (4) Finally, by a secure channel, the PKG sends $USK_{ID}$ and $QK_{ID}$ to the user. Upon receiving $USK_{ID}$ and $QK_{ID,1}$, the user randomly selects an integer $c \in Z_p^*$, and sets the user’s initial secret key $(USK_{ID,0,1}, USK_{ID,0,2}) = (g^c, USK_{ID} \cdot g^{-c})$.

- **Time key update:** In the $j$-th invocation of the Time key update algorithm, the CRA sets the current time secret key $(TSK_{j,1}, TSK_{j,2})$ by refreshing $(TSK_{j-1,1}, TSK_{j-1,2})$. Afterwards, the CRA takes as input a user’s identity $ID$ and a period $T_i$, and carries out the following steps:
(1) Choose a random integer \( b \in \mathbb{Z}_p^* \), and update the CRA’s current time secret key 
\((TSK_{i,1}, TSK_{i,2}) = (TSK_{i-1,1} \cdot g^b, TSK_{i-1,2} \cdot g^{-b})\).
(2) Randomly select an integer \( \eta \in \mathbb{Z}_p^* \), and compute \( QTK_{ID,t} = g^\eta \).
(3) Compute \( TTD = STTD \pmod{p} \), where \( STTD \) is the integer value of the bit string 
\(|ID||T_t||QTK_{ID,t}|. \) And compute the temporary information \( TI_{TKU} = TSK_{j,1} \cdot (W \cdot \langle X^{TTD} \rangle)^\eta \) and the user’s time secret key \( UTK_{ID,t} = TSK_{j,2} \cdot TI_{TKU} \).
(4) Finally, by a secure channel, the CRA sends \( UTK_{ID,t} \) and \( QTK_{ID,t} \) to the user.

**Signing:** For the \( k \)-th invocation of the Signing algorithm, the user \( ID \) sets her/his current secret key 
\((USK_{ID,k,1}, USK_{ID,k,2}) \) by refreshing \((USK_{ID,k-1,1}, USK_{ID,k-1,2}) \). At period \( T_t \), the user employs her/his current secret key \((USK_{ID,k,1}, USK_{ID,k,2}) \) and time update key \( UTK_{ID,t} \) to output a signature tuple \((ID, T_t, msg, \sigma) \) with \( \sigma = (\sigma_1 = QK_{ID}, \sigma_2 = QTK_{ID,t}, \sigma_3, \sigma_4) \) by carrying out the following steps:
(1) Choose a random integer \( c \in \mathbb{Z}_p^* \), and update the signer’s secret key 
\((USK_{ID,k,1}, USK_{ID,k,2}) = (USK_{ID,k-1,1} \cdot g^c, USK_{ID,k-1,2} \cdot g^{-c}) \).
(2) Choose a random integer \( \delta \in \mathbb{Z}_p^* \), and compute \( \sigma_3 = g^\delta \), the temporary information 
\( TI_S = USK_{ID,k,1} \cdot TUK_{ID,t} \cdot (Y \cdot Z^{msg})^\delta \) and \( \sigma_4 = USK_{ID,k,2} \cdot TI_S \).
(3) Set the signature tuple \((ID, T_t, msg, \sigma) \) with \( \sigma = (\sigma_1 = QK_{ID}, \sigma_2 = QTK_{ID,t}, \sigma_3, \sigma_4) \).

**Verifying:** For a signature tuple \((ID, T_t, msg, \sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \), the verifier compute 
\( TID = STID \pmod{p} \) and \( TTD = STTD \pmod{p} \), where \( STID \) and \( STTD \) are the integer values of the bit strings \(|ID||\sigma_1 \) and \(|ID||T_t||\sigma_2 \), respectively. The verifier outputs “accept” if the verifying equality \( \hat{e}(g, \sigma_4) = SPK \cdot TPK \cdot \hat{e}(\sigma_1, U \cdot V^{TID}) \cdot \hat{e}(\sigma_2, W \cdot X^{TTD}) \cdot \hat{e}(\sigma_3, Y \cdot Z^{msg}) \) holds; otherwise outputs “reject”.

Here, let us discuss the correctness of the verifying equality. By the key refreshing procedures of those secret values employed in the proposed scheme, we have:
- \( SSK = SSK_{0,1} \cdot SSK_{0,2} = \cdots = SSK_{i-1,1} \cdot SSK_{i-1,2} = SSK_{i,1} \cdot SSK_{i,2} \).
- \( TSK = TSK_{0,1} \cdot TSK_{0,2} = \cdots = TSK_{i-1,1} \cdot TSK_{i-1,2} = TSK_{i,1} \cdot TSK_{i,2} \).
- \( USK_{ID} = USK_{ID,0,1} \cdot USK_{ID,0,2} = \cdots = USK_{ID,k-1,1} \cdot USK_{ID,k-1,2} = USK_{ID,k,1} \cdot USK_{ID,k,2} \).

Hence, the equality is verified by
\[
\hat{e}(g, \sigma_4) = \hat{e}(g, USK_{ID,k,2} \cdot USK_{ID,k,1} \cdot UTK_{ID,t} \cdot (Y \cdot Z^{msg})^\delta) \\
= \hat{e}(g, USK_{ID} \cdot UTK_{ID,t} \cdot (Y \cdot Z^{msg})^\delta) \\
= \hat{e}(g, USK_{ID} \cdot (TSK_{j,2} \cdot TSK_{j,1} \cdot (W \cdot X^{TTD})^\eta) \cdot (Y \cdot Z^{msg})^\delta) \\
= \hat{e}(g, USK_{ID} \cdot TSK \cdot (W \cdot X^{TTD})^\eta \cdot (Y \cdot Z^{msg})^\delta) \\
= \hat{e}(g, SSK \cdot (U \cdot V^{TID})^\gamma \cdot TSK \cdot (W \cdot X^{TTD})^\eta \cdot (Y \cdot Z^{msg})^\delta) \\
= \hat{e}(g, SSK \cdot TSK \cdot (U \cdot V^{TID})^\gamma \cdot (W \cdot X^{TTD})^\eta \cdot (Y \cdot Z^{msg})^\delta) \\
= \hat{e}(g, SSK) \cdot \hat{e}(g, TSK) \cdot \hat{e}(g, (U \cdot V^{TID})^\gamma) \cdot \hat{e}(g, (W \cdot X^{TTD})^\eta) \\
= \hat{e}(g, SSK) \cdot \hat{e}(g, TSK) \cdot \hat{e}(g, (U \cdot V^{TID})^\gamma) \cdot \hat{e}(g, (W \cdot X^{TTD})^\eta)
\]
\[
\hat{e}(g, (Y \cdot Z^{msg})^\delta) \\
= \hat{e}(g, SSK) \cdot \hat{e}(g, TSK) \cdot \hat{e}(g, U \cdot V^{TID}) \cdot \hat{e}(g^n, W \cdot X^{TTD}) \cdot \hat{e}(g^\gamma, Y \cdot Z^{msg}) \\
= \hat{e}(g, SSK) \cdot \hat{e}(g, TSK) \cdot \hat{e}(QK_{ID}, U \cdot V^{TID}) \cdot \hat{e}(QTK_{ID, I}, W \cdot X^{TTD}) \\
\cdot \hat{e}(\sigma_3, Y \cdot Z^{msg}) \\
= SPK \cdot TPK \cdot \hat{e}(\sigma_1, U \cdot V^{TID}) \cdot \hat{e}(\sigma_2, W \cdot X^{TTD}) \cdot \hat{e}(\sigma_3, Y \cdot Z^{msg}).
\]

