IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 May 2023, accepted 27 May 2023, date of publication 31 May 2023, date of current version 7 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3281612

RESEARCH ARTICLE

Leakage-Resilient Certificateless Signcryption Scheme Under a Continual Leakage Model

TUNG-TSO TSAI[1], **YUH-MIN TSENG**[2], **(Member, IEEE), AND SEN-SHAN HUANG**[2]
[1]Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan
[2]Department of Mathematics, National Changhua University of Education, Changhua City 500, Taiwan

Corresponding author: Yuh-Min Tseng (ymtseng@cc.ncue.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract MOST110-2221-E-018-006-MY2 and Contract MOST110-2221-E-018-007-MY2.

ABSTRACT Signature can be used to verify the integrity of both a message and the identity of a signer, whereas encryption can be used to ensure the confidentiality of a message. In the past, cryptography researchers have studied and proposed numerous certificateless signcryption (CLSC) schemes to combine the benefits of both signature and encryption. However, these schemes may not be robust enough to withstand side-channel attacks. Through such attacks, an attacker can constantly retrieve a portion of a private key of the system, and could eventually recover the entire private key. Leakage-resilient certificateless signcryption (LR-CLSC) can ensure its security when the attacker launches such attacks. As far as we know, the existing LR-CLSC schemes can only guarantee the security under a bounded leakage model, where the portion of the private key that an attacker can obtain through side-channel attacks is limited. In this paper, we propose the *first* LR-CLSC scheme under a continual leakage model. Also, we demonstrate the proposed scheme is secure for the existential unforgeability and the ciphertexts indistinguishability against attackers with side-channel attacking abilities.

INDEX TERMS Leakage-resilience, side-channel attacks, certificateless, signcryption.

I. INTRODUCTION

Signature and encryption are two important functions of public key cryptography [1]. Signature [2] can be used to verify the integrity of both a message and the identity of a signer, whereas encryption [3], [4] can be used to ensure the confidentiality of a message. Some cryptographic mechanisms [5], [6], [7] that ensure the confidentiality and integrity of messages have also been proposed and applied in the context of IoT environments. If a message is signed first using a signature mechanism and then encrypted using an encryption mechanism, the required computation cost is the sum of the two mechanisms. It is natural to combine both signature and encryption procedures into one mechanism in order to reduce total required computation cost. Based on this idea, a novel cryptographic primitive, named signcryption, was proposed by Zheng [8]. Subsequently, a large number of studies related to signcryption have been proposed [9], [10], [11], [12], [13].

Signcryptions mentioned above are constructed under the traditional public key systems. Despite the many benefits of public key systems, they still have some drawbacks, one of which is certificate management. Certificates are used to verify the validation of a user's public key and identity. They contain information related to the public key, such as the owner's name, organization, expiration date, etc. Certificate management refers to the processes of operating these certificates, including issuing, revoking, updating, and storing. To avoid the problem of certificate management, Malone-Lee [14] employed the identity-based concept [15] to construct an identity-based signcryption (IBSC). Also, Malone-Lee demonstrated that the IBSC scheme is secure for the existential unforgeability and the ciphertexts indistinguishability. However, Libert and Quisquater [16] identified two weaknesses in Malone-Lee's scheme, which suffers from signature visibility attacks and ciphertexts

The associate editor coordinating the review of this manuscript and approving it for publication was SK. Hafizul Islam.

distinguishability. To remove potential security vulnerabilities, Libert and Quisquater [16] also proposed three IBSC schemes which satisfy the forward security. Shortly after, Boyen [17] developed a new IBSC scheme that provides forward security, ciphertext unlinkability and anonymity. To increase efficiency, Chen and Malone-Lee [18] proposed an improved IBSC. So far, several IBSC mechanisms [19], [20], [21] have been explored and studied.

IBSC schemes mentioned above are constructed under the identity-based public key systems (IB-PKS). However, IB-PKS has a known significant weakness, namely, the key escrow problem. This refers to the fact that, in an IB-PKS, a trusted third party (the private key generator, PKG) holds a copy of the private key associated with each user's public key. However, the PKG is necessary for the system, because it is responsible for generating private keys and distributing them to users. Fortunately, the certificateless SC (CLSC) schemes under the certificateless public key system [22] offer a solution that can simultaneously avoid the problems of both certificate management and key escrow. Barbosa and Farshim [23] utilized the bilinear pairing to construct the first CLSC scheme. However, Liu et al. [24] pointed out that the CLSC scheme [23] is vulnerable to attacks from a malicious-but-passive key generator center (KGC) and introduced a new CLSC scheme [24]. However, it was later found to be vulnerable to public key replacement attacks so that it loses both confidentiality and unforgeability [25]. In response to these attacks and limitations of the existing CLSC solutions, Rastegart et al. [26] proposed a practical scheme under the standard model that can withstand known session-specific temporary information attacks.

