ITC 2/47

Journal of Information Technology and Control
Vol. 47 / No. 2 / 2018
pp. 363-386
DOI 10.5755/j01.itc.47.2.17847

Leakage-Resilient Certificateless Signature Under Continual Leakage Model

Received 2017/09/12 | Accepted after revision 2018/04/15

http://dx.doi.org/10.5755/j01.itc.47.2.17847

Leakage-Resilient Certificateless Signature Under Continual Leakage Model

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

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

In the past, the security notions of cryptography were modeled under the assumption that private (or secret) keys are completely hidden to adversaries. Nowadays, these security notions could be insufficient due to a new kind of threat, called "side-channel attacks", by which an adversary obtains partial information of private (or secret) keys via employing specific properties resulting from physical implementations of cryptographic schemes. In order to resist such side-channel attacks, numerous leakage-resilient cryptographic schemes have been proposed. However, there is little work on studying leakage-resilient certificateless cryptographic schemes. In this article, we propose the *first* leakage-resilient certificateless signature (LR-CLS) scheme under the continual leakage model. In the generic bilinear group model, we demonstrate that our scheme possesses existential unforgeability against adaptive chosen-message attacks for both Type I and Type II adversaries. Finally, performance analysis is made to demonstrate that the proposed LR-CLS scheme is suitable for resource-constrained devices.

KEYWORDS: Side-channel attack, Certificateless signature, Leakage-resilience, Provable security.

1. Introduction

In the conventional public key settings [14, 33], a certificate is employed to validate the mapping between a user's identity and her/his associated public key. In order to remove the certificate usage, Shamir [36] introduced the concept of identity (ID)-based public key setting. Based on Shamir's concept, Boneh and Franklin [7] proposed the first practical construction of ID-based encryption (IBE) from bilinear pairings. In an ID-based public key setting, identity information of a user is viewed as the user's public key, by which a trusted private key generator (PKG) can produce and send the corresponding private key to the user. Under this circumstance, the PKG knows private keys of all the users. In other words, all the ID-based public key settings suffer from the key escrow problem in the sense that the PKG may decrypt all the ciphertexts or sign the messages on behalf of all users.

In 2003, Al-Riyami and Paterson [1] proposed a new public key paradigm, termed certificateless public key setting (CL-PKS), to resolve the key escrow problem mentioned above. In CL-PKS, a user's private key consists of two components, namely, an initial key and a secret key. In addition, there exists a semi-trusted third party, called the key generation center (KGC), who is responsible to produce the user's initial key by its system secret key and the user's identity information. Meanwhile, the user randomly chooses a secret key and computes the corresponding public key without requiring any certificate. Hence, the KGC cannot access the user's private key due to the lack of the secret key generated by the user. Therefore, the CL-PKS not only resolves the key escrow problem in ID-based public key settings but also removes the certificate management in conventional public key public key settings. In the past decade, the research on CL-PKS has great progress and numerous cryptographic schemes have been proposed [19-24, 29, 30, 39, 40, 44, 46].

The security notions for these public key settings mentioned above (including conventional, ID-based and certificateless) were modeled under the assumption that both the system's and users' private (or secret) keys are completely hidden to an adversary. Nowadays, these security notions could be insufficient due to a new kind of threat, called "side-channel attacks", such as fault attack [4, 6], power analysis [27], timing attack [10, 28], etc. For side-channel attacks, an adversary may obtain partial information of private (or secret) keys by employing specific properties resulting from physical implementations of cryptographic schemes. Thus, even if a cryptographic scheme was proven secure in an adversary model without addressing side-channel attacks, the cryptographic scheme could be broken in an environment where an adversary may obtain the partial information of private (or secret) keys. Leakage-resilient cryptography provides a solution to counteract side-channel attacks. Very recently, the study of leakage-resilient cryptography has received significant attention. Based on conventional public key settings, numerous leakage-resilient public key encryption schemes [2, 11, 26, 32] and leakage-resilient signature schemes [3, 15, 16, 18, 25, 38] have been proposed.

1.1. Related Work

The security notion of leakage-resilient cryptography is that a cryptographic scheme is still secure even if the partial leakage information of the private (or secret) keys involved in the scheme is visible to the adversary. In order to represent the leakage resilience of cryptographic schemes, adversary models must define the capabilities of an adversary leaking the partial information of the private (or secret) keys. For representing the leakage ability of an adversary, there are two kinds of leakage models, namely, ***bounded leakage model*** and ***continual leakage model***, which are described as follows. Typically, a cryptographic scheme consists of several computation rounds. In leakage-resilient cryptography, a leakage function f is given and $f(\tau)$ is viewed as the leakage information, where τ indicates the data (including permanent and temporary secret values) accessed during the current computation round. The output length of f is restricted to λ bits, that is, the leakage information of each computation round is bounded. On the other hand, if the total leakage information of a cryptographic scheme is unbounded, the whole private key would completely be revealed to the adversary so that it will injure the security of the cryptographic scheme. Hence, several leakage-resilient cryptographic schemes [3, 25] make a restriction on the overall leakage information to be bounded. This is called the bounded leakage model. However, this restriction is not practical. In recently proposed leakage-resilient cryptographic schemes, the continual leakage model is the most accredited model for leakage ability of an adversary, which provides the overall unbounded leakage property than the bounded leakage model. The continual leakage model possesses the following properties [9, 12, 18]:

- ***Only computation leakage***: Only temporary and permanent secret values currently accessed in a computation round could be leaked to a side-channel adversary.
- ***Bounded leakage of single observation***: The secret information leaked by single computation round (or called an observation) is bounded to λ bits. This property bounds the leakage of each computation round to some fraction of secret information.
- ***Independent leakage***: The leakage information of each computation round is independent of the other computation rounds.

- ***Overall unbounded leakage***: The overall amount of leakage information is assumed to be unbounded. Hence, after (or before) each computation round, the secret value must be refreshed (updated). It is obvious that the leakage bound can be restricted between any two successive secret value refreshes.

Based on conventional public key settings, several leakage-resilient public key encryption and signature schemes were proposed under the continual leakage model, which are surveyed as follows. In 2010, Kiltz and Pietrzak [26] proposed a leakage-resilient public key encryption in the generic bilinear group (GBG) model [5]. The GBG model is viewed as a kind of security proving technique, which will be defined in Section 2. It is worth mentioning, that the GBG model may be employed in the security proofs of cryptographic schemes under non-leakage model, bounded leakage model and continual leakage model. Following Kiltz and Pietrzak's technique in the GBG model, Galindo and Vivek [18] proposed a secure leakage-resilient signature scheme. Afterwards, based on Boneh *et al.*'s short signature [8] and GBG model, Tang *et al.* [38] presented an improved leakage-resilient signature scheme which reduces one exponential computation compared with Galindo and Vivek's scheme [18]. The security of Tang *et al.*'s scheme is based on both the GBG model and the random oracle model. In 2016, based on the generic bilinear group, Galindo *et al.* [17] also presented and implemented a new leakage-resilient ElGamal public key encryption scheme, which is the newest implementation for leakage-resilient protocols in the GBG model.

In ID-based public key settings, Brakerski *et al.* [9] proposed the first leakage-resilient ID-based encryption (LR-IBE) scheme under the continual leakage model. Afterwards, Yuen *et al.* [45] proposed an improved LR-IBE scheme to improve performance. In 2016, the first leakage-resilient ID-based signature (LR-IBS) was proposed by Wu *et al.* [42]. Under the continual leakage model, their LR-IBS scheme allows an adversary to learn partial information of both the system secret key in the key extract phase and the user's private key in the signing phase during the entire lifetime of the system. Also, their LR-IBS scheme possesses existential unforgeability against ID and adaptive chosen message (EUF-CMA) attacks. Nevertheless, Wu *et al.*'s LR-IBS scheme is constructed under the ID-based public key settings, so it suffers from the key escrow problem mentioned earlier.

1.2. Contribution and Organization

In the past, there is little work on studying the design of leakage-resilient certificateless cryptographic schemes. In 2013, Xiong *et al.* [43] proposed the first leakage-resilient certificateless public key encryption scheme (with various leakage conditions) for Type I adversary (outsider) and Type II adversary (honest-but-curious KGC), following the classification in traditional certificateless public key encryption [24]. However, Xiong *et al.*'s scheme did not resist adaptive chosen-ciphertext key-leakage attacks (IND-KL-CCA2). In 2016, Zhou *et al.* [47] improved Xiong *et al.*'s scheme to propose an IND-KL-CCA2-secure certificateless signcryption scheme based on bilinear pairings. Both Xiong *et al.*'s and Zhou *et al.* schemes are secure under the bounded leakage model, but not under the continual leakage model.

Up to now, no work has been done on the design of leakage-resilient certificateless signature (LR-CLS). In this article, we will propose the *first* leakage-resilient certificateless signature scheme under the continual leakage model. We first define the security notions for LR-CLS schemes under the continual leakage model. The security notions include two kinds of attackers, namely, Type I adversary (outsider) and Type II adversary (honest-but-curious KGC). Both kinds of adversaries are extended from the security notions of traditional certificateless signature (CLS) schemes by adding the key leakage queries. Under the continual leakage model, the proposed LR-CLS scheme is allowed to leak partial information of the system secret key in the initial key extract phase and the user's private key in the signing phase. In the generic bilinear group model, we demonstrate that our scheme possesses existential unforgeability against adaptive chosen-message attacks for both Type I and Type II adversaries. Finally, performance analysis is made to demonstrate that the proposed LR-CLS scheme is suitable for resource-constrained devices.

The rest of the paper is organized as follows. In Section 2, we present preliminaries. The framework and security notions of LR-CLS schemes are defined in Section 3, while a concrete LR-CLS scheme is proposed in Section 4. The security of the proposed LR-CLS scheme is formally proved in Section 5. In

Section 6, we demonstrate the performance analysis of the proposed LR-CLS scheme. Conclusions are drawn in Section 7.

2. Preliminaries

In this section, we briefly introduce the concepts of bilinear groups [7, 35, 41], the notions of the generic bilinear group model [5, 18, 42] and the entropy.

2.1 Bilinear Groups

Let G denote a multiplicative group of large prime order p while G_T is also a multiplicative cyclic group with the same order. Assume that g is an arbitrary generator of G. An admissible bilinear pairing is a map e: $G\times G\rightarrow G_T$ which satisfies the following three properties:

- *Bilinearity*: $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$, where $g_1, g_2 \in G$ and $a, b\in Z_p^*$.
- *Non-degeneracy*: $e(g, g) \neq 1$, where $g\in G$.
- *Computability*: $e(g_1, g_2)$ can be efficiently computed, where $g_1, g_2 \in G$.

Meanwhile, G is called a bilinear group and G_T is the target group of the admissible bilinear map e. A reader can refer to previous literatures such as [7, 35, 41] for a more comprehensive description of groups, maps and other parameters.

2.2. Generic Bilinear Groups Model

In 1997, Shoup [37] introduced the notions of the generic group model which is viewed as a kind of security proving technique for cryptographic schemes. In this model, the adversary is only given access to a randomly chosen encoding of a group controlled by a challenger. Basically, the model includes an oracle that executes the group operation [31] which takes as input two group elements a and b, and outputs $a*b$, where * denotes the group operation. One of the main usages of the generic group model is to analyze *computational hardness assumption,* i.e. the discrete logarithm problem in a group. If an adversary can efficiently find a collision encoding of a group operation, it is said to solve the computational hardness assumption.

Boneh *et al.* [5] presented the generic bilinear group (GBG) model, which is an extension of the generic group model. In the generic bilinear group model, there are two multiplicative groups G and G_T. Both groups G and G_T have their own multiplication operation. Additionally, there exists a bilinear pairing operation to map two elements of G to one element of G_T. Therefore, the elements of G and G_T are encoded by two random injective maps $\varepsilon: Z_p\rightarrow\Xi$ and $\varepsilon_T: Z_p\rightarrow\Xi_T$, respectively, where Ξ and Ξ_T are bit strings while $\Xi\cap\Xi_T=\phi$ and $|\Xi|=|\Xi_T|=p$. The operations in G, G_T and the evaluation of the bilinear map e are performed by three public oracles O, O_T and O_P, respectively. For any $a, b\in Z_p^*$, we have the following properties:

- $O\,(\varepsilon(a), \varepsilon(b)) \rightarrow \varepsilon(a+b \bmod p)$.
- $O_T\,(\varepsilon_T(a), \varepsilon_T(b)) \rightarrow \varepsilon_T(a+b \bmod p)$.
- $O_P\,(\varepsilon(a), \varepsilon(b)) \rightarrow \varepsilon_T(ab \bmod p)$.

Note that if g is a generator of the group G, we have $g = \varepsilon(1)$ and $g_T=e(g, g) = \varepsilon_T(1)$.

2.3. Entropy

Entropy is a measure of the number of possible microscopic states (or microstates) of a system in thermodynamic equilibrium. The interpretation of entropy in statistical mechanics is the measure of uncertainty. Let X be a finite random variable and Pr be the associated probability distribution. Min-entropy is a way of measuring the worst-case predictability of a random variable. We define two kinds of min-entropies as follows:

1. The min-entropy of a finite random variable X is defined as $H_\infty(X) = -\log_2(\max_x \Pr[X = x])$.
2. The average conditional min-entropy of X under a given correlated random variable Z is defined as $\widetilde{H}_\infty(X\mid Z) = -\log_2(E_{z\leftarrow Z}[\max_x \Pr[X = x \mid Z = z]])$.

Dodis *et al.* [13] provided the following result on the entropy.

Lemma 1. Let f: $X\rightarrow\{0,1\}^{\lambda'}$ be a leakage function on a given random variable X, where λ' is a fixed length. We have $\widetilde{H}_\infty(X\mid f(X)) \geqq H_\infty(X)-\lambda'$.

Furthermore, Galindo and Vivek [18] presented a result (Lemma 2 below) to measure the probability distribution of polynomial under the advantage of leakage information, which is a variant of the Schwartz-Zippel lemma [34, 48]. Based on Lemma 2, a direct result (Corollary 1 below) is obtained.

Lemma 2. Let $F\in Z_p[X_1, X_2,..., X_n]$ be a non-zero polynomial of total degree at most d. Let P_i (for i=1, 2, ..., n) be probability distributions on Z_p while $H_\infty(P_i) \geqq$

$\log p - \lambda'$ holds, where $0 \leqq \lambda' \leqq \log p$. If $x_i \xleftarrow{P_i} Z_p$ (for i=1, 2, ..., n) are independent, we have $\Pr[F(x_1, x_2, ..., x_n)=0] \leqq (d / p)2^{\lambda'}$.

Corollary 1. If $\lambda' < \log p - \omega(\log\log p)$, then $\Pr[F(x_1, x_2 ..., x_n)= 0]$ is negligible (in $\log p$).

3. Framework and Security Notions

In this section, we define the framework and security notions of leakage-resilient certificateless signature (LR-CLS) schemes under the continual leakage model.

Al-Riyami and Paterson [1] presented the concept of the certificateless public key setting (CL-PKS) and proposed a concrete certificateless signature (CLS) scheme. In CL-PKS, the key generation center (KGC) with a system secret key is responsible to produce the user's initial key, while the user randomly chooses a secret key and computes the corresponding public key. However, formal security notions of CLS schemes were not given until the work of Yum and Lee [46] and Huang et al. [22]. Later, Hu et al. [19] enhanced the definitions in [22, 46] to permit stronger queries for adversaries. Since then, Hu et al.'s security model formalizes the security notions of CLS schemes. In this model, there are two kinds of adversaries, namely, Type I (outsider), Type II (honest-but-curious KGC). A Type I adversary A_I acts as an outsider, without the system secret key, who can replace the public key of any entity with another of her/his own choice. In other words, the outsider may obtain the secret key of any entity. A Type II adversary A_{II} models an honest-but-curious KGC that owns the system secret key, but cannot perform any public key replacement. That is, the honest-but-curious KGC knows the initial key of any entity.