5. Security Analysis

Let us analyse the security of our LR-RIBS scheme with CRA. By the adversary model (i.e. security game \(G_{LR-RIBS}\)) of LR-RIBS schemes with CRA, there are two types of adversaries. In the GBG model, Theorem 1 demonstrates that our scheme is provably secure against Type I adversary. In Theorem 2, we prove that our scheme is also provably secure against Type II adversary.

**Theorem 1.** In the GBG model, our LR-RIBS scheme with CRA possesses existential unforgeability under the UF-LR-RIBS-ACMA attack of Type I adversary (a curious CRA or an outsider).

**Proof.** Let \(A_I\) denote a Type I adversary in the security game \(G_{LR-RIBS}\) played with a challenger \(C\). \(A_I\) is allowed to request all queries in the security game \(G_{LR-RIBS}\) while the number of queries issued by \(A_I\) is at most \(q\) times. In the GBG model introduced in earlier section, there are three group queries (oracles) \(Q_G, Q_T\) and \(Q_p\). In such a case, the challenger \(C\) also responds the queries \(Q_G, Q_T\) and \(Q_p\) issued by the adversary \(A_I\), where these queries are provided in the Query phase of \(G_{LR-RIBS}\). For \(G_{LR-RIBS}\) on the proposed LR-RIBS scheme with CRA, three phases (Setup, Query and Forgery) are presented as below:

- **Setup phase:** The challenger \(C\) carries out the System Setup algorithm of our scheme to generate \(SSK, TSK\), a total number \(z\) of periods \(T_0, T_1, \ldots, T_z\) and \(PP = (p, G, G_T, \hat{e}, g, SPK, TPK, U, V, W, X, Y, Z)\). Additionally, \(C\) constructs four lists \(L_G, L_T, L_K\) and \(L_{TK}\) to record the related parameters and results of the queries issued by the adversary.

  - \(L_G\) and \(L_T\) are, respectively, employed to record all group elements of \(G\) and \(G_T\).
    1. \(L_G\) contains pairs of the form \((\mathcal{E}G_{m,n,r}, \xi G_{m,n,r})\), where \(\mathcal{E}G_{m,n,r}\) represents an element (multivariate polynomial) in \(G\) and \(\xi G_{m,n,r}\) is the associated bit string. Here, \(m\) and \(n\), respectively, denote the query type and the \(n\)-th query, and the index \(r\) represents the \(r\)-th element of \(G\). Initially, nine pairs \((\mathcal{E}g, \xi G_{1,1,1}), (\mathcal{E}U, \xi G_{1,1,2}), (\mathcal{E}V, \xi G_{1,1,3}), (\mathcal{E}W, \xi G_{1,1,4}), (\mathcal{E}X, \xi G_{1,1,5}), (\mathcal{E}Y, \xi G_{1,1,6}), (\mathcal{E}Z, \xi G_{1,1,7}), (\mathcal{E}SSK, \xi G_{1,1,8})\) and \((\mathcal{E}TSK, \xi G_{1,1,9})\) are recorded in \(L_G\).
    2. \(L_T\) contains pairs of the form \((\mathcal{E}T_{m,n,r}, \xi T_{m,n,r})\), where \((\mathcal{E}T_{m,n,r}\) represents an element (multivariate polynomial) in \(G/G_T\) and \(\xi T_{m,n,r}\) is the associated bit
string. The meanings of the indices $m$, $n$ and $r$ are the same with those in $L_G$. Initially, two pairs $(\Xi \text{SPK}, \xi T_{I,1}, 1)$ and $(\Xi \text{TPK}, \xi T_{I,1}, 2)$ are recorded in $L_T$, where $\Xi \text{SPK} = \Xi g \cdot \Xi \text{SSK}$ and $\Xi \text{TPK} = \Xi g \cdot \Xi \text{TSK}$.

It is worth mentioning that $C$ employs two rules to respond the transformation request as below:

1. When $C$ receives $\Xi G_{m,n,r}/\Xi T_{m,n,r}$, $C$ looks for $(\Xi G_{m,n,r}, \xi G_{m,n,r})/(\Xi T_{m,n,r}, \xi T_{m,n,r})$ in $L_G/L_T$. If so, $C$ returns the associated bit string $\xi G_{m,n,r}/\xi T_{m,n,r}$. Otherwise, $C$ randomly selects a distinct bit string $\xi G_{m,n,r}/\xi T_{m,n,r}$ and returns it. Finally, $C$ adds $(\Xi G_{m,n,r}, \xi G_{m,n,r})/(\Xi T_{m,n,r}, \xi T_{m,n,r})$ in $L_G/L_T$.

2. When $C$ receives $\xi G_{m,n,r}/\xi T_{m,n,r}$ in $L_G/L_T$, $C$ returns the associated multivariate polynomial $\Xi G_{m,n,r}/\Xi T_{m,n,r}$ if it is found. Otherwise, $C$ terminates the game.

- $L_K$ contains tuples of the form $(ID, \Xi \text{USK}_ID, \Xi \text{QK}_ID)$, where the multivariate polynomials $\Xi \text{USK}_ID$ and $\Xi \text{QK}_ID$, respectively, denote the user’s $\text{USK}_ID$ and $\text{QK}_ID$ in the Key extract phase.