Numerous CLSC schemes have been proposed in the past, but they may not be robust enough to withstand side-channel attacks. Through such attacks, an attacker can constantly retrieve a portion of a private key of the system, and could eventually recover the entire private key. Leakage-resilient cryptography can ensure the security when the attacker launches such attacks. This type of cryptography utilizes two leakage models: bounded and continual (or unbounded). Both models impose limits on the length of leaked bits from a private key used in each cryptographic computation, and this length is tied to a pre-defined security parameter. The practicality of the bounded leakage model is limited because it restricts the total number of bits from a private key that can be disclosed to attackers during the system lifecycle to a fixed amount [27], [28]. The continual leakage model allows attackers to gradually acquire portions (partial bits) of private keys used in each computation, allowing the leakage unbounded [29], [30], [31], [32], [33]. Although some LR-CLSC schemes [34], [35] have been proposed, the schemes can only guarantee the security under a bounded leakage model.

According to our best knowledge, there is hardly any research in the LR-CLSC under a continual leakage model. In this paper, we propose the *first* leakage-resilient certificateless signcryption (LR-CLSC) scheme under a continual leakage model. We first present the syntax of LR-CLSC, and then model the security notions of LR-CLSC. Based on the syntax of LR-CLSC, a concrete scheme will be proposed. Finally, we demonstrate the proposed scheme is secure for the existential unforgeability and the ciphertexts indistinguishability against attackers with side-channel attacking abilities.

This paper consists of six sections. Section II presents some preliminary concepts. Section III introduces the syntax and security model for LR-CLSC schemes. The proposed LR-CLSC scheme is illustrated in Section IV, and its security is formally proved in Section V. Comparisons and concluding remarks are provided in Section VI and Section VII, respectively.

II. RELATED WORK

In traditional cryptographic systems, the security of ciphertext is based on the assumption that the secret key can be kept confidential. However, in many real-world scenarios, this assumption is often unrealistic. For instance, some security systems may be vulnerable to side-channel attacks [36], [37], such as timing attacks and power analysis attacks, which can extract critical information from the ciphertext and lead to the compromise of the entire system.

To address this issue, leakage-resilient cryptography provides a new approach that aims to make cryptographic systems more robust. This approach not only considers the confidentiality aspect of traditional encryption but also takes into account the potential leakage of information during the encryption process. By establishing security on stronger assumptions, such as assuming that the attacker only knows partial ciphertext or assuming that the attacker can only obtain some side-channel information, leakage-resilient cryptography provides stronger security guarantees, making cryptographic systems more durable and capable of effectively resisting different types of attacks.

Certificateless encryption is a type of public-key encryption that eliminates the need for digital certificates, which are traditionally used to bind public keys to identities. In 2013, Xiong et al. [38] presented the first certificateless public key encryption scheme that was resilient to leakage attacks. However, the scheme they proposed only has an encryption mechanism to ensure the confidentiality of the message. In 2016 and 2019, Zhou et al. [34] and Yang et al. [35] respectively proposed leakage-resilient certificateless signcryption schemes which satisfy the confidentiality, integrity, and non-repudiation of the message. While both Zhou et al.'s and Yang et al.'s schemes are secure under the bounded leakage model, they are not secure under the continual leakage model.

III. PRELIMINARIES

In this section, we introduce five parts. First, the bilinear groups are used to construct a concrete scheme. The next three parts, we employ the generic bilinear group (GBG) model, complexity assumptions and the entropy concept to demonstrate the security of the proposed scheme. The final part involves organizing the symbols that will be utilized in the proposed scheme.

A. BILINEAR GROUPS

Assume that there are two multiplicative cyclic groups G_1 and G_2 with the same prime order q. Let $\hat{e}: G_1 \times G_1 \to G_2$ be a bilinear map, and g be a generator of G_1. The parameters of a bilinear group consist of q, G_1, G_2, $\hat{e}$ and g, and the bilinear map $\hat{e}$ satisfies the following three properties.

- Bilinearity: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$, for any $a, b \in Z_q^*$.
- Non-degeneracy: $\hat{e}(g, g) \neq 1$
- Computability: for any $A, B \in G_1$, the result of $\hat{e}(A, B)$ can be effectively obtained.