Next, we introduce the so-called *stateful* from the continual leakage model in [26]. A cryptographic scheme under the continual leakage model is called stateful if the private/secret key must be updated before (or after) executing the cryptographic algorithm while the associated public key remains fixed. To be stateful, each private/secret key must be divided into two parts and stored in different parts of the memory. Hence, for a CLS scheme, we separate the initial key extract algorithm, as well as the signing algorithm, into two steps. In addition, the system secret key and user's private key are separated into two parts, respectively. That is, the two steps of the signing algorithm are carried out by the two parts of the private key, respectively, while the two steps of the initial key extract algorithm are carried out by the two parts of the system secret key.

3.1. Framework of LR-CLS

Following Hu *et al*.'s framework and security notions for CLS schemes, we define a new framework of LR-CLS schemes under the continual leakage model. A LR-CLS scheme consists of the following seven algorithms:

- ***Setup***: This algorithm is run by the key generation center (KGC) that takes a security parameter as input, and outputs the first system secret key ($S_{0,1}$, $S_{0,2}$) and the public parameters PP. PP is made public and available for all the other algorithms.
- ***Initial key extract***: The KGC is responsible to run this algorithm which consists two sub-algorithms *Extract*-1 and *Extract*-2. For the i-th round along with a user's identity ID, the KGC uses the current system secret key ($S_{i-1,1}$, $S_{i-1,2}$) to generate the first initial key (DID_0, QID) of the user while updating the current system secret key with ($S_{i,1}$, $S_{i,2}$). Two sub-algorithms are defined as follows:
  - ***Extract*-1**: Given $S_{i-1,1}$ and the user's identity ID, the algorithm chooses a random number γ, and outputs $S_{i,1}$, temporary information TI_{IE} and QID.
  - ***Extract*-2**: Given $S_{i-1,2}$ and TI_{IE}, the algorithm outputs $S_{i,2}$ and DID_0.

  The PKG then sends the initial key (DID_0, QID) to the user.
- ***Set secret value***: A user with identity ID runs this algorithm to set the secret key of the user. The algorithm randomly selects a secret key SID_0, computes the partial public key RID, and then returns SID_0 and RID.
- ***Set private key***: This deterministic algorithm is run by a user with identity ID and takes as input the user's initial key (DID_0, QID) and secret key SID_0, and returns the user's private key (($DID_{0,1}$, $DID_{0,2}$), ($SID_{0,1}$, $SID_{0,2}$)).
- ***Set public key***: This deterministic algorithm is

run by a user with identity ID and takes as input the user's initial key (DID_0, QID) and the partial public key RID, and returns the user's public key PID=(QID, RID).

- ***Sign***: A user with identity ID runs this algorithm which consists of two sub-algorithms *Sign*-1 and *Sign*-2. For the j-th *Sign* round, the user employs the current private key (DID_{j-1}=($DID_{j-1,1}$, $DID_{j-1,2}$), SID_j=($SID_{j-1,1}$, $SID_{j-1,2}$)) to generate a signature σ while updating the current private key with (DID_j=($DID_{j,1}$, $DID_{j,2}$), SID_j=($SID_{j,1}$, $SID_{j,2}$)). Two sub-algorithms are presented as follows:
  - ***Sign*-1**: Given $DID_{j-1,1}$ and $SID_{j-1,1}$ of the current private key and a message m, the algorithm chooses a random number η, and outputs $DID_{j,1}$, $SID_{j,1}$ and the temporary information TI_S.
  - ***Sign*-2**: Given $DID_{j-1,2}$ and $SID_{j-1,2}$ of the user's current private key and the temporary information TI_S, the algorithm outputs $DID_{j,2}$, $SID_{j,2}$ and a signature σ.
- ***Verify***: This deterministic algorithm takes as input a message m, a signature σ, a user identity ID with PID, and outputs either "accept" or "reject".

3.2. Security Notions of LR-CLS

In the presence of the continual leakage model, an adversary A can get leakage information from four sub-algorithms, namely, *Extract*-1, *Extract*-2, *Sign*-1 and *Sign*-2. In order to represent the leakage information, we use two leakage functions $f_{IE,i}$ and $h_{IE,I}$, respectively, to model the adversary's ability in *Extract*-1 and *Extract*-2 of the i-th *Initial key extract* round. Meanwhile, two leakage functions $f_{S,j}$ and $h_{S,j}$ are used to model the adversary's ability in *Sign*-1 and *Sign*-2 of a user's j-th *Sign* round. Note that four leakage functions $f_{IE,i}$, $h_{IE,i}$, $f_{S,j}$ and $h_{S,j}$ can be efficiently computed with bounded output length $\{0, 1\}^\lambda$ (λ is the leakage parameter), namely, $|f_{IE,i}|$, $|h_{IE,i}|$, $|f_{S,j}|$, $|h_{S,j}| \leq \lambda$, where $|func|$ denotes the output length of the function *func*. The outputs of four leakage functions are defined as follows.

- $\Lambda f_{IE,i}$=$f_{IE,i}$ ($S_{i-1,1}$, *parameters*).
- $\Lambda h_{IE,i}$=$h_{IE,i}$ ($S_{i-1,2}$, TI_{IE}, *parameters*).
- $\Lambda f_{S,j}$=$f_{S,j}$($DID_{j-1,1}$, $SID_{j-1,1}$, *parameters*).
- $\Lambda h_{S,j}$=$h_{S,j}$($DID_{j-1,2}$, $SID_{j-1,2}$, TI_S, *parameters*).

Here, parameters are the random values involved in the computation of each *Extract* and *Sign* round. Note that TI_{IE} and TI_S are the outputs of *Extract*-1 and *Sign*-1, respectively.

In the LR-CLS scheme under the continual leakage model, the security notions include two kinds of attackers, namely, Type I attacker (outsider) and Type II attacker (honest-but-curious KGC). Both kinds of attackers are extended from the security notions of traditional certificateless signature (CLS) schemes [19, 22, 46] by adding the key leakage queries. In such a scheme, the system secret key is used to generate the user's initial key by the KGC and the user's private key is used to generate the signature by the signer. Hence, under the continual leakage model, LR-CLS schemes are allowed to leak partial information of the system secret key in the *Initial key extract* phase and the user's private key in the *Sign* phase.

The adversary model of LR-CLS schemes under the continual leakage model consists of two kinds of adversaries, namely, Type I (outsider), Type II (honest-but-curious KGC).

- Type I adversary (outsider): An adversary of this type cannot access the system secret key, but she/he can replace the public key of any entity with another of her/his own choice. In other words, the adversary may obtain the secret key of any entity. Meanwhile, the adversary may obtain not only the leakage information of a user's initial key of the private key in the *Sign* phase, but also the leakage information of the KGC's system secret key in the *Initial key extract* phase.
- Type II adversary (honest-but-curious KGC): An adversary of this type is an honest-but-curious KGC who has access to the system secret key, but cannot perform any public key replacement. That is, the honest-but-curious KGC knows the initial key of any entity while obtaining the leakage information of a user's secret key of the private key in the *Sign* phase.

In the following, we employ a security game to model security notions of LR-CLS schemes under the continual leakage model. The security game describes the interactions between a challenger and an adversary.

Definition 1. A LR-CLS scheme possesses existential unforgeability against adaptive chosen-message attacks under continual leakage model (UF-LR-CLS-ACMA) if no probabilistic polynomial-time adversary A (including Types I and II adversaries) has a non-negligible advantage in the following UF-LR-CLS-ACMA game played with a challenger C. The advantage of the adversary A is defined as the probability that A wins the games. Such an adversary A is referred as an UF-LR-CLS-ACMA adversary.

- *Setup*. The challenger C takes as input a security parameter and runs the *Setup* algorithm to produce the first system secret key ($S_{0,1}$, $S_{0,2}$) and a list of public parameters *PP*. *PP* is given to the adversary A. Meanwhile, if A is of Type II adversary, C gives the system secret key ($S_{0,1}$, $S_{0,2}$) to the adversary A. If A is of Type I adversary, the system secret key ($S_{0,1}$, $S_{0,2}$) is kept secret by the challenger C.
- *Queries*. The adversary A can adaptively make numerous queries to the challenger C as follows.
  - *Initial key extract query*(ID). For the i-th *Extract* round, upon receiving this query along with a user's identity ID, the challenger C uses the current system secret key ($S_{i-1,1}$, $S_{i-1,2}$) to generate the first initial key (DID_0, QID) of the user while updating the current system secret key with ($S_{i,1}$, $S_{i,2}$) by running two sub-algorithms *Extract*-1 and *Extract*-2. Finally, C sends (DID_0, QID) to A.
  - *Initial key extract leak query* ($f_{IE,i}$, $h_{IE,i}$, i): For the i-th *Extract query, A* can issue the *Initial key extract leak query* only once by providing two leakage functions $f_{IE,i}$ and $h_{IE,i}$. C computes the leakage information of ($\Lambda f_{IE,i}$, $\Lambda h_{IE,i}$) and sends it to A. Here we assume that two leakage functions $f_{IE,i}$ and $h_{IE,i}$ can be efficiently computed with bounded length output in $\{0, 1\}^{\lambda}$, namely, $|f_{IE,i}|$, $|h_{IE,i}| \leq \lambda$.
  - *Public key retrieve query* (ID). When A issues this query along with an identity ID, the challenger C returns the corresponding public key PID=(QID, RID) to A.
  - *Public key replace query* (ID, PID'=(QID', RID')). Upon receiving this query, the user's original public key is replaced with PID'=(QID', RID') and the challenger C records the replacement.
  - *Secret key extract query* (ID). When A issues this query along with an identity ID, the challenger C returns the secret key SID_0. Here, the query is forbidden if the identity ID has already appeared in the *public key replace query*.
  - *Sign query* (ID, m). For the j-th *Sign* round, upon receiving this query along with a user's identity ID and a message m, the challenger C uses the user's current private key (DID_{j-1}=($DID_{j-1,1}$, $DID_{j-1,2}$), SID_j=($SID_{j-1,1}$, $SID_{j-1,2}$)) to produce a signature σ on the message m by running two sub-algorithms *Sign*-1 and *Sign*-2 while updating the current private key with (DID_j=($DID_{j,1}$, $DID_{j,2}$), SID_j=($SID_{j,1}$, $SID_{j,2}$)). The challenger C then returns σ to A.
  - *Sign leak query* ($f_{S,j}$, $h_{S,j}$, j): For the j-th *Sign query* of the user with identity ID, the adversary A can issue the *Sign leak query* only once by providing two leakage functions $f_{S,j}$ and $h_{S,j}$. After receiving this query, the challenger C computes and sends the leakage information ($\Lambda f_{S,j}$, $\Lambda h_{S,j}$) to A, where $f_{S,j}$ and $h_{S,j}$ can be efficiently computed with bounded length output in $\{0, 1\}^{\lambda}$. Meanwhile, an adversary of Type II (honest-but-curious KGC) knows the initial key of any entity so that ($\Lambda f_{S,j}$, $\Lambda h_{S,j}$) includes only the leakage information of a user's secret key ($SID_{j-1,1}$, $SID_{j-1,2}$) of the private key. An adversary of Type I (outsider) can obtain the leakage information of a user's initial key ($DID_{j-1,1}$, $DID_{j-1,2}$) of the private key since an outsider owns the secret key of any entity.
- *Forgery*. The adversary A generates a tuple (m^*, ID^*, σ^*, PID^*=(QID^*, RID^*)). We say that A wins the game if the following conditions are satisfied.
  1. The response of the *Verify* algorithm on (m^*, ID^*, σ^*, PID^*) is "accept".
  2. (m^*, ID^*) has never been issued during the *Sign query*.
  3. If A is of Type I adversary (outsider), ID^* has never been issued during the *Initial key extract* query. If A is of Type II adversary (honest-but-curious KGC), it is disallowed to issue the queries on the *public key replace query* and *secret key extract query* on ID^*.

4. The Proposed LR-CLS Scheme

Based on the leakage-resilient signature scheme in [18] and the leakage-resilient ID-based signature scheme in [42], we present the first LR-CLS scheme, as defined in Section 3.1, which consists of seven algorithms. Fig. 1 depicts the key generation processes of the KGC and users. The functionalities of the *Sign* and *Verify* algorithms are depicted in Figure 2. The details of seven algorithms are given as follows.

- ***Setup***: The KGC chooses two multiplicative cyclic groups G and G_T of sufficiently large prime order p while picking an arbitrary generator g of the group G. Let e: $G \times G \to G_T$ be an admissible bilinear pairing. The KGC runs the following steps:
  1. Pick a random value $x \in Z_p^*$, and compute $X=g^x$ and $X_T= e(g^x, g)$.
  2. Pick a random value $\alpha \in Z_p^*$ and set the first system secret key $(S_{0,1}, S_{0,2}) = (g^\alpha, X \cdot g^{-\alpha})$.
  3. Pick four values ui_0, ui_1, mi_0, $mi_1 \in Z_p^*$ at random, and compute $U_0=g^{ui_0}$, $U_1=g^{ui_1}$, $M_0=g^{mi_0}$ and $M_1= g^{mi_1}$.
  4. Publish the public parameters $PP = (G, G_T, e, p, g, X_T, U_0, U_1, M_0, M_1)$.
- ***Initial key extract***: For the i-th round along with a user's identity ID, the KGC uses the current system secret key $(S_{i-1,1}, S_{i-1,2})$ to generate the first initial key (DID_0, QID) of the user while updating the current system secret key with $(S_{i,1}, S_{i,2})$ by running two sub-algorithms *Extract*-1 and *Extract*-2 as follows:
  - ***Extract*-1**: Given the user's identity ID, the KGC uses $S_{i-1,1}$ to generate the temporary information and QID as follows.
    1. Randomly select two values $\gamma, a \in Z_p^*$.
    2. Compute $QID=g^\gamma$ and $S_{i,1}= S_{i-1,1} \cdot g^a$.
    3. Compute the temporary information $TI_{IE}= S_{i,1} \cdot (U_0 \cdot U_1^{ID})^\gamma$.
  - ***Extract*-2**: Given TI_{IE}, the KGC uses $S_{i-1,2}$ to generate DID_0 as follows.
    1. Compute $S_{i,2} = S_{i-1,2} \cdot g^{-a}$.
    2. Set $DID_0 =S_{i,2} \cdot TI_{IE}$.

  Finally, the KGC updates the current system secret key by $(S_{i,1}, S_{i,2})$ and sends the first initial key $(DID_0, QID)=(X \cdot (U_0 \cdot U_1^{ID})^\gamma, g^\gamma)$ to the user via a secure channel. Meanwhile, the user can validate the correctness of the first initial key by checking $e(g, DID_0)=X_T \cdot e(QID, U_0 \cdot U_1^{ID})$.
- ***Set secret value***: A user with identity ID randomly selects a number $z \in Z_p^*$, and computes the secret key $SID_0=g^z$ and the partial public key $RID=e(g^z, g)$.
- ***Set private key***: Given the initial key $(DID_0, QID)=(X \cdot (U_0 \cdot U_1^{ID})^\gamma, g^\gamma)$ and the secret key $SID_0=g^z$, the user with identity ID chooses two random numbers β, $\omega \in Z_p^*$ and sets her/his current private key ($(DID_{0,1}, DID_{0,2})=(g^\beta, DID_0 \cdot g^{-\beta})$, $(SID_{0,1}, SID_{0,2})= (g^w, SID_0 \cdot g^{-w})$).
- ***Set public key***: Given the initial key $(DID_0, QID)=(X \cdot (U_0 \cdot U_1^{ID})^\gamma, g^\gamma)$ and the partial public key $RID=e(g^z, g)$, the user with identity ID sets her/his public key $PID=(QID=g^\gamma, RID=e(g^z, g))$.
- ***Sign***: For the j-th round of the signer with identity ID, given a message m, the signer employs the current private key $((DID_{j-1,1}, DID_{j-1,2}), (SID_{j-1,1}, SID_{j-1,2}))$ to generate a signature σ while updating the current private key to $(DID_j=(DID_{j,1}, DID_{j,2}), SID_j=(SID_{j,1}, SID_{j,2}))$. The signer runs two sub-algorithms as follows:
  - ***Sign*-1**: Given the message m, the signer uses $DID_{j-1,1}$ and $SID_{j-1,1}$ to generate the temporary information TI_S and compute new $DID_{j,1}$ and $SID_{j,1}$ by the following steps:
    1. Choose three random numbers $b, c, \eta \in Z_p^*$.
    2. Compute $DID_{j,1}=DID_{j-1,1} \cdot g^b$ and $SID_{j,1}= SID_{j-1,1} \cdot g^c$.
    3. Compute the temporary information $TI_S= SID_{j,1} \cdot DID_{j,1} \cdot (M_0 \cdot M_1^m)^\eta$.
    4. Compute $\sigma_2=g^\eta$.
  - ***Sign*-2**: Given TI_S, the signer uses $DID_{j-1,2}$ and $SID_{j-1,2}$ to generate a signature σ and compute new $DID_{j,2}$, $SID_{j,2}$ by the following steps:
    1. Compute $DID_{j,2}=DID_{j-1,2} \cdot g^{-b}$ and $SID_{j,2}= SID_{j-1,2} \cdot g^{-c}$.
    2. Compute $\sigma_1=SID_{j,2} \cdot DID_{j,2} \cdot TI_S$.