- $L_{TK}$ contains tuples of the form $(ID, T, \Xi \text{UTK}_ID,t, \Xi \text{QTK}_ID,t)$, where the multivariate polynomials $\Xi \text{UTK}_ID,t$ and $\Xi \text{QTK}_ID,t$, respectively, denote the user’s $\text{UTK}_ID,t$ and $\text{QTK}_ID,t$ in the Time key update phase.

Finally, $C$ sends these public parameters $\Xi g$, $\Xi U$, $\Xi V$, $\Xi W$, $\Xi X$, $\Xi Y$, $\Xi Z$, $\Xi \text{SPK}$ and $\Xi \text{TPK}$ to $A_I$.

- **Query phase**: $A_I$ can request the following queries to $C$ adaptively:

  - **Group query** $Q_G(\xi G_{Q,i,1}, \xi G_{Q,i,2}, OP)$: Upon receiving the $i$-th $Q_G$ with a pair of bit strings $(\xi G_{Q,i,1}, \xi G_{Q,i,2})$ and an $OP$ operation, $C$ carries out the following steps:
    1. Transform $\xi G_{Q,i,1}$ and $\xi G_{Q,i,2}$, respectively, to gain the corresponding polynomials $\Xi G_{Q,i,1}$ and $\Xi G_{Q,i,2}$ in $L_G$.
    2. Compute the resulting polynomial $\Xi G_{Q,i,3} = \Xi G_{Q,i,1} + \Xi G_{Q,i,2}$ if $OP$ = “multiplication”, and $\Xi G_{Q,i,3} = \Xi G_{Q,i,1} - \Xi G_{Q,i,2}$ if $OP$ = “division”.
    3. Transform and return the bit string $\xi G_{Q,i,3}$ of $\Xi G_{Q,i,3}$.

  - **Pairing query** $Q_P(\xi G_{P,i,1}, \xi G_{P,i,2})$: Upon receiving the $i$-th $Q_P$ with a pair of bit strings $(\xi G_{P,i,1}, \xi G_{P,i,2})$, $C$ carries out the following steps:
    1. Transform $\xi G_{P,i,1}$ and $\xi G_{P,i,2}$, respectively, to gain the corresponding polynomials $\Xi G_{P,i,1}$ and $\Xi G_{P,i,2}$.
    2. Compute the resulting polynomial $\Xi T_{P,i,1} = \Xi G_{P,i,1} \cdot \Xi G_{P,i,2}$.
    3. Transform and return the bit string $\xi T_{P,i,1}$ of $\Xi T_{P,i,1}$.

  - **Key extract query (ID)**: Upon receiving the $i$-th Key extract query with a user’s $ID$, $C$ looks for $(ID, \Xi \text{USK}_ID, \Xi \text{QK}_ID)$ in $L_K$. If so, $C$ returns two corresponding bit strings $\xi \text{USK}_ID$ of $\Xi \text{USK}_ID$ and $\xi \text{QK}_ID$ of $\Xi \text{QK}_ID$ to $A_I$. Otherwise, $C$ carries out the following steps:
    1. Choose a new variate $\Xi T G_{KE,i,1}$ in $G$. 
(2) Set the polynomial $\mathcal{Z}QK_{ID} = \mathcal{Z}TG_{KE,i,1}$ and $\mathcal{Z}TID = ID||\mathcal{Z}QK_{ID}$.
(3) Compute the user’s secret key $\mathcal{Z}USK_{ID} = \mathcal{Z}SSK + \mathcal{Z}TG_{KE,i,1} \cdot (\mathcal{Z}U + \mathcal{Z}V \cdot \mathcal{Z}TID)$ while adding $(ID, \mathcal{Z}USK_{ID}, \mathcal{Z}QK_{ID})$ in $L_K$.
(4) Transform and return two corresponding bit strings $\xi USK_{ID}$ of $\mathcal{Z}USK_{ID}$ and $\xi QK_{ID}$ of $\mathcal{Z}QK_{ID}$ to $A_I$.

- **Key extract leak query** ($i, f_{KE,i}, h_{KE,i}$): Upon receiving the $i$-th Key extract leak query with two leakage functions $f_{KE,i}$ and $h_{KE,i}$, $C$ returns the fraction leakage content $\Lambda f_{KE,i}$ and $\Lambda h_{KE,i}$ to $A_I$, where $\Lambda f_{KE,i} = f_{KE,i}(SSK_{i,1}, a, \gamma)$ and $\Lambda h_{KE,i} = h_{KE,i}(SSK_{i,2}, a, \gamma, TI_{KE})$. Note that in the $i$-th Key extract query, $A_I$ is allowed to issue the Key extract leak query only once.

- **Time key update query** ($ID, T_t$): In the $j$-th Time key update query with $ID$ and $T_t$, $C$ looks for $(ID, T_t, \mathcal{Z}UTK_{ID,t}, \mathcal{Z}QTK_{ID,t})$ in $L_{TK}$. If so, $C$ returns two corresponding bit strings $\xi UTK_{ID,t}$ and $\xi QTK_{ID,t}$ to $A_I$. Otherwise, $C$ carries out the following steps:
  (1) Choose a new variate $\mathcal{Z}TG_{TKU,ID,t,1}$ in $G$.
  (2) Set the polynomial $\mathcal{Z}QTK_{ID,t} = \mathcal{Z}TG_{TKU,ID,t,1}$ and $\mathcal{TID} = ID||\mathcal{Z}QTK_{ID,t}$.
  (3) Set the user’s time update key query $\mathcal{Z}UTK_{ID,t} = \mathcal{Z}TSK + \mathcal{Z}TG_{TKU,ID,t,1} \cdot (\mathcal{Z}W + \mathcal{Z}X \cdot \mathcal{TID}ID_{t,1})$ while adding $(ID, T_t, \mathcal{Z}UTK_{ID,t}, \mathcal{Z}QTK_{ID,t})$ in $L_{TK}$.
  (4) Transform and return two corresponding bit strings $\xi UTK_{ID,t}$ of $\mathcal{Z}UTK_{ID,t}$ and $\xi QTK_{ID,t}$ of $\mathcal{Z}QTK_{ID,t}$ to $A_I$.

Note that $A_I$ is a curious CRA or an outsider who can gain the user’s time update key by the Time key update query. Hence $A_I$ has no need to request the Time key update leak query.