  Finally, the signer outputs a signature $\sigma = (\sigma_1, \sigma_2)$.
- ***Verify***: Given a signature $\sigma=(\sigma_1, \sigma_2)$ on the message m for the signer with identity ID and public key $PID=(QID=g^\gamma, RID=e(g^z, g))$, a verifier accepts the signature if $e(g, \sigma_1)= RID \cdot X_T \cdot e(QID, U_0 \cdot U_1^{ID}) \cdot e(\sigma_2, M_0 \cdot M_1^m)$; or rejects it otherwise. In the following, we show the correctness of the verifying equality as follows.

$$
\begin{aligned}
& e(g, \sigma_1) \\
&= e(g, SID_{j,2} \cdot DID_{j,2} \cdot SID_{j,1} \cdot DID_{j,1} \cdot (M_0 \cdot M_1^m)^\eta) \\
&= e(g, SID_{j,2} \cdot SID_{j,1} \cdot DID_{j,2} \cdot DID_{j,1} \cdot (M_0 \cdot M_1^m)^\eta) \\
&= e(g, g^z \cdot X \cdot (U_0 \cdot U_1^{ID})^\gamma \cdot (M_0 \cdot M_1^m)^\eta) \\
&= e(g, g^z \cdot g^x \cdot (U_0 \cdot U_1^{ID})^\gamma \cdot (M_0 \cdot M_1^m)^\eta) \\
&= e(g, g^z) \cdot e(g, g^x) \cdot e(g, (U_0 \cdot U_1^{ID})^\gamma) \cdot e(g, (M_0 \cdot M_1^m)^\eta) \\
&= e(g^z, g) \cdot e(g^x, g) \cdot e(g^\gamma, (U_0 \cdot U_1^{ID})) \cdot e(g^\eta, (M_0 \cdot M_1^m)) \\
&= RID \cdot X_T \cdot e(QID, U_0 \cdot U_1^{ID}) \cdot e(\sigma_2, M_0 \cdot M_1^m).
\end{aligned}
$$

Figure 1

The key generation processes of the KGC and users

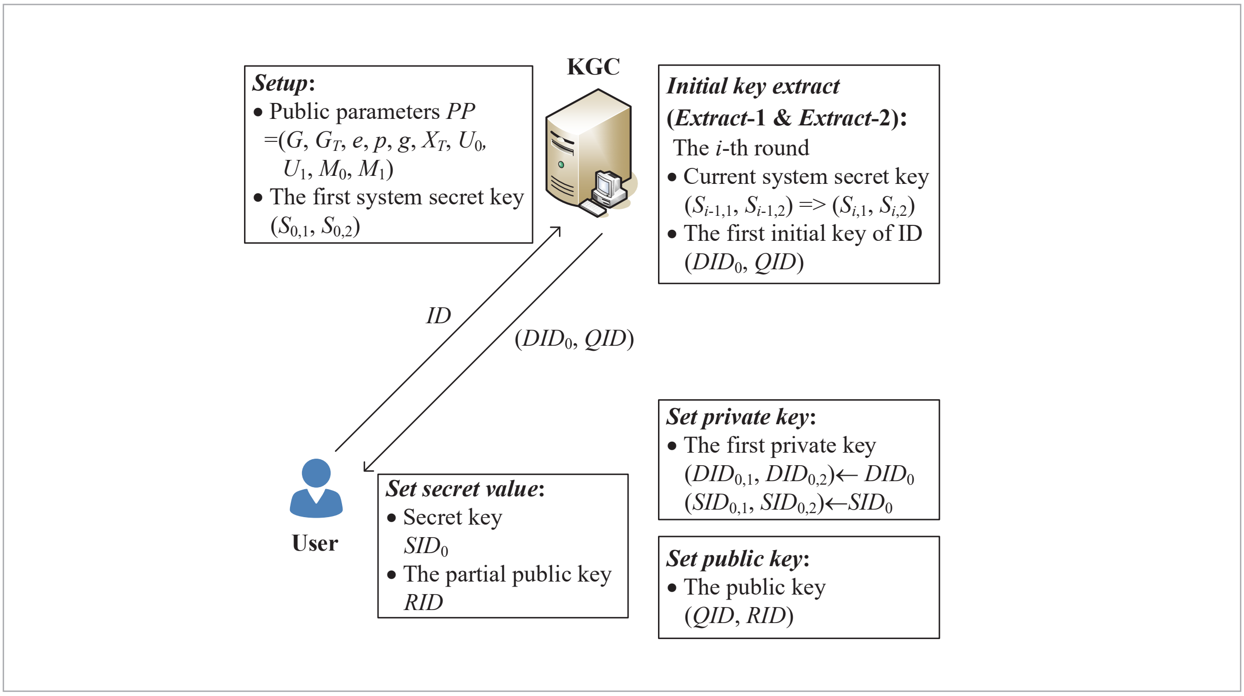
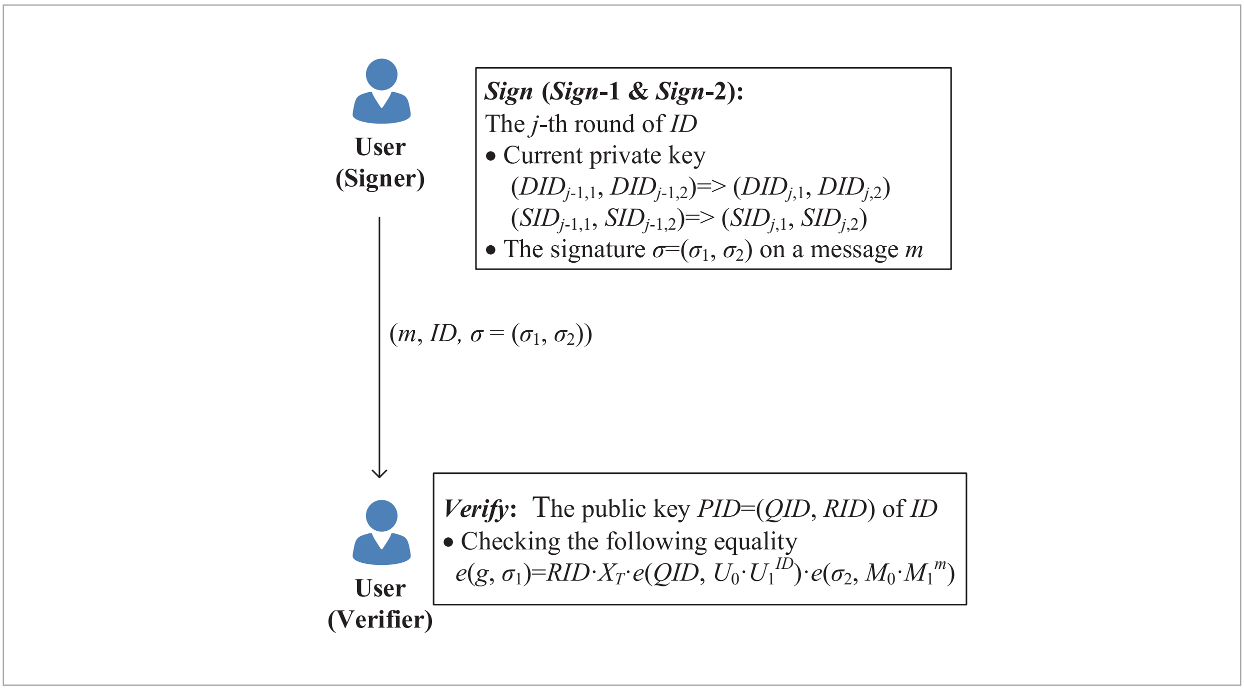


Figure 2

The *Sign* and *Verify* algorithms of the proposed scheme



5. Security Analysis

In the proposed LR-CLS scheme, a user's private key consists of two components, namely, an initial key and a secret key. As the aforementioned UF-LR-CLS-ACMA game in Definition 1, there are two kinds of adversaries, which include Type I (outsider) and Type II (honest-but-curious KGC). In the generic bilinear group model, we demonstrate that our LR-CLS scheme possesses existential unforgeability against adaptive chosen-message attacks for both Type I and Type II adversaries under the continual leakage model. We first prove that the non-leakage version of our LR-CLS scheme without leakage queries, denoted by Π_{NL}, is UF-CLS-ACMA secure in the generic bilinear group model. Then, based on the security of the non-leakage version, we demonstrate that our proposed LR-CLS scheme under the continual leakage model is UF-LR-CLS-ACMA secure in the generic bilinear group model.

The non-leakage version Π_{NL} of our LR-CLS scheme consists of seven algorithms $Setup_{NL}$, $Initial\ key\ extract_{NL}$, $Set\ secret\ value_{NL}$, $Set\ private\ key_{NL}$, $Set\ public\ key_{NL}$, $Sign_{NL}$ and $Verify_{NL}$:

- ***Setup***$_{NL}$: In this algorithm, the system key is generated by $X=g^x$ where x is picked from Z_p^* randomly. The generation of the public parameters $PP = (G, G_T, e, p, g, X_T, U_0, U_1, M_0, M_1)$ is identical to that of the proposed LR-CLS scheme. At the end of this algorithm, the KGC publishes the public parameters PP.
- ***Initial key extract***$_{NL}$: Upon receiving a user's identity ID, the KGC uses the system key X to generate the user's initial key (DID).
- , QID) = $(X\cdot(U_0\cdot U_1^{ID})^\gamma,\ g^\gamma)$ where γ is a random number picked from Z_p^*. The KGC then sends the user's private key pair (DID, QID) to the user via a secure channel.
- ***Set secret value***$_{NL}$: A user with identity ID randomly selects a random number $z\in Z_p^*$ and computes the secret key $SID=g^z$ and the partial public key $RID=e(g^z, g)$.
- ***Set private key***$_{NL}$: Given the initial key (DID, QID)=$(X\cdot(U_0\cdot U_1^{ID})^\gamma,\ g^\gamma)$ and the secret key $SID=g^z$, the user with identity ID sets her/his private key $(DID, SID)=(X\cdot(U_0\cdot U_1^{ID})^\gamma, g^z)$.
- ***Set public key***$_{NL}$: This phase is identical to that of the proposed LR-CLS scheme.
- ***Sign***$_{NL}$: For the signer with identity ID, given a message m, the signer employs the user's private key DID and the user's secret key SID to generate a signature $\sigma = (\sigma_1, \sigma_2) = (SID\cdot DID\cdot(M_0\cdot M_1^{m})^\eta,\ g^\eta)$, where $\eta\in Z_p^*$. The signer then outputs σ.
- ***Verify***$_{NL}$: Upon receiving the signature (σ_1, σ_2), a verifier accepts the signature if $e(g, \sigma_1)=RID\cdot X_T\cdot e(QID, U_0\cdot U_1^{ID})\cdot e(\sigma_2, M_0\cdot M_1^{m})$, where QID and RID are the public keys of the user with identity ID; or rejects it otherwise.

In the generic bilinear group (GBG) model, we first prove that our non-leakage version Π_{NL} is UF-CLS-ACMA secure against Type I and Type II adversaries in Theorems 1 and 2, respectively. Based on the security of the non-leakage version, by adding extra leak queries, we then prove that our LR-CLS scheme under the continual leakage model is UF-LR-CLS-ACMA secure against Type I and Type II adversaries in Theorems 3 and 4, respectively. Figure. 3 demonstrates the relationships of the associated four security theorems. In addition, Figure 4 depicts the conceptual principle of the security games $g_{NL\text{-I}}$ and $g_{NL\text{-II}}$ employed in Theorems 1 and 2.

Theorem 1. In the generic bilinear group model, the non-leakage version Π_{NL} of the proposed LR-CLS scheme is provably secure against the Type I adversary (outsider).

Proof: Let $A_{NL\text{-I}}$ be a Type I adversary who can break the non-leakage CLS scheme Π_{NL} while $A_{NL\text{-I}}$ is allowed to issue all the queries at most q times. The advantage of $A_{NL\text{-I}}$ is defined as the probability that $A_{NL\text{-I}}$ wins the following game $g_{NL\text{-I}}$ played with a challenger C.

Game $g_{NL\text{-I}}$: In the game $g_{NL\text{-I}}$, there are three phases, *Setup*, *Queries* and *Forgery* phases. At the end of this game, $A_{NL\text{-I}}$ outputs a forgery signature. In *Queries* phase, $A_{NL\text{-I}}$ may issue eight kinds of queries in any order at most q times. Three phases are described as below:

- ***Setup*** phase: The challenger C builds and maintains two lists L_G and L_T which are used to record group elements in G and G_T, respectively, described below.
  - The list L_G consists of pairs of the form $(F_{G,w,k,l}, \xi_{G,w,k,l})$, where $F_{G,w,k,l}$ is a multivariate polynomial with coefficients in Z_p and variates in G and $\xi_{G,w,k,l}$ is the bit string denoting $F_{G,w,k,l}$. The first index of $F_{G,w,k,l}$ is "G", which denotes this elements represents an element in G, on

Figure 3

The relationships of four security theorems

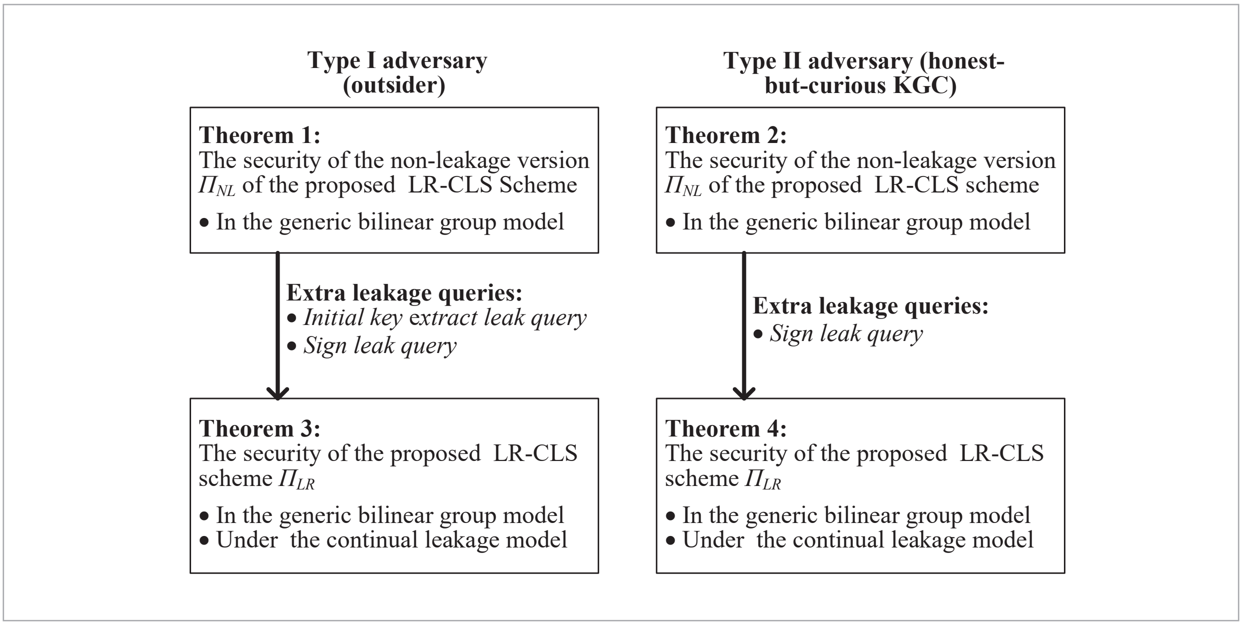
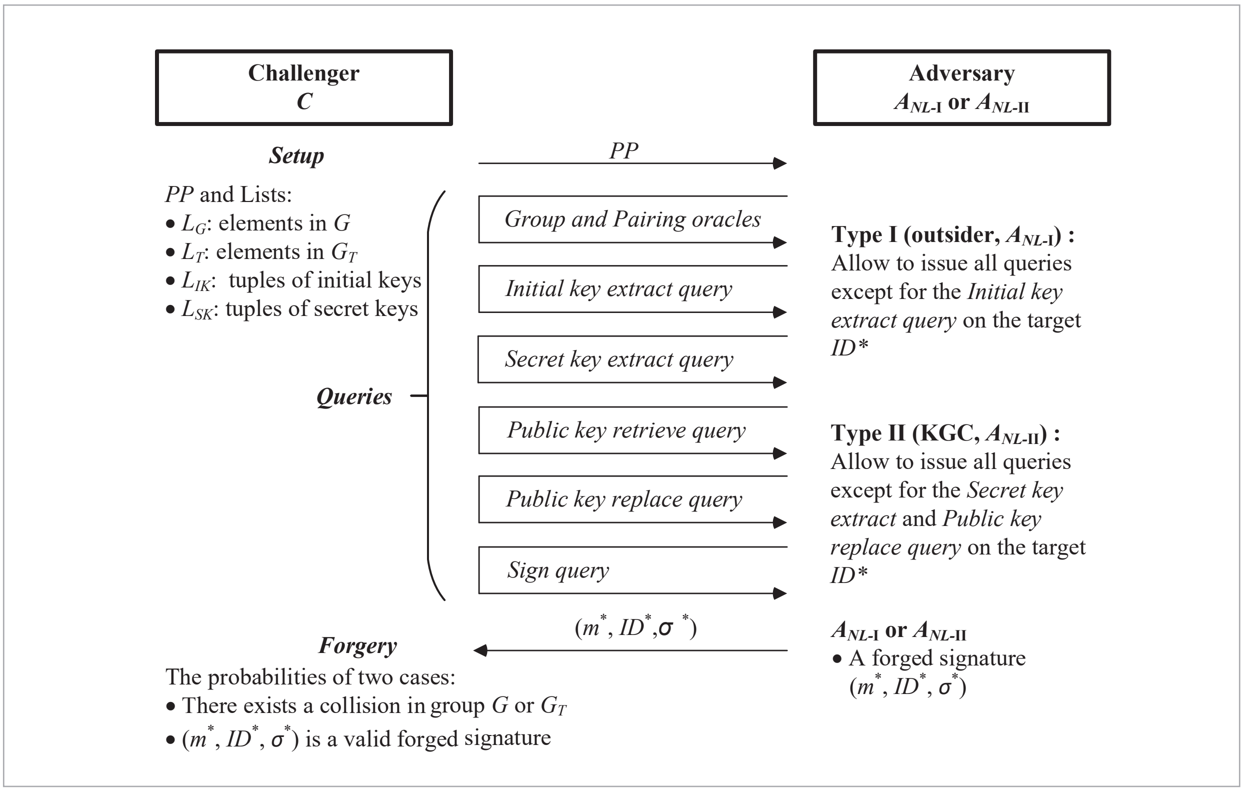


Figure 4

The conceptual principle of the security games $g_{NL\text{-I}}$ and $g_{NL\text{-II}}$ in Theorems 1 and 2



the other hand if the first index is "T", which denotes this elements in G_T. The second index "w" is indicating the type of query. The third and fourth index "k" and "l" represent the l-th element appeared in the k-th type "w" query in this game. Meanwhile, six initial tuples (g, $\xi_{G,I,1,1}$), (X, $\xi_{G,I,1,2}$), (U_0, $\xi_{G,I,1,3}$), (U_1, $\xi_{G,I,1,4}$), (M_0, $\xi_{G,I,1,5}$) and (M_1, $\xi_{G,I,1,6}$) are added in L_G, where $\xi_{G,I,1,i}$ (for i=1, 2, ..., 6) are six different bit strings generated randomly representing the elements in G.

- The List L_T consists of pairs of the form ($F_{T,w,k,l}$, $\xi_{T,w,k,l}$). The meanings of the indexes of $F_{T,w,k,l}$ are the same with the descriptions of $F_{G,w,k,l}$ before. The only difference is that $F_{T,w,k,l}$ is a multivariate polynomial with coefficients in Z_p and variates in G or G_T. The initial tuple (X_T, $\xi_{T,I,1,1}$) is added in L_T, where $\xi_{T,w,k,l}$ is a bit string generated randomly representing X_T. The challenger sends the bit strings of the public parameters to $A_{NL\text{-}I}$ at the end of this phase.

Moreover, the challenger C also maintains two lists L_{IK} and L_{SK} to record the tuples of the users' initial keys and secret keys, respectively. More precisely, L_{IK} and L_{SK} consists, respectively, of tuples of the forms (*ID*, *DID*, *QID*) and (*ID*, *SID*, *RID*), where *ID* is in Z_p^*. Here, *DID*, *QID*, *SID* and *RID* are multivariate polynomials.

- ***Queries*** phase: In this phase, the adversary $A_{NL\text{-}I}$ may issue eight kinds of queries to the challenger C at most q times in any order.
  - ***Group oracle O$_G$*** ($\xi_{G,O,i,1}$, $\xi_{G,O,i,2}$, *operation*): For the i-th group oracle O_G, upon receiving this query along with two bit strings $\xi_{G,O,i,1}$, $\xi_{G,O,i,2}$ and an *operation* (multiplication or division), C runs the following three steps:
    - Translates the bit strings $\xi_{G,O,i,1}$ and $\xi_{G,O,i,2}$ back into two polynomials $F_{G,O,i,1}$ and $F_{G,O,i,2}$, respectively, in the following way: C tries to find a pair ($F_{G,\omega,k,l}$, $\xi_{G,\omega,k,l}$) in L_G such that $\xi_{G,\omega,k,l}$= $\xi_{G,O,i,1}$. If so, C sets $F_{G,O,i,1}$= $F_{G,\omega,k,l}$. Otherwise, C randomly chooses a new variate $S_{G,O,i,1}$ in G, sets $F_{G,O,i,1}$=$S_{G,O,i,1}$, and records ($F_{G,O,i,1}$, $\xi_{G,O,i,1}$) in L_G. Similarly, C translates the bit string $\xi_{G,O,i,2}$ into $F_{G,O,i,2}$.
    - Set the polynomial $F_{G,O,i,3}$=$F_{G,O,i,1}$+$F_{G,O,i,2}$ if the *operation* is a multiplication, and $F_{G,O,i,3}$=$F_{G,O,i,1}$− $F_{G,O,i,2}$ if the *operation* is a division.
    - Try to find a pair ($F_{G,\omega,k,l}$, $\xi_{G,\omega,k,l}$) in L_G such that $F_{G,\omega,k,l}$ = $F_{G,O,i,3}$. If so, C returns $\xi_{G,\omega,k,l}$ to $A_{NL\text{-}I}$. Otherwise, C randomly selects a bit string $\xi_{G,O,i,3}$ which is distinct from all the $\xi_{G,\omega,k,l}$ appeared in L_G. Finally, C records ($F_{G,O,i,3}$, $\xi_{G,O,i,3}$) in L_G and returns $\xi_{G,\omega,k,l}$= $\xi_{G,O,i,3}$ to $A_{NL\text{-}I}$.

  Note that the polynomials $F_{G,O,i,1}$, $F_{G,O,i,2}$ and $F_{G,O,i,3}$ mentioned above are recorded in the list L_G.
  - ***Group oracle O$_T$*** ($\xi_{T,O,i,1}$, $\xi_{T,O,i,2}$, *operation*): This oracle is similar to the *Group oracle O_G* above. For the i-th group oracle O_T, upon receiving this query along with two bit strings $\xi_{T,O,i,1}$, $\xi_{T,O,i,2}$ and an *operation* (multiplication or division), C returns $\xi_{T,\omega,k,l}$= $\xi_{T,O,i,3}$ to $A_{NL\text{-}I}$ and the polynomials $F_{T,O,i,1}$, $F_{T,O,i,2}$ and $F_{T,O,i,3}$ are recorded in L_T after this query.
  - ***Pairing oracle O$_P$***($\xi_{G,P,i,1}$, $\xi_{G,P,i,2}$): For the i-th pairing oracle O_P, upon receiving this query along with two bit strings $\xi_{G,P,i,1}$, $\xi_{G,P,i,2}$, C runs the following steps:
    - Similarly as in the Step 1 of the *Group oracle O_G*, C translates the bit strings $\xi_{G,P,i,1}$ and $\xi_{G,P,i,2}$ back into two polynomials $F_{G,P,i,1}$ and $F_{G,P,i,2}$, respectively. Additionally, C computes the polynomial $F_{T,P,i,1}$ = $F_{G,P,i,1}$·$F_{G,P,i,2}$.
    - C tries to find a pair ($F_{T,\omega,k,l}$, $\xi_{T,\omega,k,l}$) in L_T such that $F_{T,\omega,k,l}$=$F_{T,P,i,1}$. If so, C returns $\xi_{T,\omega,k,l}$ to $A_{NL\text{-}I}$. Otherwise, C randomly selects a bit string $\xi_{T,P,i,1}$ which is distinct from all the $\xi_{T,\omega,k,l}$ appeared in L_T. Finally, C records ($F_{T,P,i,1}$, $\xi_{T,P,i,1}$) in L_T and returns $\xi_{T,\omega,k,l}$= $\xi_{T,P,i,1}$ to $A_{NL\text{-}I}$.

  It is worth mentioning, after this query, that the polynomials $F_{G,O,i,1}$ and $F_{G,O,i,2}$ have been recorded in the list L_G while $F_{T,O,i,1}$ has also been recorded in the list L_T.
  - ***Initial key extract query Q$_{IE}$***($ID_{IE,i}$): For the i-th *initial key extract query*, upon receiving this query along with a user's identity $ID_{IE,i}$∈Z_p^*, C first checks whether $ID_{IE,i}$ has been recorded in the list L_{IK}. If so, C returns the bit strings ($\xi_{G,IE,i,1}$, $\xi_{G,IE,i,2}$) representing the initial key (*DID*,*QID*) of the user with identity $ID_{IE,i}$ to $A_{NL\text{-}I}$. Otherwise, C runs the following steps:
    - C defines one variate $T_{G,IE,i,2}$ in G for representing *QID* of the identity $ID_{IE,i}$ and sets $F_{G,IE,i,2}$=$T_{G,IE,i,2}$. Additionally, C selects

a random bit string $\xi_{G,IE,i,2}$ which is distinct from all the $\xi_{G,\omega,k,l}$ appeared in L_G, and records $(F_{G,IE,i,2}, \xi_{G,IE,i,2})$ in L_G. Furthermore, C computes the polynomial $F_{G,IE,i,1}=X+(U_0+ID_{IE,i}\cdot U_1)\cdot T_{G,IE,i,2}$ for representing *DID* of the identity $ID_{IE,i}$.

- C selects a random bit string $\xi_{G,IE,i,1}$ which is distinct from all the $\xi_{G,\omega,k,l}$ appeared in L_G. Finally, C records $(F_{G,IE,i,1}, \xi_{G,IE,i,1})$ in L_G and returns $(\xi_{G,IE,i,1}, \xi_{G,IE,i,2})$ to $A_{NL\text{-}I}$.

Finally, the challenger C also maintains an element $(ID_{IE,i}, F_{G,IE,i,1}, F_{G,IE,i,2})$ in the list L_{IK}.

- ***Secret key extract query*** $\mathbf{Q_{SE}}(ID_{SE,i})$: When $A_{NL\text{-}I}$ issues the i-th *Secret key extract query* along with an identity $ID_{SE,i}$, the challenger C returns the bit strings $(\xi_{T,SE,i,1}, \xi_{T,SE,i,2})$ for representing the secret key pair (*SID*, *RID*) by running the steps as below.
  - The challenger C checks whether the secret key pair of identity $ID_{SE,i}$ has been recorded in L_{SK}. If so, C returns the bit strings $(\xi_{G,SE,i,1}, \xi_{T,SE,i,2})$ representing the secret key (*SID*,*RID*) of the user with identity $ID_{SE,i}$ to $A_{NL\text{-}I}$.
  - If the identity $ID_{SE,i}$ is not recorded in L_{SK}, C defines one variate $T_{G,SE,i,1}$ in G and sets the polynomial $F_{G,SE,i,1}=T_{G,SE,i,1}$ for representing *SID* of $ID_{SE,i}$. Moreover, C randomly chooses a bit string $\xi_{G,SE,i,1}$ which is distinct from all the $\xi_{G,\omega,k,l}$ appeared in L_G. Then C records $(F_{G,SE,i,1}, \xi_{G,SE,i,1})$ in L_G.
  - C sets the polynomial $F_{T,SE,i,2}=T_{G,SE,i,1}\cdot g$ for representing *RID* for $ID_{SE,i}$. Moreover, C selects a random bit string $\xi_{T,SE,i,2}$ which is distinct from all the $\xi_{T,\omega,k,l}$ appeared in L_T. Then C records $(F_{T,SE,i,2}, \xi_{T,SE,i,2})$ in L_T and returns $(\xi_{G,SE,i,1}, \xi_{T,SE,i,2})$ to $A_{NL\text{-}I}$.

Finally, the challenger C also maintains the element $(ID_{SE,i}, F_{G,SE,i,1}, F_{T,SE,i,2})$ in L_{SK}.

- ***Public key retrieve query*** $\mathbf{Q_{PK}}(ID_{PK,i})$: When $A_{NL\text{-}I}$ issues the i-th *Public key retrieve query* along with an identity $ID_{PK,i}\in Z_p^*$, the challenger C performs the following steps:
  - C checks whether $ID_{PK,i}$ has been recorded in the list L_{IK}. If so, C obtains the polynomial of *QID* for $ID_{PK,i}$ in L_{IK}. Otherwise, C performs the *Initial Key extract query*$(ID_{PK,i})$ to set the polynomial of *QID* for $ID_{PK,i}$.
  - C checks whether $ID_{PK,i}$ has been record in the list L_{SK}. If so, C obtains the polynomial of *RID* for $ID_{PK,i}$ in L_{SK}. Otherwise, C performs the *Secret key extract query* $(ID_{PK,i})$ to set the polynomial of *RID* for $ID_{PK,i}$.
  - Finally, C answers the query by two bit strings of *QID* and *RID* by searching the lists L_G and L_T, respectively.
- ***Public key replace query*** $\mathbf{Q_{PR}}(ID_{PR,i}, \xi_{T,PR,i,2})$: By this query, a type I adversary $A_{NL\text{-}I}$ can replace the original partial public key *RID* of a user with identity $ID_{PR,I}$ by the bit string $\xi_{T,PR,i,2}$. In other words, $A_{NL\text{-}I}$ can choose a valid *SID* and set the corresponding *RID* by herself/himself. C must record this replacement. More precisely, C first translates $\xi_{T,PR,i,2}$ to the polynomial $F_{T,PR,i,2}$ by searching the list L_T. Since $A_{NL\text{-}I}$ can generate valid user's secret key by using the group oracles, thus C can obtain the polynomial $F_{G,PR,i,1}$ by searching $F_{T,PR,i,2}=F_{G,PR,i,1}\cdot g$ in the list L_G. The challenger C then update the user's secret key $(ID_{PR,i}, SID_{PR,i}, RID_{PR,i})=(ID_{PR,i}, F_{G,PR,i,1}, F_{T,PR,i,2})$ in the list L_{SK}.
- ***Sign query*** $\mathbf{Qs}(ID_{S,i}, m_i)$: For the i-th *Sign query*, upon receiving this query along with an identity $ID_{S,i}\in Z_p^*$ and a message $m_i\in Z_p^*$, the challenger C returns $(\xi_{G,S,i,1}, \xi_{G,S,i,2})$ to $A_{NL\text{-}I}$ at the end of this query. When C receives the query, C respectively obtains the user's private key *DID* and secret key *SID* from the lists L_{IK} and L_{SK} by the following steps:
  - C checks whether the user's private key of $ID_{S,i}$ has been recorded in the list L_{IK}. If so, C obtains *DID* of $ID_{S,i}$ in L_{IK}. Otherwise, C performs the query $Q_{IE}(ID_{S,i})$ to obtains *DID*.
  - C checks whether the user's secret key *SID* of $ID_{S,i}$ has been recorded in the list L_{SK}. If so, C obtains *SID* of $ID_{S,i}$ in L_{SK}. Otherwise, C performs the query $Q_{SE}(ID_{S,i})$ to obtain *SID*.
  - Hence, C can obtain the polynomials $F_{G,IE,k,1}$ and $F_{G,SE,l,1}$ representing *DID* and *SID*, respectively.