- **Signing query** ($ID, T_t, \text{msg}$): Upon receiving the $k$-th Signing query of the user $ID$, by taking the period $T_t$ and the message $\text{msg}$ as input, $C$ carries out the following steps:
  (1) By $ID$, look for $(ID, \mathcal{Z}USK_{ID}, \mathcal{Z}QK_{ID})$ in $L_K$.
  (2) By $ID$ and $T_t$, look for the time update key query $(ID, T_t, \mathcal{Z}UTK_{ID,t}, \mathcal{Z}QTK_{ID,t})$ in $L_{TK}$.
  (3) Set $\mathcal{Z}\sigma_1 = \mathcal{Z}QK_{ID}$ and $\mathcal{Z}\sigma_2 = \mathcal{Z}QTK_{ID,t}$.
  (4) Choose a new variate $\mathcal{Z}TG_{S,k,1}$ in $G$ and set $\mathcal{Z}\sigma_3 = \mathcal{Z}TG_{S,k,1}$.
  (5) Set $\mathcal{Z}\sigma_4 = \mathcal{Z}USK_{ID} + \mathcal{Z}UTK_{ID,t} + \mathcal{Z}TG_{S,i,1} \cdot (\mathcal{Z}Y + \mathcal{Z}msg \cdot \mathcal{Z}Z)$. 
  (6) Transform $(\mathcal{Z}\sigma_1, \mathcal{Z}\sigma_2, \mathcal{Z}\sigma_3, \mathcal{Z}\sigma_4)$ to gain the corresponding bit strings $(\xi_\sigma_1, \xi_\sigma_2, \xi_\sigma_3, \xi_\sigma_4)$ and return them to $A_I$.

- **Signing leak query** ($ID, k, f_{S,k}, h_{S,k}$): Upon receiving the $k$-th Signing query of the user $ID$, by taking as input two leakage functions $f_{S,k}$ and $h_{S,k}$, $C$ returns the fraction leakage content $\Lambda f_{S,k}$ and $\Lambda h_{S,k}$ to $A_I$, where $\Lambda f_{S,k} = f_{S,k}(USK_{ID,k,1}, UTK_{ID,t,1}, c, \delta)$ and $\Lambda h_{S,k} = h_{S,k}(USK_{ID,k,2}, c, TI_S)$. Note that for the $k$-th Signing query, $A_I$ is allowed to issue the Signing leak query only once.

  - **Forgery phase**: $A_I$ outputs $(ID^*, T_t^*, \text{msg}^*, (\xi_\sigma_1^*, \xi_\sigma_2^*, \xi_\sigma_3^*, \xi_\sigma_4^*))$. $A_I$ is disallowed to request the Signing query $(ID^*, T_t^*, \text{msg}^*)$. Since $A_I$ is a curious CRA or an outsider, $A_I$ may request the Time key update query $(ID^*, T_t^*)$, but does not request the Key extract query $(ID^*)$. $C$ transforms $(\xi_\sigma_1^*, \xi_\sigma_2^*, \xi_\sigma_3^*, \xi_\sigma_4^*)$ to gain the corresponding polynomials $\mathcal{Z}\sigma_1^*, \mathcal{Z}\sigma_2^*, \mathcal{Z}\sigma_3^*, \mathcal{Z}\sigma_4^*$ while setting $TID^* = ID^*||\xi_\sigma_1^*$ and
Let \( TTD^* = ID^*||T^*_1||\xi\sigma_2^* \). If the equality \( \Xi G \cdot \Xi \sigma_4^* = \Xi SPK + \Xi kTPK + \Xi \sigma_1^* \cdot (\Xi U + TID^* \cdot \Xi V) + \Xi \sigma_2^* \cdot (\Xi W + TTD^* \cdot \Xi X) + \Xi \sigma_3^* \cdot (\Xi Y + msg^* \cdot \Xi Z) \) holds, we say that \( A_I \) wins the game \( G_{LR-RIBS} \).

For evaluating the probability that \( A_I \) wins \( G_{LR-RIBS} \), let us first compute the number of group elements and the maximal degrees of polynomials in \( L_G/L_T \).

(1) The number of group elements in \( L_G \) and \( L_T \) is at most \( 6q \) by the following evaluations:
- In the Setup phase, nine group elements are initially added in \( L_G \) and two group elements are initially added in \( L_T \).
- For each \( Q_G \), \( Q_T \) and \( Q_P \) query, three new group elements could be generated and added in \( L_G \) or \( L_T \).
- In the Key extract query for a new user, two new group elements are generated and added in \( L_G \).
- In the Time key update query for a user at a period, two new group elements are generated and added in \( L_G \).
- In each Signing query, six new group elements are added in \( L_G \).

The total number of \( Q_G \), \( Q_T \) and \( Q_P \) queries is denoted by \( q_O \). Additionally, \( q_{KE} \), \( q_{TKU} \) and \( q_S \), respectively, represent the numbers of the Key extract query, Time key update query and Signing query. In the Query phase, \( A_I \) is allowed to request the queries at most \( q \) times. Therefore, we have \( |L_G| + |L_T| \leq 11 + 3q_O + 2q_{KE} + 2q_{TKU} + 6q_S \leq 6q \).

(2) The maximal degree of polynomials in \( L_G \) is 3 because of the following reasons:
- In the Setup phase, since these polynomials \( \Xi g \), \( \Xi U \), \( \Xi V \), \( \Xi W \), \( \Xi X \), \( \Xi Y \), \( \Xi Z \), \( \Xi SSK \) and \( \Xi TSK \) are new variates, they have degree 1. \( \Xi SPK \) and \( \Xi TPK \) have degree 2.
- In \( Q_G \), \( \Xi G_{Q.i.3} \) has the maximal degree of \( \Xi G_{Q.i.1} \) or \( \Xi G_{Q.i.2} \).
- In the Key extract query, \( \Xi T_{GKE.i.1} \), \( \Xi TID \) and \( \Xi USK_{ID} \) have degrees 1, 1 and 3, respectively.
- In the Time key update query, \( \Xi QTK_{ID.i.1} \), \( \Xi TTD \) and \( \Xi UTK_{ID.i} \) have degrees 1, 1 and 3, respectively.
- In the Signing query, \( \Xi Q_{KE} \) and \( \Xi \sigma_4 \) have degrees 1 and 3, respectively.

(3) The maximal degree of polynomials in \( L_T \) is 6 because of the following reasons:
- In the Setup phase, both \( \Xi SPK \) and \( \Xi TPK \) have degree 2.
- In \( Q_T \), \( \Xi T_{Q.i.3} \) has the maximal degree of \( \Xi T_{Q.i.1} \) or \( \Xi T_{Q.i.2} \).
- In \( Q_P \), the maximal degree of \( \Xi T_{P.i.1} \) in \( L_T \) is 6 because the maximal degree of polynomials in \( L_G \) is 3 and \( \Xi T_{P.i.1} = \Xi G_{P.i.1} \cdot \Xi G_{P.i.2} \).

If one of the following two cases occurs, we say that \( A_I \) wins \( G_{LR-RIBS} \):

Case 1. \( A_I \) discovers a collision of group elements in \( L_G \) or \( L_T \). Let \( n \) denote the total number of all variates in \( L_G \) and \( L_T \). Now, \( C \) selects \( n \) random values \( v_i \in Z_p^* \) for \( i = 1, \ldots, n \). In this case, there exist two polynomials \( \Xi G_i \) and \( \Xi G_j \), both in \( L_G \) or both in \( L_T \), that satisfy \( \Xi G_i(v_1, v_2, \ldots, v_n) = \Xi G_j(v_1, v_2, \ldots, v_n) \).
Case 2. \( A_I \) outputs a correct signature \((ID^*, T^*_i, msg^*, (\xi\sigma_1^*, \xi\sigma_2^*, \xi\sigma_3^*, \xi\sigma_4^*))\) that satisfies \( \mathcal{E}g \cdot \mathcal{E}\sigma_4^* = \mathcal{E}SPK + \mathcal{E}TPK + \mathcal{E}\sigma_1^* \cdot (\mathcal{E}U + TID^* \cdot \mathcal{E}V) + \mathcal{E}\sigma_2^* \cdot (\mathcal{E}W + TTD^* \cdot \mathcal{E}X) + \mathcal{E}\sigma_3^* \cdot (\mathcal{E}Y + msg^* \cdot \mathcal{E}Z) \), where \( TID^* = ID^* || \xi\sigma_1^* \) and \( TTD^* = ID^* || T_I || \xi\sigma_2^* \).

Let us evaluate \( A_I \)'s advantage of winning \( G_{LR-RIBS} \) without requesting Key extract leak query and Signing leak query. Note that \( A_I \) is a curious CRA or an outsider who can gain the user's time update key \( UTK_{ID, t} \) by the Time key update query. Hence, \( A_I \) has no need to request the Time key update leak query. Subsequently, \( A_I \)'s advantage in \( G_{LR-RIBS} \) with requesting two kinds of leak queries (Key extract leak query and Signing leak query) is evaluated.

**Without requesting two kinds of leak queries:** Under this circumstance, \( A_I \) may request all the queries in \( G_{LR-RIBS} \) except for the Key extract leak query and Signing leak query. In the following, let us discuss the probability that \( A_I \) wins \( G_{LR-RIBS} \) without requesting two kinds of leak queries.

Case 1. The probability that \( A_I \) discovers a collision of group elements in \( L_G \) or \( L_T \) is evaluated. Let \( \mathcal{E}G_i \) and \( \mathcal{E}G_j \) denote two distinct polynomials in \( L_G \). The collision probability is the probability that \( \mathcal{E}G_C = \mathcal{E}G_i - \mathcal{E}G_j \) is a zero polynomial, namely, \( \mathcal{E}G_C(v_1, v_2, \ldots, v_n) = 0 \). By Lemma 2, the probability of \( \mathcal{E}G_C(v_1, v_2, \ldots, v_n) = 0 \) is at most \( 3/p \) because there is no fraction leakage content \((\lambda = 0)\) and the maximal polynomial degree in \( L_G \) is 3. We have that the probability of discovering a collision in \( L_G \) is \((3/p)^{|L_G|/2}) \) since there are \(|L_G|/2 \) distinct pairs \((\mathcal{E}G_i, \mathcal{E}G_j)\) in \( L_G \). By similar arguments, the probability that \( A_I \) discovers a collision in \( L_T \) is \((6/p)^{|L_T|/2}) \). Moreover, the total number of group elements in \( L_G \) and \( L_T \) is at most 6q, namely, \(|L_G| + |L_T| \leq 6q \). Then the probability of Case 1, denoted by Pr[Case 1], satisfies the inequality

\[
\Pr[\text{Case 1}] \leq (3/p)^{|L_G|/2} + (6/p)^{|L_T|/2} \\
\leq (6/p)(|L_G| + |L_T|)^2 \\
\leq 216q^2/p.
\]

Case 2. Let us evaluate the probability that \( A_I \) outputs a signature \((ID^*, T^*_i, msg^*, (\xi\sigma_1^*, \xi\sigma_2^*, \xi\sigma_3^*, \xi\sigma_4^*))\) that satisfies \( \mathcal{E}f = \mathcal{E}SPK + \mathcal{E}TPK + \mathcal{E}\sigma_1^* \cdot (\mathcal{E}U + TID^* \cdot \mathcal{E}V) + \mathcal{E}\sigma_2^* \cdot (\mathcal{E}W + TTD^* \cdot \mathcal{E}X) + \mathcal{E}\sigma_3^* \cdot (\mathcal{E}Y + msg^* \cdot \mathcal{E}Z) - \mathcal{E}g \cdot \mathcal{E}\sigma_4^* = 0 \). The probability of outputting a correct signature is \( 5/p \) because the degree of \( \mathcal{E}f \) is at most 5.

Let \( \Pr_{A-I-W} \) denote the advantage that \( A_I \) wins \( G_{LR-RIBS} \) without requesting two kinds of leak queries. By Cases 1 and 2, we have the inequality

\[
\text{Adv}_{A-I-W} \leq \Pr[\text{Case 1}] + \Pr[\text{Case 2}] \\
\leq 216q^2/p + 5/p \\
\leq O(q^2/p).
\]
• With requesting two kinds of leak queries: Under this circumstance, $A_I$ is allowed to issue all the leak queries in $G_{LR-RIBS}$. In the $i$-th key extract leak query with $|f_{KE,i}| \leq \lambda$ and $|h_{KE,i}| \leq \lambda$, $A_I$ gains the fraction leakage content $Af_{KE,i} = f_{KE,i}(SSK_{i,1}, a, \gamma)$ and $Ah_{KE,i} = h_{KE,i}(SSK_{i,2}, a, \gamma, TI_{KE})$ discussed as below.

- $a, \gamma$: In each key extract query, $a$ and $\gamma$ are random values. Therefore, the leakage of $a$ or $\gamma$ is of no help to learn the system secret key SSK.
- $(SSK_{i,1}, SSK_{i,2})$: We have $SSK = SSK_{i-1,1} \cdot SSK_{i-1,2} = SSK_{i,1} \cdot SSK_{i,2}$. By the multiplicative blinding technique, the fraction leakage content of $SSK_{i-1,1}/SSK_{i-1,2}$ is independent of that of $SSK_{i,1}/SSK_{i,2}$. Hence, $A_I$ gains at most $\lambda$ bits of SSK.