Then C can return $(\sigma_1, \sigma_2)=(\xi_{G,S,i,1}, \xi_{G,S,i,2})$ to $A_{NL\text{-}I}$ by running the following steps:

- In order to generate σ_2, C first defines a new variate $T_{G,S,i,2}$ in G and sets $F_{G,S,i,2}=T_{G,S,i,2}$.

Moreover, C selects a random bit string $\xi_{G,S,i,2}$ which is distinct from all the $\xi_{G,\omega,k,l}$ in L_G. Then C adds ($F_{G,S,i,2}$, $\xi_{G,S,i,2}$) in L_G while computing the polynomial $F_{G,S,i,1} = F_{G,IE,k,2} + F_{G,SE,l,1}+(M_0+m_jM_1)\cdot T_{G,S,i,2}$.

- Finally, C tries to find a pair ($F_{G,\omega,j,l}$, $\xi_{G,\omega,j,l}$) in L_G such that $F_{G,\omega,j,l} = F_{G,S,i,1}$. If so, C returns ($\xi_{G,\omega,j,l}$, $\xi_{G,S,i,2}$) to $A_{NL\text{-}I}$. Otherwise, C selects a random bit string $\xi_{G,S,i,1}$ which is distinct from all the $\xi_{G,\omega,k,l}$ in L_G. Finally, C adds ($F_{G,S,i,1}$, $\xi_{G,S,i,1}$) in L_G and returns (σ_1, σ_2)= ($\xi_{G,S,i,1}$, $\xi_{G,S,i,2}$) to $A_{NL\text{-}I}$.

- ***Forgery*** phase: In this phase, the adversary $A_{NL\text{-}I}$ outputs a forgery signature (m^*, ID^*, σ^* =($\xi_{G,f,i,1}{}^*$, $\xi_{G,f,i,2}{}^*$)), where there are two restrictions: (1) ID^* has never been issued during the *Initial key extract query* Q_{IE}; (2) (m^*, ID^*) has never been issued during the *Sign query* Q_S.

In the following, before discussing the probability that $A_{NL\text{-}I}$ wins the game $g_{NL\text{-}I}$, we define several restrictions and notations as below:

1 In the game $g_{NL\text{-}I}$, $A_{NL\text{-}I}$ can issue eight kinds of queries O_G, O_T, O_P, Q_{IE}, Q_{SE}, Q_{PK}, Q_{PR} and Q_S. Let q_O denote the total number of three oracles O_G, O_T and O_P issued by $A_{NL\text{-}I}$. In addition, let q_{IE}, q_{SE}, q_{PK}, q_{PR} and q_S respectively denote the numbers of the queries Q_{IE}, Q_{SE}, Q_{PK}, Q_{PR} and Q_S issued by $A_{NL\text{-}I}$. Since $A_{NL\text{-}I}$ can issue queries at most q times, we have $q \geqq q_O+q_{IE}+q_{SE}+q_{PK}+q_{PR}+q_S$. Moreover, we define several sets as follows.
   - $\{S\}$: The set of both variates $S_{G,O,i,j}$ defined in O_G and $S_{G,P,i,j}$ defined in O_P.
   - $\{V\}$: The set of the variates $V_{T,O,i,j}$ defined in O_T.
   - $\{T\}$: The set of the variates $T_{G,IE,i,2}$ defined in Q_{IE}, $T_{G,S,i,3}$ defined in Qs and $T_{G,SE,i,1}$ defined in Q_{SE}.
   - $\{F_G\}$: The set of the polynomials $F_{G,O,i,k}$, $F_{G,IE,i,k}$ and $F_{G,S,i,k}$ in the *Queries* phase.
   - $\{F_T\}$: The set of the polynomials $F_{T,O,i,k}$ and $F_{T,P,i,k}$ in the *Queries* phase.

2 Let $|L_G|$ and $|L_T|$ denote the total numbers of tuples in the lists L_G and L_T, respectively. Meanwhile, we have $|L_G|+|L_T| \leqq 3q_O+2q_{IE}+2q_{SE}+4q_{PK}+5q_S+9 \leqq 5q$ due to the following reasons:
   - In each query of O_G, O_T and O_P, there are at most three elements involved in the query. So, the elements generated O_G, O_T and O_P is bounded by $3q_O$, where q_O denotes the total time of three oracles issued by $A_{NL\text{-}I}$. In addition, no new elements of both L_G and L_T are generated in the query Q_{PR}.
   - For the query Q_{IE}, there are at most two new elements added in the list L_G. So, the increasing numbers of $|L_G|+|L_T|$ is bounded by $2q_{IE}$.
   - For the query Q_{SE}, there are at most two new elements added in the list L_G or L_T. So, the increasing numbers of $|L_G|+|L_T|$ is bounded by $2q_{SE}$.
   - For the query Q_{PK}, there are at most four new elements added in the list L_G or L_T. So, the increasing numbers of $|L_G|+|L_T|$ is bounded by $4q_{PK}$.
   - For the query Q_S, there are at most five new elements added in the list L_G or L_T. So, the increasing numbers of $|L_G|+|L_T|$ is bounded by $6q_S$.

   Hence $|L_G|+|L_T| \leqq 7+3q_O+2q_{IE}+2q_{SE}+4q_{PK}+6q_S+2$. Let $9 \leqq 3q_O+4q_{IE}+4q_{SE}+2q_{PK}+6q_{PR}$, we have $|L_G|+|L_T| \leqq 3q_O+2q_{IE}+2q_{SE}+4q_{PK}+6q_S+9 \leqq 6q$.

3 The degrees of all multivariate polynomials in the set $\{F_G\}$ are at most 2 by the following reasons:
   - All the elements in $\{S\}$ and $\{T\}$ are polynomials with only one term, hence all the polynomials in $\{S\}$ and $\{T\}$ are of degree 1.
   - For Q_{IE}, each polynomial $F_{G,IE,i,k}$ has degree at most 2.
   - For Q_{SE}, each polynomial $F_{G,SE,i,1}$ has degree 1.
   - For Q_S, each polynomial $F_{G,S,i,k}$ has degree at most 2.
   - For O_G, the degree of $F_{G,O,i,1}+F_{G,O,i,2}$ is equal to the maximal degree of $F_{G,O,i,1}$ and $F_{G,O,i,2}$.

4 The degrees of all multivariate polynomials in the set $\{F_T\}$ are at most 4 by the following reasons:
   - All the elements in $\{V\}$ are polynomials with only one term, hence all the polynomials in $\{V\}$ are of degree 1.
   - For O_P, each polynomial $F_{T,P,i,k}$ has degree at most 4 since the degree of F_G is at most 2.
   - For Q_{SE}, each polynomial $F_{T,SE,i,2}$ has degree 2.
   - For O_T, the degrees of $F_{T,O,i,1}+F_{T,O,i,2}$ are equal to the maximal degree of $F_{T,O,i,1}$ and $F_{T,O,i,2}$.

Assume that $A_{NL\text{-}I}$ generates a signature (m^*, ID^*,

$\sigma^*=(\xi_{G,f,1,1}*, \xi_{G,f,1,2}*)$) while $ID*$ is not issued in the query Q_{IE} and (m^*, ID^*) is not issued in the query Q_S. C first uses $ID*$ to obtain the polynomials of QID and RID, denoted by QID_{ID^*} and RID_{ID^*}, respectively. Let $F_{G,f,1,3}$ and $F_{G,f,1,4}$ be the polynomials corresponding to QID_{ID^*} and RID_{ID^*}, respectively. Also, let $F_{G,f,1,1}$ and $F_{G,f,1,2}$ be the polynomials corresponding to $\xi_{G,f,1,1}*$ and $\xi_{G,f,1,2}*$, respectively. Then C computes the polynomial $F_{G,f,1,5}=X+F_{G,f,1,4}+(U_0+ID^*\cdot U_1)\cdot F_{G,f,1,3}+(M_0+m^*M_1)\cdot F_{G,f,1,2}-F_{G,f,1,1}$. Note that the polynomial $F_{G,f,1,5}$ has degree at most 3. Moreover, C selects the random values x, u_0, u_1, m_0, m_1, $\{s_1, s_2, ..., s_{|\{S\}|}\}$ and $\{t_1, t_2, ..., t_{|\{T\}|}\}$ in Z_p^* and generates the corresponding values X, U_0, U_1, M_0, M_1, $\{S\}$, $\{T\}$ in the group G. C also selects the random values $\{v_1, v_2, ..., v_{|\{V\}|}\}$ in Z_p^* and generates $\{V\}$ in the group G_T.

Here, let us discuss the situations that $A_{NL\text{-I}}$ wins the game $g_{NL\text{-I}}$. We say that $A_{NL\text{-I}}$ wins the game $\boldsymbol{g}_{NL\text{-I}}$ if one of the following two cases occurs:

- **Case 1**. There exists a collision in group G or G_T. We describe them as below:
  - There are two polynomials $F_{G,i}$ and $F_{G,j}$ in the list L_G such that $F_{G,i}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\})=F_{G,j}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\})$.
  - There are two polynomials $F_{T,i}$ and $F_{T,j}$ in the list L_T such that $F_{T,i}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\}, \{v\})=F_{T,j}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\}, \{v\})$.
- **Case 2**. In the forgery phase, the adversary $A_{NL\text{-I}}$ generates the forgery signature ($m*$, $ID*$, ($\xi_{G,f,i,1}*$, $\xi_{G,f,i,2}*$)) which satisfies the equality $F_{G,f,1,5}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\})= 0$, where $F_{G,f,1,5}$ is computed earlier.

In the real UF-CLS-ACMA game defined in Definition 1, the success probability in the game $g_{NL\text{-I}}$ is an upper bound of the advantage of $A_{NL\text{-I}}$. In the following, we discuss the probabilities of two cases in the game $g_{NL\text{-I}}$. The probabilities of two cases are computed as below:

- **Case 1.** If there exists a collision in group G or G_T, then one may solve the discrete logarithm problem in G or G_T [26]. Assume that $F_{G,i}$ and $F_{G,j}$ denote two distinct polynomials in L_G such that $F_{G,i}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\})=F_{G,j}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\})$. In such a case, the polynomial $F_{G,C}=F_{G,i}-F_{G,j}$ is a non-zero polynomial, whose degree is at most 2. By Lemma 2 in Section 2, the probability of $F_{G,C}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\})=0$ in Z_p is at most $2/p$. Since $|L_G|$ denotes the total number of tuples in the list L_G, there are $\binom{|L_G|}{2}$ possible pairs ($F_{G,i}$, $F_{G,j}$). The collision probability in L_G is at most $(2/p)\binom{|L_G|}{2}$. Similarly, since the maximal degree of polynomials in L_T is at most 4, the collision probability in L_T is at most $(4/p)\binom{|L_T|}{2}$.
- **Case 2**. In this case, the success probability of $A_{NL\text{-I}}$ is the probability that $A_{NL\text{-I}}$ can forge a valid signature ($m*$, $ID*$, $\sigma^*=(\xi_{G,f,i,1}*$, $\xi_{G,f,i,2}*$)) which satisfies the equality $F_{G,f,1,5}(x, m_0, m_1, u_0, u_1, \{s\}, \{t\})= 0$, where $F_{G,f,1,5}=X+F_{G,f,1,4}+(U_0+ID^*\cdot U_1)\cdot F_{G,f,1,3}+(M_0+m^*M_1)\cdot F_{G,f,1,2}-F_{G,f,1,1}$. Here, the polynomial $F_{G,f,1,5}$ has degree at most 3. In the meantime, $F_{G,f,1,5}$ is a non-zero polynomial that will be proved in Lemma 3 later. In such a case, by Lemma 2 in Section 2, the probability of Case 2 is at most $3/p$.

Since $|L_G|+|L_T|\leqq 6q$ as mentioned earlier, the advantage that $A_{NL\text{-I}}$ wins the game $g_{NL\text{-I}}$ in Case 1 or 2 is at most

$$(2/p)\binom{|L_G|}{2}+(4/p)\binom{|L_T|}{2}+(3/p)\leqq(2/p)\,(\,|L_G|+|L_T|)^2\leqq 72q^2/p,$$

which is negligible if $q = poly(\log p)$. □

Lemma 3. The polynomial $F_{G,f,1,5}= X + F_{G,f,1,4} + (U_0 + ID^*\cdot U_1)\cdot F_{G,f,1,3} + (M_0+m^*M_1)\cdot F_{G,f,1,2} - F_{G,f,1,1}$ is a non-zero polynomial.

Proof: By the group oracle O_G in the game $g_{NL\text{-I}}$, the increased elements (polynomials) in L_G are obtained by adding or subtracting two polynomials in L_G. In such a case, we may write $F_{G,f,1,l}$, for l=1, 2, 3, 4, as the following form,

$$\begin{aligned}F_{G,f,1,l} &= c_{l,1}+c_{l,2}U_1+c_{l,3}U_0+c_{l,4}M_0\\&+c_{l,5}M_1+\sum_{i=1}^{3q_g}\left(d_{l,6,i}\cdot S_i\right)+\sum_{i=1}^{q_S+q_{IE}}\left(d_{l,7,i}\cdot T_i\right)\\&+\sum_{i=1}^{q_{IE}}\left(d_{l,8,i}\cdot DID_{IE,i}\right)+\sum_{j=1}^{q_{SE}}\left(d_{l,9,j}\cdot SID_{SE,j}\right)\\&+\sum_{i=1}^{q_S}\left(d_{l,10,k}\left(DID_{IE,i}+SID_{SE,k}+\left(M_0+m_k\cdot M_1\right)\cdot T_{G,S,k,2}\right)\right),\end{aligned}$$

where $DID_{IE,i}=X+(U_0+ID_{IE,i}\cdot U_1)\cdot T_{G,IE,i,2}$ for $1\leqq i\leqq q_{IE}$, and $SID_{SE,j}=T_{G,SE,j,1}$ for $1\leqq j\leqq q_{SE}$. In addition, S_i and T_i respectively run through all the elements in the sets $\{S\}$ and $\{T\}$. It is worth mentioning, that each $c_{i,j}$ and $d_{i,j,k}$ in Z_p are randomly selected by the adversary $A_{NL\text{-I}}$. In the following, we discuss three cases to show that $F_{G,f,1,5}$ is a non-zero polynomial.

1 **Case 1**. If $\sum_{j=1}^{q_{IE}} d_{2,8,j} = \sum_{j=1}^{q_{IE}} d_{3,8,j} = (\sum_{j=1}^{q_{IE}} d_{4,8,j} -$

$\sum_{j=1}^{q_{IE}} d_{1,8,j}$)=0 and $d_{2,10,k}=d_{3,10,k}=$ $(d_{4,10,k}-d_{1,10,k}$)=0 for all $1 \leqq j \leqq q_{IE}$ and $1 \leqq k \leqq q_S$, then $F_{G,f,1,2}$, $F_{G,f,1,3}$ and $(F_{G,f,1,4}-F_{G,f,1,1})$ do not contain the indeterminate X. In such a case, the coefficient of the term X in $F_{G,f,1,5}$ is 1. Therefore, $F_{G,f,1,5}$ must be non-zero.