- $TI_{KE}$: The temporary value $TI_{KE}$ is employed to compute the user’s secret key $USK_{ID}$. Since $A_I$ can obtain the entire $USK_{ID}$ except for $ID^*$, $TI_{KE}$ is helpless for $A_I$.

In the $k$-th Signing query of the user $ID$, by taking as input two leakage functions $f_{S,k}$ and $h_{S,k}$ with $|f_{S,k}| \leq \lambda$ and $|h_{S,k}| \leq \lambda$, $A_I$ gains the fraction leakage content $Af_{S,k} = f_{S,k}(USK_{ID,k,1}, UTK_{ID,k}, c, \delta)$ and $Ah_{S,k} = h_{S,k}(USK_{ID,k,2}, c, TI_{S})$ discussed as below.

- $c, \delta$: In each signing query, $c$ and $\delta$ are random values. Therefore, the leakage about $c$ and $\delta$ is of no help to learn the user’s secret key $USK_{ID}$.
- $(USK_{ID,k,1}, USK_{ID,k,2})$: We have $USK_{ID} = USK_{ID,k-1,1} \cdot USK_{ID,k-1,2} = USK_{ID,k,1} \cdot USK_{ID,k,2}$. By the multiplicative blinding technique, the fraction leakage content of $USK_{ID,k-1,1}/USK_{ID,k-1,2}$ is independent of that of $USK_{ID,k,1}/USK_{ID,k,2}$. Hence, $A_I$ gains at most $\lambda$ bits of $USK_{ID}$.

- $TI_{S}$: The temporary value $TI_{S}$ is used to generate the signature $\sigma_4$. Since $A_I$ can obtain the entire $\sigma_4$ by the Sign query, $TI_{S}$ is helpless for $A_I$.

Let $Adv_{A-I}$ be the advantage that $A_I$ wins $G_{LR-RIBS}$ with requesting the key extract leak query and signing leak query. For forging a correct signature, let us discuss the useful leakage content of the target user’s secret $USK$ and the system secret key SSK that consists of three events as below:

1. $ESSK$ denotes the event that $A_I$ knows the whole SSK by $Af_{KE,i}$ and $Ah_{KE,i}$, and its complement event is denoted by $\overline{ESSK}$.

2. $EUSK$ denotes the event that $A_I$ knows the whole $USK_{ID}$ by $Af_{S,k}$ and $Ah_{S,k}$, and its complement event is denoted by $\overline{EUSK}$.

3. $ESF$ denotes the event that $A_I$ forges a correct signature.

Hence, the advantage $Adv_{A-I}$ is $Pr[ESF]$ and satisfies the inequality

$$Adv_{A-I} = Pr[ESF] = Pr[ESF \land (ESSK \lor EUSK)] + Pr[ESF \land (\overline{ESSK} \land \overline{EUSK})]$$

$$\leq Pr[ESSK] + Pr[ESF \land EUSK] + Pr[ESF \land \overline{ESSK} \land \overline{EUSK}]$$

In our LR-RIBS scheme with CRA, the PKG employed SSK and a user’s information $ID||QK_{ID}$ to generate the user’s secret key $USK_{ID}$ by using the signature scheme in.
Galindo and Virek (2013). By Lemma 5 in Galindo and Virek (2013), we have $\Pr[\mathsf{ESSK}] \leq O((q^2/p) \cdot 2^{2\lambda})$. Next, $A_I$ may gain fractional content of $\mathsf{USK}_{ID}$ by the Signing leak query. Hence, $\Pr[\mathsf{ESF} \land \mathsf{EUSK}]$ is the probability that $A_I$ can get fractional content of $\mathsf{USK}_{ID}$ by $A\bar{f}_{S,k}$ and $A\bar{h}_{S,k}$. Thus, we can gain the probability $\Pr[\mathsf{ESF} \land \mathsf{EUSK}] \leq O((q^2/p) \cdot 2^{2\lambda})$.

Finally, the event $\mathsf{ESSK} \land \mathsf{EUSK}$ is that $A_I$ can gain fractional content of $(\mathsf{USK}_{ID,k,1}, \mathsf{USK}_{ID,k,2})$ by $A\bar{f}_{S,k}$ and $A\bar{h}_{S,k}$. In such a case, $A_I$ can gain at most $\lambda$ bits about $\mathsf{USK}_{ID}$, and so we have $\Pr[\mathsf{ESF} \land \mathsf{ESSK} \land \mathsf{EUSK}] \leq O((q^2/p) \cdot 2^\lambda)$. According to the events discussed above, we reach the inequality

$$\text{Adv}_{A-I} = \Pr[\mathsf{ESF}]$$

$$\leq \Pr[\mathsf{ESSK}] + \Pr[\mathsf{ESF} \land \mathsf{EUSK}] + \Pr[\mathsf{ESF} \land (\mathsf{ESSK} \land \mathsf{EUSK})]$$

$$\leq O\left((q^2/p)^{\cdot 2\lambda}\right) + O\left((q^2/p)^{\cdot 2\lambda}\right) + O\left((q^2/p)^{\cdot 2\lambda}\right).$$

Therefore, $\text{Adv}_{A-I} \leq O\left((q^2/p)^{\cdot 2\lambda}\right)$. Finally, by Lemma 2, $\text{Adv}_{A-I}$ is negligible if $\lambda < \log p - \omega(\log \log p)$. □

**Theorem 2.** In the GBG model, our LR-RIBS scheme with CRA possesses existential unforgeability under the UF-LR-RIBS-ACMA attack of Type II adversary (a revoked user).