2 **Case 2**: At least one of $d_{2,10,k}$, $d_{3,10,k}$ and $(d_{4,10,k} - d_{1,10,k})$ is non-zero for some k.

- If $d_{2,10,k} \neq 0$ for some k, $F_{G,f,1,5}$ is non-zero since $d_{2,10,k}$ is the coefficient of the term $M_0^{\,2} \cdot T_{G,S,k,2}$ in $F_{G,f,1,5}$.
- If $d_{3,10,k} \neq 0$ for some k, $F_{G,f,1,5}$ is non-zero since $d_{3,10,k}$ is the coefficient of the term $U_0 \cdot M_0 \cdot T_{G,S,k,2}$ in $F_{G,f,1,5}$.
- If $(d_{4,10,k} - d_{1,10,k}) \neq 0$ for some k, we discuss the following two cases.
  - If $(\sum_{i=1}^{q_S+q_{IE}} d_{2,7,i})+(d_{4,10,k}-d_{1,10,k}) \neq 0$, then $F_{G,f,1,5}$ is non-zero since $(\sum_{i=1}^{q_S+q_{IE}} d_{2,7,i})+(d_{4,10,k}-d_{1,10,k}) = d_{4,9,k}+(\sum_{i=1}^{q_S+q_{IE}} d_{2,7,i}) - d_{1,9,k}$ is the coefficient of the term $M_0 \cdot T_{G,S,k,2}$ in $F_{G,f,1,5}$.
  - If $(\sum_{i=1}^{q_S+q_{IE}} d_{2,7,i}) + (d_{4,10,k} - d_{1,10,k}) = 0$, then $(\sum_{i=1}^{q_S+q_{IE}} d_{2,7,i}) = -(d_{4,10,k} - d_{1,10,k}) \neq 0$. And, in this case, the coefficient of the term $M_1 \cdot T_{G,S,k,2}$ in $F_{G,f,1,5}$ is $m^* \cdot (\sum_{i=1}^{q_S+q_{IE}} d_{2,7,i}) +(d_{4,10,k} - d_{1,10,k})$. Without loss of generality, letting $m^* \neq -1$, we have $F_{G,f,1,5}$ is non-zero .

3 **Case 3**: Otherwise, under the condition $d_{2,10,k} = d_{3,10,k} = (d_{4,10,k} - d_{1,10,k}) = 0$ for all $1 \leqq k \leqq q_S$, the terms $\sum_{i=1}^{q_S} (d_{l,10,k} (DID_{IE,i} + SID_{SE,k} + (M_0 + m_k \cdot M_1) \cdot T_{G,S,k,2}))$ in $F_{G,f,1,l}$, for l=1, 2, 3, 4, no longer affect $F_{G,f,1,5}$. In such a case, the polynomial $F_{G,f,1,l}$ can be simplified to

$$F_{G,f,1,l} = c_{l,1}+ c_{l,2}\, U_1+ c_{l,3}\, U_0+ c_{l,4}\, M_0+ c_{l,5}\, M_1$$
$$+ (d_{l,6,i} \cdot S_i)+\sum_{i=1}^{q_S+q_{IE}} \left(d_{l,7,i} \cdot T_i\right)$$
$$+\sum_{i=1}^{q_{IE}} (d_{l,8,i} \cdot DID_{IE,i})+\sum_{j=1}^{q_{SE}} (d_{l,9,j} \cdot SID_{SE,j}) \,,$$

where $DID_{IE,i} = X +(U_0+ID_{IE,i} \cdot U_1) \cdot T_{G,IE,i,2}$ for $1 \leqq i \leqq q_{IE}$. We discuss the following three cases:

- If $\sum_{i=1}^{q_{IE}} d_{3,8,i} \neq 0$, then $d_{3,8,i} \neq 0$ for some i. Hence, $F_{G,f,1,5}$ is non-zero since $d_{3,8,i}$ is the coefficient of the term $U_0^{\,2} \cdot T_{G,IE,j,2}$ in $F_{G,f,1,5}$.
- If $\sum_{i=1}^{q_{IE}} d_{2,8,i} \neq 0$, then at least one $d_{2,8,i}$ is non-zero for some i. Hence, the coefficient of the term $M_0 \cdot U_0 \cdot T_{G,IE,i,2}$ in $F_{G,f,1,5}$ is non-zero and so $F_{G,f,1,5}$ is non-zero .
- If $(\sum_{i=1}^{q_{IE}} d_{4,8,j} - \sum_{i=1}^{q_{IE}} d_{1,8,j}) \neq 0$, then at least one $(d_{4,8,j}-d_{1,8,j})$ is non-zero for some j. If $d_{3,7,i} +(d_{4,8,j}-d_{1,8,j}) \neq 0$, then $F_{G,f,1,5}$ is non-zero since $d_{3,7,i}+(d_{4,8,j}-d_{1,8,j})$ is the coefficient of the term $U_0 \cdot T_{G,IE,i,2}$ in $F_{G,f,1,5}$. Otherwise, $d_{3,7,i}= d_{1,8,j}-d_{4,8,j} \neq 0$. In this case, the coefficient of the term $U_1 \cdot T_{G,IE,j,2}$ in $F_{G,f,1,5}$ is $ID^* \cdot d_{3,7,i}-(d_{1,8,j}-d_{4,8,j}) \cdot ID_{IE,j}=(d_{1,8,j} - d_{4,8,j}) \cdot (ID^* - ID_{IE,j})$ which is non-zero since $ID^* \neq ID_{IE,j}$ for $1 \leqq j \leqq q_{IE}$. □

Theorem 2. In the generic bilinear group model, the non-leakage version Π_{NL} of the proposed LR-CLS scheme is provably secure against the Type II adversary (honest-but-curious KGC).

Proof: Let $A_{NL\text{-II}}$ be a Type II adversary who can break the non-leakage CLS scheme Π_{NL} while $A_{NL\text{-II}}$ is allowed to issue all the queries at most q times. The advantage of $A_{NL\text{-II}}$ is defined as the probability that $A_{NL\text{-II}}$ wins the following game $g_{NL\text{-II}}$ played with a challenger C.

Game $g_{NL\text{-II}}$: In the game $g_{NL\text{-II}}$, there are three phases, namely, *Setup*, *Queries* and *Forgery* phases. At the end of this game, $A_{NL\text{-II}}$ outputs a forgery signature. Three phases are described as below:

- ***Setup* phase**: In this phase, the challenger C prepares two initial-empty lists L_G and L_T to record the tuples in G and G_T, respectively. The forms of L_G and L_T are the same with those described in the game $g_{NL\text{-I}}$. The challenger C also maintains two lists L_{IK} and L_{SK} to record the tuples of users' initial keys and secret keys, respectively. At the end of this phase, C sends the bit strings of the public parameters to $A_{NL\text{-II}}$. Since the type II adversary $A_{NL\text{-II}}$ models an honest-but-curious KGC, C sends the bit string of the system secret key X along with the public parameters to $A_{NL\text{-II}}$.
- ***Queries* phase**: Since $A_{NL\text{-II}}$ models an honest-but-curious KGC, $A_{NL\text{-II}}$ can compute the user's initial key by issuing the oracles O_G, O_T and O_E. Meanwhile, $A_{NL\text{-II}}$ cannot perform the public key replacement query in this game. Hence in this phase, $A_{NL\text{-II}}$ can issue six kinds of queries as below:
  - ***Group oracle O_G*** ($\xi_{G,O,i,1}$, $\xi_{G,O,i,2}$, *operation*): This query is identical to O_G described in $g_{NL\text{-I}}$.

- ***Group oracle*** O_T ($\xi_{T,O,i,1}$, $\xi_{T,O,i,2}$, *operation*): This query is identical to O_T described in $g_{NL\text{-I}}$.
- ***Pairing oracle*** O_P($\xi_{G,P,i,1}$, $\xi_{G,P,i,2}$): This query is identical to O_P described in $g_{NL\text{-I}}$.
- ***Secret key extract query*** Q_{SE}($ID_{SE,i}$): This query is identical to Q_{SE} described in $g_{NL\text{-I}}$.
- ***Public key retrieve query*** Q_{PK}($ID_{PK,i}$): When $A_{NL\text{-II}}$ issues the i-th *Public key retrieve query* along with an identity $ID_{PK,i} \in Z_p^*$, the challenger C performs the following three steps:
  - C checks whether $ID_{PK,i}$ has been recorded in the list L_{IK}. If so, C obtains the polynomial of QID for $ID_{PK,i}$ in L_{IK}. Otherwise, C checks the records of the oracles O_G, O_T and O_E to obtain the polynomials of (DID, QID) for $ID_{PK,i}$ and update the list L_{IK} for $ID_{PK,i}$.
  - C checks whether $ID_{PK,i}$ has been recorded in the list L_{SK}. If so, C obtains the polynomial of RID for $ID_{PK,i}$ in L_{SK}. Otherwise, C performs the *Secret key extract query* ($ID_{PK,i}$) to set the polynomial of RID for $ID_{PK,i}$.
  - Finally, C answers the query by two bit strings of QID and RID by searching the lists L_G and L_T, respectively.
- ***Sign query*** Q_S($ID_{S,i}$, m_i) : For the i-th *Sign query*, upon receiving this query along with an identity $ID_{S,i} \in Z_p^*$ and a message $m_i \in Z_p^*$, the challenger C returns ($\xi_{G,S,i,1}$, $\xi_{G,S,i,2}$) to $A_{NL\text{-II}}$ at the end of this query. When C receives the query, C respectively obtains the user's private key DID and secret key SID from the lists L_{IK} and L_{SK} by the following steps:
  - C checks whether the user's private key of $ID_{S,i}$ has been recorded in the list L_{IK}. If so, C obtains DID of $ID_{S,i}$ in L_{IK}. Otherwise, C first checks the record of the oracles O_G, O_T and O_E to obtain the polynomials of (DID, QID) for $ID_{S,i}$ and then update the list L_{IK} for $ID_{S,i}$.
  - C checks whether the user's secret key SID of $ID_{S,i}$ has been recorded in the list L_{SK}. If so, C obtains SID of $ID_{S,i}$ in L_{SK}. Otherwise, C performs the query Q_{SE}($ID_{S,i}$) to obtain the tuple SID of $ID_{S,i}$.
  - Hence, C can obtain the polynomials $F_{G,IE,k,1}$ and $F_{G,\ SE,l,1}$ representing DID and SID, respectively.

The rest steps are identical to Q_S described in the game $g_{NL\text{-I}}$. Finally, C generates and returns (σ_1, σ_2)=($\xi_{G,S,i,1}$, $\xi_{G,S,i,2}$) to $A_{NL\text{-II}}$.

- ***Forgery*** phase: In this phase, the type II adversary $A_{NL\text{-II}}$ outputs a forgery signature (m^*, ID^*, σ^* = ($\xi_{G,f,i,1}{}^*$, $\xi_{G,f,i,2}{}^*$)). It is worth mentioning, where there are two restrictions: (1) ID^* has never been issued during the *Secret key extract query* Q_{SE}; (2) (m^*, ID^*) has never been issued during the *Sign query* Q_S.

In the real UF-CLS-ACMA game defined in Definition 1, the success probability in the game $g_{NL\text{-II}}$ is an upper bound of the advantage of $A_{NL\text{-II}}$. As the same arguments in Theorem 1, we can compute the success probability of $A_{NL\text{-II}}$ in game $g_{NL\text{-II}}$. By applying the same steps in Theorem 1 We have $|L_G|+ |L_T| \leqq 7+3q_O+2q_{SE}+4q_{PK}+4q_S+2$. Let $9 \leqq q_O+2q_{SE}$, we have $|L_G|+|L_T| \leqq 3q_O+2q_{SE}+4q_{PK}+3q_S+9 \leqq 4q$. Now we can compute the success probability of $A_{NL\text{-II}}$ in game $g_{NL\text{-II}}$. The advantage that $A_{NL\text{-II}}$ wins the game $g_{NL\text{-II}}$ is at most

$$(2/p)\binom{|L_G|}{2}+(4/p)\binom{|L_T|}{2}$$
$$+(3/p) \leqq (2/p)(|L_G|+|L_T|)^2 \leqq 32q^2/p,$$

which is negligible if q=*poly*(log$p$).

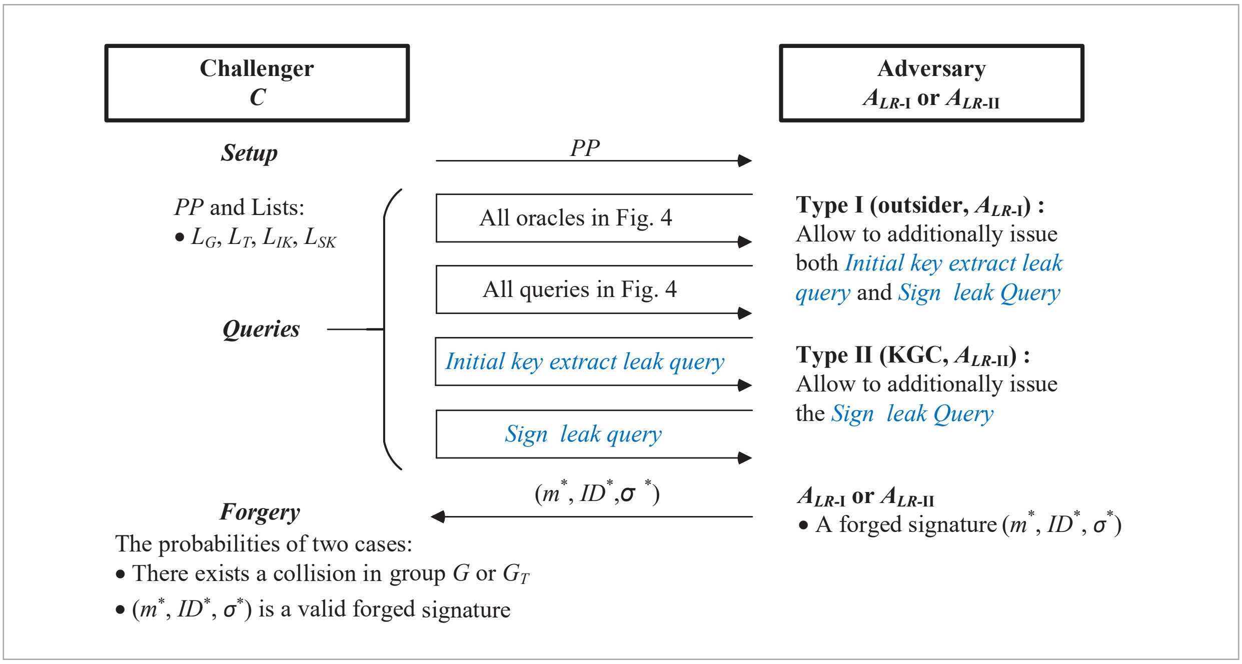
In Theorems 1 and 2, we have proved the security of the non-leakage version of the proposed LR-CLS scheme. In the following, based on the security of the non-leakage version, we prove that the proposed LR-CLS scheme under the continual leakage model is UF-LR-CLS-ACMA secure against Type I and Type II adversaries in Theorems 3 and 4, respectively. Figure 5 demonstrates the conceptual principle of the security games $g_{LR\text{-I}}$ and $g_{LR\text{-II}}$ employed in Theorems 3 and 4.

Theorem 3. In the generic bilinear group model, the proposed LR-CLS scheme is provably secure against the Type I adversary (outsider) under the continual leakage model.

Proof: We have proven that the non-leakage version of our proposed scheme is secure against the Type I adversary in Theorem 1. Here, the adversary is allowed to issue two extra queries, namely, *Initial key extract leak query* and *Sign leak query*. Hence we modify the game described in Theorem 1. Let $A_{LR\text{-I}}$ be a Type I adversary who can break our LR-CLS scheme Π_{LR} while $A_{LR\text{-I}}$ is allowed to issue all the queries at most q times. The advantage of $A_{LR\text{-I}}$ is defined as the probability

Figure 5

The conceptual principle of the security games $g_{LR\text{-I}}$ and $g_{LR\text{-II}}$ in Theorems 3 and 4



that $A_{LR\text{-I}}$ wins the following game $g_{LR\text{-I}}$ played with a challenger C.