*Proof.* Let $A_H$ denote a Type II adversary in the security game $G_{LR-RIBS}$ played with a challenger $C$. Hence, $A_H$ may acquire the user’s secret key $\mathsf{USK}_{ID}$ and time update key $\mathsf{UTK}_{ID,j}$ for any $ID$ at any period $T_j$, except for the target identity $ID^*$ at period $T^*_{I}$. For $G_{LR-RIBS}$ on the proposed LR-RIBS scheme with CRA, three phases (Setup, Query and Forgery) are described as below:

- **Setup phase:** The phase is the same with that of the proof in Theorem 1.
- **Query phase:** $A_H$ is allowed to issue the queries at most $q$ times adaptively as below.
  - **$Q_G$, $Q_T$, $Q_P$, Key extract query, Time key update query, Singing query and Signing leak query** are identical to these queries in Theorem 1. Note that $A_H$ is a revoked user who may acquire the user’s secret key $\mathsf{USK}_{ID}$ for any $ID$. Hence, $A_H$ has no need to request the Key extract leak query.
  - **Time key update leak query ($j$, $f_{TKU,j}$, $h_{TKU,j}$):** Upon receiving the $j$-th Key Time key update leak query with $f_{TKU,j}$ and $h_{TKU,j}$, $C$ returns the fraction leakage content $A\bar{f}_{TKU,j} = f_{TKU,j}(\mathsf{TSK}_{j,1}, b, \eta)$ and $A\bar{h}_{TKU,j} = h_{TKU,j}(\mathsf{TSK}_{j,2}, b, \eta, \mathsf{TI}_{TKU})$ to $A_H$. Note that for the $j$-th Time key update query, $A_H$ is allowed to request the Time key update leak query only once.
  - **Forgery phase:** $A_H$ outputs $(ID^*, T^*_I, msg^*, (\xi_{\sigma_1^*}, \xi_{\sigma_2^*}, \xi_{\sigma_3^*}, \xi_{\sigma_4^*}))$. $A_H$ is a revoked user who may request the Key extract query (ID$^*$) to obtain $\mathsf{USK}_{ID^*}$, but does not request the Time key update query (ID$^*$, $T^*_I$). $C$ first transforms $(\xi_{\sigma_1^*}, \xi_{\sigma_2^*}, \xi_{\sigma_3^*}, \xi_{\sigma_4^*})$ to gain the corresponding polynomials $\mathcal{E}\sigma_1^*$, $\mathcal{E}\sigma_2^*$, $\mathcal{E}\sigma_3^*$ and $\mathcal{E}\sigma_4^*$. $C$ sets $TID^* = ID^*||\xi_{\sigma_1^*}$ and $TTID^* = ID^*||T^*_I||\xi_{\sigma_2^*}$. If the equality $\mathcal{E}g \cdot \mathcal{E}\sigma_4^* = \mathcal{E}SPK + \mathcal{E}TPK + \mathcal{E}\sigma_1^* \cdot (\mathcal{E}U + TID^* \cdot \mathcal{E}V) + \mathcal{E}\sigma_2^* \cdot (\mathcal{E}W + TTD^* \cdot \mathcal{E}X) + \mathcal{E}\sigma_3^* \cdot (\mathcal{E}Y + msg^* \cdot \mathcal{E}Z)$ holds, we say that $A_H$ wins $G_{LR-RIBS}$. 
By the same arguments in Theorem 1, the total number of group elements in $L_G$ and $L_T$ is at most $6q$, namely, $|L_G| + |L_T| \leq 6q$. The maximal polynomial degrees in $L_G$ and $L_T$ are 3 and 6, respectively. Let us evaluate $A_H$’s advantage winning $G_{LR-RIBS}$ without requesting Time key update leak query and Signing leak query. Subsequently, $A_H$’s advantage in $G_{LR-RIBS}$ with requesting two kinds of leak queries is evaluated.

- **Without requesting two kinds of leak queries:** Let $Pr_{A_1 \rightarrow LR}$ denote the advantage that $A_H$ wins $G_{LR-RIBS}$ without requesting two kinds of leak queries. By the similar discussions as Theorem 1, we have the inequality

$$Adv_{A_1 \rightarrow LR} \leq Pr[Case\ 1] + Pr[Case\ 2] \leq 216q^2/p + 5/p \leq O(q^2/p).$$

Hence, $Adv_{A_1 \rightarrow LR}$ is negligible if $q = \text{poly}(\log p)$.

- **With requesting two kinds of leak queries:** Under this circumstance, $A_H$ is allowed to request all queries in $G_{LR-RIBS}$. For the $j$-th time key update leak query with $f_{TKU,j}$ and $h_{TKU,j}$ that satisfy $|f_{TKU,j}| \leq \lambda$ and $|h_{TKU,j}| \leq \lambda$, $A_H$ can gain the fraction leakage content $A_{f_{TKU,j}} = f_{TKU,j}(TSK_{j,1}, b, \eta)$ and $A_{h_{TKU,j}} = h_{TKU,j}(TSK_{j,2}, b, \eta, TI_{TKU})$ discussed below:

  - **$b, \eta$:** In each time key update query, $b$ and $\eta$ are random values. Therefore, the leakage about $b$ and $\eta$ is of no help to learn the time secret key $TSK$.
  - **$(TSK_{j,1}, TSK_{j,2})$:** For the time secret key $TSK$, we have $TSK = TSK_{j-1,1} \cdot TSK_{j-1,1} = TSK_{j,1} \cdot TSK_{j,2}$. By the multiplicative blinding technique, the fraction leakage content of $TSK_{j-1,1}/TSK_{j-1,2}$ is independent of that of $TSK_{j,1}/TSK_{j,2}$. Thus, $A_H$ gains at most $\lambda$ bits of $TSK$.
  - **$TI_{TKU}$:** The temporary value $TI_{TKU}$ is employed to generate user’s time key $UTK_{ID,t}$. Since $A_H$ can obtain the whole $UTK_{ID,t}$ by the time key update query, $TI_{TKU}$ is helpless for $A_H$.

For the $k$-th Signing leak query of the user $ID$, by taking as input two leakage functions $f_{S,k}$ and $h_{S,k}$ such that $|f_{S,k}| \leq \lambda$ and $|h_{S,k}| \leq \lambda$, $A_H$ gains the fraction leakage content $A_{f_{S,k}} = f_{S,k}(USK_{ID,k,1}, UTK_{ID,t}, c, \delta)$ and $A_{h_{S,k}} = h_{S,k}(USK_{ID,k,2}, c, TI_{S})$. Indeed, a revoked user has possessed the user secret key $USK_{ID}$. In particular, since the user’s time update key $UTK_{ID,t}$ is not generated, the Signing leak query does not leak any content.

Let $Adv_{A_1 \rightarrow LR}$ be the advantage that $A_H$ wins $G_{LR-RIBS}$ with requesting the time key update query. Since $A_H$ simulates a revoked user, she/he can obtain the target user’s secret key $USK_{ID}$ $D$. For forging a correct signature, let us discuss the helpful leakage content about the target user’s time key $UTK_{ID,t}$ that consists of two events as below:

(1) $ETSK$ denotes the event that $A_H$ gains the whole $TSK$ by $A_{f_{TKU,j}}$ and $A_{h_{TKU,j}}$, and $ETSK^c$ denotes the complement event of $ETSK$.
(2) $ESF$ denotes the event that $A_{II}$ forges a correct signature. Hence, the advantage $Adv_{A-II}$ is $\Pr[ESF]$ and satisfies the inequality

$$Adv_{A-II} = \Pr[ESF]$$

$$= \Pr[ESF \land ETSK] + \Pr[ESF \lor ETSK]$$

$$\leq \Pr[ETSK] + \Pr[ESF \land ETSK].$$

In the Time key update phase of our scheme, the CRA employed the time secret key $TSK$ and a user’s content $TTD = ID || T|| QTK_{ID,t}$ to generate the user’s secret key $UTK_{ID,t}$ by using the signature scheme in Galindo and Virek (2013). The probability $\Pr[ETSK]$ is identical to $\Pr[ESSK]$ in Theorem 1 so that we have $\Pr[ETSK] \leq O((q^2/p)^*2^{2\lambda})$. Next, the event is that $A_{II}$ can gain at most $\lambda$ bits of $(TSK_j, 1, TSK_j, 2)$, we have $\Pr[ESF \land -ETSK] \leq O((q^2/p)^*2^{\lambda})$. According to the events discussed above, we reach the inequality

$$Adv_{A-II} \leq \Pr[ETSK] + \Pr[ESF \land ETSK]$$

$$\leq O\left((q^2/p)^*2^{2\lambda}\right) + O\left((q^2/p)^*2^{\lambda}\right).$$

Therefore, $Adv_{A-II} \leq O((q^2/p)^*2^{2\lambda})$. Finally, by Lemma 2, $Adv_{A-II}$ is negligible if $\lambda < \log p - \omega(\log \log p)$. □

6. Performance Comparisons

Here, we compare the performance between previously proposed RIBS schemes (Tsai et al., 2013b; Jia et al., 2017) and our LR-RIBS scheme with CRA. In the following, four notations are defined respectively to represent four time-consuming operation costs of bilinear groups:

- $T_{bp}$: The executing cost of a bilinear map $\hat{e} : G \times G \rightarrow G_T$.
- $T_{me}$: The executing cost of a scalar multiplication on an additive cycle group $G$ or an exponentiation operation on a multiplicative cycle group $G$.
- $T_{ex}$: The executing cost of an exponentiation operation on a multiplicative cycle group $G_T$.
- $T_{mh}$: The executing cost of a map-to-point hash function operation in $G$.

Indeed, the cost of the operation (additive/multiplicative) on an (additive/multiplicative) cyclic group $G$ is smaller than $T_{bp}$, $T_{me}$, $T_{ex}$ and $T_{mh}$ (Scott, 2011; Lynn, 2015), and so is negligible. The simulation experiences (Lynn, 2015) on a PC and a smartphone are employed as the benchmark costs of $T_{bp}$, $T_{me}$, $T_{ex}$ and $T_{mh}$. The simulation platform on both the PKG and CRA sides is an Intel Core-2 Quad Q6600 PC with 2.4 GHz processor and Ubuntu OS. Meanwhile, the simulation platform on the user side is an HTC Desire HD-A9191 smartphone with Qualcomm 1 GHz processor and Android 2.2 OS. Additionally, under the same security level with 1024-bit RSA system, an elliptic curve over a finite
field $E(F_q)$ are employed for bilinear pairing groups with a prime order $p$, where $p$ and $q$ are 160 and 512 bits, respectively. Table 1 lists the executing cost (in milliseconds) of $T_{bp}$, $T_{me}$, $T_{ex}$ and $T_{mh}$ on both a PC and a smartphone in Lynn (2015).

Table 2 lists the performance comparisons between two previously proposed RIBS schemes (Tsai et al., 2013b; Jia et al., 2017) and our LR-RIBS scheme with CRA in terms of resisting side-channel attacks, overall unbounded leakage property, outsourced revocation authority and the computation costs of four phases. In Tsai et al. (2013b), the PKG is responsible to carry out both the Key extract and Time key update phases. On the other hand, the scheme in Jia et al. (2017) and our scheme employ a CRA to outsource the functionality of user revocation. Note that the executing costs of both the Key extract and Time key update phases are measured under a PC while the executing costs of both the Signing and Verifying phases are measured under a smartphone.

By Table 2, although performing worse than the other two schemes in the computation costs, our scheme is still well suited for a smartphone with limited computing capability. We should emphasize that our scheme can resist side-channel attacks with overall unbounded leakage property, but the other two cannot.
7. Conclusions

In the continual leakage model, we have defined a novel adversary model of LR-RIBS schemes with CRA. In the adversary model, Type I adversary (a curious CRA or an outsider) is allowed to extract fractional content of the target signer’s secret key and the PKG’s system secret key. Also, Type II adversary (a revoked user) is allowed to extract fractional content of the CRA’s time secret key. We have proposed the first LR-RIBS scheme with CRA and it possesses the overall unbounded leakage property. In the GBG model, security analysis demonstrated that the proposed LR-RIBS scheme with CRA is secure against Types I and II adversaries under the continual leakage model. Performance comparisons demonstrated that the proposed LR-RIBS scheme with CRA requires some additional computation costs than the previously proposed RIBS schemes. This point is that our scheme not only can resist side-channel attacks, but also is still suitable for mobile devices with limited computing capability.

Funding

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

References

Leakage-Resilient Revocable Identity-Based Signature with Cloud Revocation Authority


J.-D. Wu received the BS degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2006. He received the MS degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2008. He is currently a PhD candidate at the Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include applied cryptography and pairing-based cryptography.

Y.-M. Tseng is currently the dean of Science College and a professor at the Department of Mathematics, National Changhua University of Education, Taiwan. He is a member of IEEE Computer Society, IEEE Communications Society and the Chinese Cryptology and Information Security Association (CCISA). He has published over one hundred scientific journals and conference papers on various research areas of cryptography, security and computer network. His research interests include cryptography, network security, computer network and mobile communications. He serves as an editor of several international journals.

S.-S. Huang is currently a professor at the Department of Mathematics, National Changhua University of Education, Taiwan. His research interests include number theory, cryptography, and network security. He received his PhD from the University of Illinois at Urbana-Champaign in 1997 under the supervision of professor Bruce C. Berndt.

T.-T. Tsai is currently a senior engineer at HON HAI Technology Group, Taiwan. His research interests include applied cryptography and pairing-based cryptography. He received the PhD degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2014, under the supervision of professor Yuh-Min Tseng.