Game $g_{LR\text{-I}}$: In the game $g_{LR\text{-I}}$, there are three phases, namely, *Setup*, *Queries* and *Forgery* phases. At the end of this game, $A_{LR\text{-I}}$ outputs a forgery signature. In *Queries* phase, $A_{LR\text{-I}}$ may issue ten kinds of queries in any order at most q times. Three phases are described as below:

- ***Setup* phase**: This phase is identical to that of the game $g_{NL\text{-I}}$.
- ***Queries* phase**: In addition to the eight kinds of queries in the game $g_{NL\text{-I}}$, $A_{LR\text{-I}}$ may issue two extra leakage queries (*Initial key extract leak query* and *Sign leak query*). In order to represent the leakage information, two leakage functions $f_{IE,i}$ and $h_{IE,i}$ model the ability of the adversary for *Extract*-1 and *Extract*-2 of the i-th *Initial key extract* round, respectively. Also, two leakage functions $f_{S,j}$ and $h_{S,j}$ are used to model the ability of the adversary for *Sign*-1 and *Sign*-2 of a user's j-th *Sign* round. Note that four leakage functions $f_{IE,i}$, $h_{IE,i}$, $f_{S,j}$ and $h_{S,j}$ respectively generate the leakage information $\Lambda f_{IE,i}$, $\Lambda h_{IE,i}$, $\Lambda f_{S,j}$ and $\Lambda h_{S,j}$. Meanwhile, four initial-empty lists $L_{f,IE}$, $L_{h,IE}$, $L_{f,S}$ and $L_{h,S}$ are used to record the related leakage functions and leakage information as follows:

$$L_{f,IE}=\{(f_{IE,i}, \Lambda f_{IE,i}), 1 \leqq i \leqq q_{IE}\},$$
$$L_{h,IE}=\{(h_{IE,i}, \Lambda h_{IE,i}), 1 \leqq i \leqq q_{IE}\},$$
$$L_{f,S}=\{(f_{S,j}, \Lambda f_{S,j}), 1 \leqq j \leqq q_S\},$$
$$L_{h,S}=\{(h_{S,j}, \Lambda h_{S,j}), 1 \leqq j \leqq q_S\}.$$

In addition, we describe two extra leakage queries as below:

- ***Initial key extract leak query*** $\mathbf{Q}_{IE\text{-}L}(f_{IE,i}, h_{IE,i}, i)$: For the i-th *Initial key extract leak query*, upon receiving this query along with two leakage functions $f_{IE,i}$ and $h_{IE,I}$ such that $|f_{IE,i}| \leq \lambda$ and $|h_{IE,i}| \leq \lambda$, C generates the leakage information $\Lambda f_{IE,i} = f_{IE,i}(S_{i-1,1}, \gamma_i, a_i)$ and $\Lambda h_{IE,i} = h_{IE,i}(S_{i-1,2}, TI_{IE}, a_i)$ and returns them to $A_{LR\text{-I}}$. Meanwhile, C adds $(f_{IE,i}, \Lambda f_{IE,i})$ and $(h_{IE,i}, \Lambda h_{IE,i})$ in the lists $L_{f,IE}$ and $L_{h,IE}$, respectively. It is worth mentioning, that $A_{LR\text{-I}}$ can ask the $\mathbf{Q}_{IE\text{-}L}$ for the same identity only once.
- ***Sign leak query*** $\mathbf{Q}_{S\text{-}L}(f_{S,j}, h_{S,j}, j)$: For the j-th *Sign query*, upon receiving this query along with two leakage functions $f_{S,j}$ and $h_{S,j}$ such that $|f_{S,j}| \leq \lambda$

and $|h_{S,j}| \le \lambda$, C generates the leakage information $\Lambda f_{S,j} = f_{S,j}(DID_{j-1,1}, \eta_j, b_j, c_j)$ and $\Lambda h_{S,j} = h_{S,j}(DID_{j-1,2}, TI_S, b_j, c_j)$ and returns them to $A_{LR\text{-}I}$. Meanwhile, C adds $(f_{S,j}, \Lambda f_{S,j})$ in $L_{f,S}$ and $(h_{S,j}, \Lambda h_{S,j})$ in $L_{h,S}$.

- ***Forgery* phase**: In this phase, the type I adversary $A_{LR\text{-}I}$ outputs a forgery signature $(m^*, ID^*, \sigma^*=(\xi_{G,f,i,1}{}^*, \xi_{G,f,i,2}{}^*))$. It is worth mentioning, where there are two restrictions: (1) ID^* has never been issued during the *Initial key extract query* Q_{IE}; (2) (m^*, ID^*) has never been issued during the *Sign query* Qs.

By making use of the leakage functions, it is obvious that the success probability (advantage) of $A_{LR\text{-}I}$ in $g_{LR\text{-}I}$ is higher than that of $A_{NL\text{-}I}$ in the game $g_{NL\text{-}I}$. For the *Initial key extract leak query* with two leakage functions $f_{IE,I}$ and $h_{IE,i}$, the adversary $A_{LR\text{-}I}$ can obtain partial information of $(S_{i-1,1}, \gamma_i, a_i)$ and $(S_{i-1,2}, TI_{IE}, a_i)$ by the leakage information $\Lambda f_{IE,i}$ and $\Lambda h_{IE,I}$, respectively, as follows.

- γ_i: The random value γ_i is involved in the computation of the *Initial key extract query* to generate the initial key of the user with identity $ID_{IE,i}$. If $A_{LR\text{-}I}$ has issued the *Initial key extract query* on $ID_{IE,i}$, any forgery signature for $ID_{IE,i}$ is not accepted in the *Forgery* phase. In such a case, the leakage of γ_i is useless to generate a signature for $A_{LR\text{-}I}$ in the *Forgery* phase.
- $(S_{i-1,1}, S_{i-1,2})$: The partial information of $(S_{i-1,1}, S_{i-1,2})$ could help $A_{LR\text{-}I}$ to learn the partial information of the system secret key X. So, $A_{LR\text{-}I}$ can get at most 2λ bits information of X.
- a_i: The random value a_i is involved in the computation of the current system secret key $(S_{i,1}, S_{i,2})$, but it is independent to the system secret key X. In such a case, $A_{LR\text{-}I}$ can learn at most λ bits of $S_{i,1}$ and $S_{i,2}$, respectively.
- TI_{IE}: The temporary information TI_{IE} is useless to obtain the user's initial private key. In addition to that, it is also useless to generate a signature for $A_{LR\text{-}I}$ in the *Forgery* phase.

On the other hand, for the *Sign leak query* with two leakage functions $f_{S,j}$ and $h_{S,j}$, the adversary $A_{LR\text{-}I}$ can obtain partial information of $(DID_{j-1,1}, \eta_j, b_j, c_j)$ and $(DID_{j-1,2}, TI_S, b_j, c_j)$ by the leakage information $\Lambda f_{S,j}$ and $\Lambda h_{S,j}$, respectively, as follows.

- η_j: The random value η_j is involved in the computation of generating a signature on $(ID_{S,j}, m_j)$. If $A_{LR\text{-}I}$ has issued the *Sign query* on $(ID_{S,j}, m_j)$, any forgery signature for $(ID_{S,j}, m_j)$ is not accepted in the *Forgery* phase. In such a case, the leakage of η_j is useless to generate a signature for $A_{LR\text{-}I}$ in the *Forgery* phase.
- $(DID_{j-1,1}, DID_{j-1,2})$: The partial information of $(DID_{j-1,1}, DID_{j-1,2})$ could help $A_{LR\text{-}I}$ to learn the partial information of the user's first initial key DID_0. So, $A_{LR\text{-}I}$ can get at most 2λ bits information of DID_0.
- b_j: The random value b_j is involved in the computation of generating the user's initial key $(DID_{j,1}, DID_{j,2})$. In such a case, $A_{LR\text{-}I}$ learns at most λ bits information about $DID_{j,1}$ and $DID_{j,2}$, respectively.
- c_j: The random value c_j is involved in the computation of generating the user's secret key $(SID_{j,1}, SID_{j,2})$. In such a case, $A_{LR\text{-}I}$ learns at most λ bits information about $SID_{j,1}$ and $SID_{j,2}$, respectively.
- TI_S: The temporary information TI_S is useless to generate a signature for $A_{LR\text{-}I}$ in the *Forgery* phase.

Here, we evaluate the probability that $A_{LR\text{-}I}$ wins the game $g_{LR\text{-}I}$, denoted by $\Pr_{LR\text{-}I}$. Note that since the type I adversary $A_{LR\text{-}I}$ can obtain the secret key of any entity, $A_{LR\text{-}I}$ can forge a valid signature whenever she/he obtains the system secret key X or the target user's initial key DID_0. In the following, we define three events of $\Pr_{LR\text{-}I}$, namely, SSK, UIK and VFS.

1. The event SSK denotes that $A_{LR\text{-}I}$ can get the system secret key X completely by two leakage functions $f_{IE,i}$ and $h_{IE,I}$.
2. The event UIK denotes that $A_{LR\text{-}I}$ can get the target user's initial key DID_0 completely by two leakage functions $f_{S,j}$ and $h_{S,j}$.
3. The event VFS denotes that $A_{LR\text{-}I}$ can generate a valid forgery signature.

Meanwhile, we denote the events $\overline{SSK}$ and $\overline{UIK}$ are the complement events of SSK and UIK respectively. Therefore, the probability that $A_{LR\text{-}I}$ wins the game $g_{LR\text{-}I}$ is bounded by

$$
\begin{aligned}
&\Pr_{LR\text{-}I} = \Pr[VFS] \\
&= \Pr[VFS \wedge SSK] + \Pr[VFS \wedge \overline{SSK}] \\
&= \Pr[VFS \wedge SSK] + \Pr[VFS \wedge \overline{SSK} \wedge UIK] \\
&\quad + \Pr[VFS \wedge \overline{SSK} \wedge \overline{UIK}] \\
&= \Pr[VFS \wedge SSK] + \Pr[VFS \wedge \overline{SSK} \wedge UIK] \\
&\quad + \Pr[VFS \mid \overline{SSK} \wedge \overline{UIK}] \cdot \Pr[\overline{SSK} \wedge \overline{UIK}].
\end{aligned}
$$

Since $\Pr[VFS \wedge SSK] \leqq \Pr[SSK]$, $\Pr[VFS \wedge \overline{SSK} \wedge UIK] \leqq \Pr[\overline{SSK} \wedge UIK]$ and $\Pr[\overline{SSK} \wedge \overline{UIK}] \leqq 1$, we have

$$\Pr_{LR\text{-}I} \leqq \Pr[SSK] + \Pr[\overline{SSK} \wedge UIK] + \Pr[VFS \mid \overline{SSK} \wedge \overline{UIK}].$$

By Lemmas 4, 5 and 6 later, we respectively evaluate three probabilities $\Pr[SSK]$, $\Pr[\overline{SSK} \wedge \overline{UIK}]$ and $\Pr[\overline{SSK} \wedge UIK]$. Thus, we have

$$\begin{aligned}\Pr_{LR\text{-I}} &\leqq \Pr[SSK]+\Pr[VFS \mid \overline{SSK} \wedge \overline{UIK}]+ \Pr[\overline{SSK} \wedge UIK]\\ &\leqq O((q^2/p)2^{2\lambda}) + O((q^2/p)2^{\lambda}) + O((q^2/p)2^{2\lambda})\\ &\leqq O((q^2/p)2^{2\lambda}).\end{aligned}$$

Hence, the advantage of $A_{LR\text{-I}}$ breaking the proposed LR-CLS scheme is $O((q^2/p)2^{2\lambda})$. By Corollary 1, if $\lambda << \frac{\log(p)}{2}$, the proposed LR-CLS scheme is existential unforgeability against adaptive chosen-message attacks.

Lemma 4. $\Pr[SSK] \leqq O((q^2/p)2^{2\lambda})$.

Proof. In the *Initial key extract* phase of the proposed LR-CLS scheme, a user's initial key is a signature on the user's identity ID generated by Galindo and Vivek's LRS signature scheme in [18]. Therefore, by applying the Lemma 5, we have $\Pr[SSK] \leqq O((q^2/p)2^{2\lambda})$. □

Lemma 5. $\Pr[VFS \mid \overline{SSK} \wedge \overline{UIK}] \leqq O((q^2/p)2^{\lambda})$.

Proof. In Theorem 1, we have proved that an adversary has the success probability $O(q^2/p)$ to break the non-leakage version Π_{NL} of the proposed LR-CLS scheme. For both events $\overline{SSK}$ and $\overline{UIK}$, $A_{LR\text{-I}}$ can learn at most λ bits information about the user's current private key ($DID_{j,1}$, $DID_{j,2}$). Therefore, the probability of $A_{LR\text{-I}}$ forging a valid signature by using at most λ bits leakage information is $\Pr[VFS \mid \overline{SSK} \wedge \overline{UIK}] \leqq O((q^2/p)2^{\lambda})$.

Lemma 6. $\Pr[\overline{SSK} \wedge UIK] \leqq O((q^2/p)2^{2\lambda})$.

Proof. For the event $\overline{SSK}$, $A_{LR\text{-I}}$ is unable to obtain any useful information by *Initial key extract leak query*, but may obtain the useful information by *Sign leak query* to forge a signature. However, $A_{LR\text{-I}}$ may obtain useful information by *Sign leak query.* In this case, $\Pr[\overline{SSK} \wedge UIK]$ is the event that $A_{LR\text{-I}}$ can obtain the partial information of a user's first initial key by the *Sign leak query* with two leakage functions $f_{S,i}$ and $h_{S,i}$. Hence, the probability to forge a signature under the event $\overline{SSK} \wedge UIK$ is identical to the probability $\Pr[SSK]$ of generating a user's first initial key under the event SSK. Therefore, by Lemma 4, we have $\Pr[\overline{SSK} \wedge UIK] \leqq O((q^2/p)2^{2\lambda})$.

Theorem 4. In the generic bilinear group model, the proposed LR-CLS scheme is provably secure against the Type II adversary (honest-but-curious KGC) under the continual leakage model.

Proof: We have proven that the non-leakage version of our proposed scheme is secure against the Type II adversary in Theorem 2. Here, the adversary is allowed to issue an extra query, i.e. *Sign leak query*. Let $A_{LR\text{-II}}$ be a Type I adversary who can break our LR-CLS scheme Π_{LR} while $A_{LR\text{-II}}$ is allowed to issue all the queries at most q times. The advantage of $A_{LR\text{-II}}$ is defined as the probability that $A_{LR\text{-I}}$ wins the following game $g_{LR\text{-II}}$ played with a challenger C.

Game $g_{LR\text{-II}}$: In the game $g_{LR\text{-}II}$, there are three phases, namely, *Setup*, *Queries* and *Forgery* phases. At the end of this game, $A_{LR\text{-II}}$ outputs a forgery signature. In *Queries* phase, $A_{LR\text{-II}}$ may issue seven kinds of queries in any order at most q times. Three phases are described as below:

- ***Setup* phase**: This phase is identical to that of the game $g_{NL\text{-}II}$.
- ***Queries* phase**: In addition to the six kinds of queries in the game $g_{NL\text{-}II}$, $A_{LR\text{-II}}$ may issue one extra leakage query (*Sign leak query*). In order to represent the leakage information, two leakage functions $f_{S,j}$ and $h_{S,j}$ are used to model the ability of the adversary for *Sign*-1 and *Sign*-2 of a user's j-th *Sign* round. Note that two leakage functions $f_{S,j}$ and $h_{S,j}$ respectively generate the leakage information $\Lambda f_{S,j}$ and $\Lambda h_{S,j}$. Meanwhile, two initial-empty lists $L_{f,S}$ and $L_{h,S}$ are used to record the related leakage functions and leakage information:

$$L_{f,S}=\{(f_{S,j}, \Lambda f_{S,j}), 1 \leqq j \leqq q_S\}, L_{h,S}=\{(h_{S,j}, \Lambda h_{S,j}), 1 \leqq j \leqq q_S\}.$$

  - ***Sign leak query*** **($f_{S,j}$, $h_{S,j}$, j)**: This query is identical to the *Sign leak query* described in the game $g_{LR\text{-I}}$.
- ***Forgery* phase**: In this phase, the type II adversary $A_{LR\text{-II}}$ outputs a forgery signature (m^*, ID^*, $\sigma^*=(\xi_{G,f,i,1}{}^*, \xi_{G,f,i,2}{}^*)$). It is worth mentioning, that there are two restrictions: (1) ID^* has never been issued during the *Secret key extract query* Q_{SE}; (2) (m^*, ID^*) has never been issued during the *Sign query Qs*.

By making use of the leakage functions, it is obvious that the success probability (advantage) of $A_{LR\text{-II}}$ in $g_{LR\text{-II}}$ is higher than that of $A_{NL\text{-II}}$ in the game $g_{NL\text{-II}}$. For the *Sign leak query* with two leakage functions $f_{S,j}$ and $h_{S,j}$, the adversary $A_{LR\text{-II}}$ can obtain partial information of $(DID_{j-1,1}, \eta_j, b_j, c_j)$ and $(DID_{j-1,2}, b_j, c_j)$ by the leakage information $\Lambda f_{S,j}$ and $\Lambda h_{S,j}$, respectively. The discussions on the partial leakage information of η_j, $(DID_{j-1,1}, DID_{j-1,2})$, b_j and c_j are the same with those in Theorem 3.

Here, we evaluate the probability that $A_{LR\text{-II}}$ wins the game $g_{LR\text{-II}}$, denoted by $\Pr_{LR\text{-II}}$. Note that since the type II adversary $A_{LR\text{-II}}$ has obtained the system secret key X, $A_{LR\text{-II}}$ can generate the user's initial key DID_0 of any entity. In such a case, $A_{LR\text{-II}}$ can forge a valid signature if $A_{LR\text{-II}}$ gets the user's secret key SID. In the following, we define two events of $\Pr_{LR\text{-II}}$: (1) The event USK denotes that $A_{LR\text{-II}}$ can get the user's secret key SID completely by two leakage functions $f_{S,j}$ and $h_{S,j}$. The event $\overline{USK}$ is the complement event of USK; (2) The event VFS denotes that $A_{LR\text{-II}}$ can generate a valid forgery signature. Therefore, the probability that $A_{LR\text{-II}}$ wins the game $g_{LR\text{-II}}$ is bounded by

$$\Pr_{LR\text{-II}} = \Pr[VFS] = \Pr[VFS \wedge USK] + \Pr[VFS \wedge \overline{USK}]$$
$$= \Pr[VFS \wedge USK] + \Pr[VFS \mid \overline{USK}] \cdot \Pr[\overline{USK}].$$

Since $\Pr[VFS \wedge USK] \leqq \Pr[USK]$ and $\Pr[\overline{USK}] \leqq 1$, we have $\Pr_{LR\text{-II}} \leqq \Pr[USK] + \Pr[VFS \mid \overline{USK}]$.

By Lemmas 7 and 8 later, we respectively evaluate two probabilities $\Pr[USK]$ and $\Pr[VFS \mid \overline{USK}]$. Thus, we have

$$\Pr_{LR\text{-II}} \leqq \Pr[USK] + \Pr[VFS \mid \overline{USK}],$$
$$\leqq O((1/p)^* 2^{2\lambda}) + O((q^2/p)^* 2^{2\lambda})$$
$$\leqq O((q^2/p)^* 2^{2\lambda}).$$

Hence, the advantage of $A_{LR\text{-II}}$ breaking the proposed LR-CLS scheme is $O((q^2/p)2^{2\lambda})$. By Corollary 1, if $\lambda \ll \frac{\log(p)}{2}$, the proposed LR-CLS scheme is existential unforgeability against adaptive chosen-message attacks.

Lemma 7. $\Pr[USK] \leqq O((1/p)^* 2^{2\lambda})$.

Proof. Since $A_{LR\text{-II}}$ can learn at most 2λ bits information for the user current secret key by the *Sign leak query* with two leakage functions $f_{S,i}$ and $h_{S,i}$, we have $\Pr[USK] \leqq O((1/p)^* 2^{2\lambda})$.

Lemma 8. $\Pr[VFS \mid \overline{USK}] \leqq O((q^2/p)^* 2^{\lambda})$.

Proof. In Theorem 3, we have proved that an adversary has the success probability $O(q^2/p)$ to break the non-leakage version Π_{NL} of the proposed LR-CLS scheme. For the event $\overline{USK}$, $A_{LR\text{-II}}$ can learn at most λ bits information about the user's current secret key $(SID_{j,1}, SID_{j,2})$. Therefore, the probability of $A_{LR\text{-II}}$ forging a valid signature by using at most λ bits leakage information is $\Pr[VFS \mid \overline{USK}] \leqq O((q^2/p)^* 2^{\lambda})$.

6. Performance Analysis

Our proposed scheme is the first LR-CLS scheme under the continual leakage model. To achieve leakage-resilient property, we added some extra computations in our scheme so that the performance is not better than that of the previously proposed CLS schemes [19-23, 44, 46]. Fortunately, the performance of our LR-CLS scheme is still suitable for mobile devices. For practicality, a suitable bilinear pairing group to implement our LR-IBS scheme is the pairing-friendly curves presented by Scott in [35]. In the following, we demonstrate the performance analysis of the proposed LR-CLS scheme.

For convenience, we define the following notations to analyze the computational costs.

- T_p: The time of executing a bilinear pairing operation $e: G \times G \rightarrow G_T$.
- T_e: The time of executing an exponentiation operation in G.

Here, we analyze the computational costs of the proposed LR-CLS scheme in terms of *Initial key extract*, *Sign* and *Verify* phases. The *Initial key extract* phase including *Extract*-1 and *Extract*-2 sub-algorithms requires $5T_e$ to update the current system secret key to $(S_{i,1}, S_{i,2})$ and produce the first initial key of a user with identity ID. The *Sign* phase including *Sign*-1 and *Sign*-2 sub-algorithms requires $7T_e$ to generate a signature σ while updating the current private key of the signer with identity ID. In addition, the *Verify* phase requires $3T_p+2T_e$ to validate a signature.

We use the newest implementation results of the related operations in the generic bilinear group made by Galindo *et al.* [17] to measure the computational cost of the proposed LR-CLS scheme. Their implementation environment is presented as follows. The processor is an ARM Cortex-M3 CPU. The group G is an elliptic curve group over F_p with a bit-length of 254 bits while G_T is a subgroup of the multiplicative group of the extension field F_{p12}. The required computational costs (in 106 clock cycles) of T_e and T_p are 4.5 and 65, respectively. Here, the multiplication operation is ignored as compared with T_e and T_p. Table 1 lists the required computational costs (in 106 clock cycles) of the *Initial key extract*, *Sign* and *Verify* phases in the proposed LR-CLS scheme. It is obvious that the proposed LR-CLS scheme is well suitable for mobile devices.

Table 1

Computational costs of the proposed LR-CLS scheme

Phase	Required operations	Running cost (in 106 clock cycles)
Initial key extract	$5T_e$	22.5
Sign	$7T_e$	31.5
Verify	$3T_p$+$2T_e$	204

7. Conclusions

In this article, we proposed the first LR-CLS scheme under the continual leakage model. We defined the new security notions for LR-CLS schemes under the continual leakage model. In the security notions, there are two kinds of attackers, namely, Type I adversary (outsider) and Type II adversary (honest-but-curious KGC). Both kinds of attackers are extended from the security notions of traditional certificateless signature (CLS) schemes by adding the *Initial key extract leak query* and the *Sign leak query*. Type I adversary may obtain not only the leakage information of a user's initial key of the private key in the *Sign* phase, but also the leakage information of the KGC's system secret key in the *Initial key extract* phase. Type II adversary knows the initial key of any entity while obtaining the leakage information of a user's secret key of the private key in the *Sign* phase. In the generic bilinear group model, we demonstrate that our LR-CLS scheme possesses existential unforgeability against adaptive chosen-message attacks for both Type I and Type II adversaries under the continual leakage model.

Acknowledgment

This research was partially supported by Ministry of Science and Technology, Taiwan, under contract no. MOST105-2221-E-018-013.

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., Naor, M., Segev, G., Walfish, S., Wichs, D. Public-Key Encryption in the Bounded-Retrieval Model. In: EUROCRYPT'10, LNCS, 6110, Springer, Belin-Heidelberg, 2010, 113–134. https://doi.org/10.1007/978-3-642-13190-5_6
3. 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
4. Biham, E., Carmeli, Y., Shamir, A. Bug Attacks. In: CRYPTO'08, LNCS, 5157, Springer, Belin-Heidelberg, 2008, 221–240. https://doi.org/10.1007/978-3-540-85174-5_13
5. Boneh, D., Boyen, X., Goh, E. J. Hierarchical Identity-Based Encryption with Constant Size Ciphertext. In: EUROCRYPT'05, LNCS, 3494, Springer, Belin-Heidelberg, 2005, 440–456. https://doi.org/10.1007/11426639_26
6. Boneh, D., Demillo, R. A., Lipton, R. J. On the Importance of Checking Cryptographic Protocols for Faults. In: EUROCRYPT'97, LNCS, 1233, Springer, Belin-Heidelberg, 1997, 37–51. https://doi.org/10.1007/3-540-69053-0_4
7. 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
8. Boneh, D., Lynn, B., Shacham, H. Short signatures from

the Weil pairing. In: ASIACRYPT'01, LNCS, 2248, Springer, Belin-Heidelberg, 2001, 514–532. https://doi.org/10.1007/3-540-45682-1_30

9. Brakerski, Z., Kalai, Y. T., Katz, J., Vaikuntanathan, V. Cryptography Resilient to Continual Memory Leakage. Proceedings of 51st Annual IEEE Symposium on Foundations of Computer Science, 2010, 501–510.

10. 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

11. 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

12. Dodis, Y., Haralambiev, K., Lopez-Alt, A., Wichs, D. Cryptography Against Continuous Memory Attacks. Proceedings of 51st Annual IEEE Symposium on Foundations of Computer Science, 2010, 511–520. https://doi.org/10.1109/FOCS.2010.56

13. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM Journal on Computing, 2008, 38(1), 97–139. https://doi.org/10.1137/060651380

14. 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

15. Faust, S., Hazay, C., Nielsen, J. B., Nordholt, P. S., Zottarel, A. Signature Schemes Secure Against Hard-to-Invert Leakage. In: ASIACRYPT'12, LNCS, 7658, Springer, Belin-Heidelberg, 2012, 98–115. https://doi.org/10.1007/978-3-642-34961-4_8

16. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G. N. Leakage-Resilient Signatures. In: TCC'10, LNCS, 5978, Springer, Belin-Heidelberg, 2010, 343–360. https://doi.org/10.1007/978-3-642-11799-2_21

17. 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

18. Galindo, D., Virek, S. A Practical Leakage-Resilient Signature Scheme in the Generic Group Model. In: SAC'12, LNCS, 7707, Springer, Belin-Heidelberg, 2013, 50–65. https://doi.org/10.1007/978-3-642-35999-6_4

19. Hu, B., Wong, D., Zhang, Z., Deng, X. Key Replacement Attack Against a Generic Construction of Certificateless Signature. In: ACISP'06, LNCS, 4058, Springer, Belin-Heidelberg, 2006, 235–346. https://doi.org/10.1007/11780656_20

20. Hu, B., Wong, D., Zhang, Z., Deng, X. Certificateless Signature: A New Security Model and an Improved Generic Construction. Designs, Codes and Cryptography, 2007, 42(2), 109–126. https://doi.org/10.1007/s10623-006-9022-9

21. Huang, X., Mu, Y., Susilo, W., Wong, D., Wu, W. Certificateless Signature Revisited. In: ACISP'07, LNCS, 4586, Springer, Belin-Heidelberg, 2007, 308–322. https://doi.org/10.1007/978-3-540-73458-1_23

22. Huang, X., Susilo, W., Mu, Y., Zhang, F. On the Security of Certificateless Signature Schemes from Asiacrypt 2003. In: CANS'05, LNCS, 3810, Springer, Belin-Heidelberg, 2005, 13–25. https://doi.org/10.1007/11599371_2

23. 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

24. Hwang, Y. H., Liu, J. K., Chow, S. S. M. Certificateless Public Key Encryption Secure Against Malicious KGC Attacks in the Standard Model. Journal of Universal Computer Science, 2008, 14(3), 463–480.

25. 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

26. Kiltz, E., Pietrzak, K. Leakage Resilient Elgamal Encryption. In: ASIACRYPT'10, LNCS, 6477, Springer, Belin-Heidelberg, 2010, 595–612. https://doi.org/10.1007/978-3-642-17373-8_34

27. 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

28. Kocher, P. C. 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

29. Libert, B., Quisquater, J. J. On Constructing Certificateless Cryptosystems from Identity Based Encryption. In: PKC'06, LNCS, 3958, Springer, Belin-Heidelberg, 2006, 474–490. https://doi.org/10.1007/11745853_31

30. Lin, H. Y. Secure Certificateless Two-Party Key Agreement with Short Message. Information Technology and Control, 2016, 45(1), 71–76. https://doi.org/10.5755/j01.itc.45.1.12595

31. Maurer, U., Wolf, S. Lower Bounds on Generic Algorithms in Groups. In: EUROCRYPT'98, LNCS, 1403, Springer, Belin-Heidelberg, 1998, 72–84. https://doi.org/10.1007/BFb0054118

32. Naor, M., Segev, G. Public-Key Cryptosystems Resilient to Key Leakage. In: CRYPTO'09, LNCS, 5677, Springer, Belin-Heidelberg, 2009, 18–35. https://doi.org/10.1007/978-3-642-03356-8_2

33. 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

34. Schwartz, J. T. Fast Probabilistic Algorithms for Verification of Polynomial Identities. Journal of the ACM, 1980, 27(4), 701–717. https://doi.org/10.1145/322217.322225

35. 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

36. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In: CRYPTO'84, LNCS, 196, Springer, Belin-Heidelberg, 1984, 47–53.

37. 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

38. 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

39. 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

40. Tsai, T. T., Tseng, Y. M., Huang, S. S. Efficient Revocable Certificateless Public Key Encryption with a Delegated Revocation Authority. Security and Communication Networks, 2015, 8(18), 3713–3725. https://doi.org/10.1002/sec.1294

41. Waters, B. Efficient Identity-Based Encryption Without Random Oracles. In: EUROCRYPT'05, LNCS, 3494, Springer, Belin-Heidelberg, 2005, 114–127. https://doi.org/10.1007/11426639_7

42. 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

43. Xiong, H., Yuen, T. H., Zhang, C., Yiu, S. M., He, Y. J. Leakage-Resilient Certificateless Public Key Encryption. Proceedings of the 1st ACM Workshop on Asia Public-Key Cryptography, 2013, 13–22. https://doi.org/10.1145/2484389.2484394

44. Yu, Y., Mu, Y., Wang, G., Xia, Q., Yang, B. Improved Certificateless Signature Scheme Provably Secure in the Standard Model. IET Information Security, 2012, 6(2), 102–110. https://doi.org/10.1049/iet-ifs.2011.0004

45. Yuen, T. H., Chow, S. S. M., Zhang, Y., Yiu, S. M. Identity-Based Encryption Resilient to Continual Auxiliary Leakage. In: EUROCRYPT'12, LNCS, 7237, Springer, Belin-Heidelberg, 2012, 117–134. https://doi.org/10.1007/978-3-642-29011-4_9

46. Yum, D. H., Lee, P. J. Generic Construction of Certificateless Encryption. In: ICCSA'04, LNCS, 3043, Springer, Belin-Heidelberg, 2004, 802–811. https://doi.org/10.1007/978-3-540-24707-4_93

47. 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

48. Zippel, R. Probabilistic Algorithms for Sparse Polynomials. In: EUROSAM'79, LNCS, 72, Springer, Belin-Heidelberg, 1979, 216–226. https://doi.org/10.1007/3-540-09519-5_73