For additional details regarding bilinear groups, the reader may refer to [39].

B. GENERIC BILINEAR GROUP MODEL

The generic bilinear group (GBG) model [40] was used to prove the security of a leakage-resilient cryptographic scheme even in the event that attackers could obtain a portion of the private key. To determine the maximum length of bit strings of private keys allowed to be leaked, all elements in G_1 and G_2 must be expressed as bit strings. Therefore, we employ two injective functions, $IF_1 = Z_q^* \to \mathbb{B}G_1$ and $IF_2 = Z_q^* \to \mathbb{B}G_2$, where $\mathbb{B}G_1$ and $\mathbb{B}G_2$ denote, respectively, the sets collecting bit strings transformed from G_1 and G_2. In this case, the sets $\mathbb{B}G_1$ and $\mathbb{B}G_2$ are distinct from each other, and both have the same size of q, i.e., $|\mathbb{B}G_1| = |\mathbb{B}G_2| = q$. Under the GBG model, the three primary operations of a bilinear group consist of the multiplications in G_1 and G_2, as well as a bilinear map $\hat{e}$. We represent these primary operations as follows.

- $PO_{G_1}(IF_1(u), IF_1(v)) \to IF_1(u + v \ mod \ q)$.
- $PO_{G_2}(IF_2(u), IF_2(v)) \to IF_2(u + v \ mod \ q)$.
- $PO_{\hat{e}}(IF_1(u), IF_1(v)) \to IF_2(u \cdot v \ mod \ q)$.

Here, $u, v \in Z_q^*$, $g = IF_1(1)$ and $\hat{e}(g, g) = IF_2(1)$.

C. COMPLEXITY ASSUMPTIONS

We rely on the discrete logarithm problem (DL) and a hash function (HF) to establish the security of the proposed LR-CLSC scheme. Specifically, we utilize the following two assumptions in our proof.

Definition 1 (DL Assumption): The DL problem involves finding an unknown value $c \in Z_q^*$ for given parameters of a bilinear group, where c is originally hidden in either g^c or g_2^c, where $g_2 = \hat{e}(g, g)$. A probabilistic polynomial-time (PPT) adversary is hard to evaluate c, namely, the solution to the DL problem.

Definition 2 (HF Assumption): A hash function (HF) holds one-way and strong-collision resistant properties. That is, a PPT adversary is hard to find two values V_1, V_2 such that $HF(V_1) = HF(V_2)$.

D. ENTROPY

Entropy is a measure of uncertainty that can describe the probability of an event occurring, such as an attacker recovering a private key from a portion of the private key in a leakage-resilient cryptographic scheme. Two types of min-entropy are defined as follows.

1. The min-entropy of a finite random variable V is
$$H_\infty(V) = -\log_2(\max_v \Pr[V = v])$$
2. The average conditional min-entropy of a finite random variable V with a condition D is
$$\widetilde{H}_\infty(V|D) = -\log_2(D[\max_v \Pr[V = v|D]]).$$

The entropy of a finite random variable (a single private key) is used to evaluate its security. Dodis et al. [41] proposed the following lemma to measure the security of a single private key.

Lemma 1: Let V represent a random variable and λ represent the maximum length of leaked bits of the private key. Assume that $LF : V \to \{0, 1\}^\lambda$ is a leakage function. We have the inequality $\widetilde{H}_\infty(V|LF(V)) \geqq H_\infty(V) - \lambda$.

The entropy of finite multiple random variables (multiple private keys) is used to evaluate their security. Galindo and Vivek [42] proposed the following lemma to measure the security of multiple private keys.

Lemma 2: Let $V_1, V_2, \ldots, V_n$ represent finite multiple random variables and $V \in Z_q[V_1, V_2, \ldots, V_n]$ represent a polynomial of degree at most d. Assume that $PD_1, PD_2, \ldots, PD_n$ are probability distributions of $V_1 = v_1, V_2 = v_2, \ldots, V_n = v_n$ such that $H_\infty(PD_i) \geqq \log q - \lambda$ and $0 \leqq \lambda \leqq \log q$ for $i \in [1, n]$. For $i \in [1, n]$, if all $v_i \xleftarrow{PD_i} Z_q$ are independent and λ is less than or equal to ϵ (a positive fraction), then we have the inequality $\Pr[V(V_1 = v_1, V_2 = v_2, \ldots, V_n = v_n){=}0] \leqq (d/q)2^\lambda$.

E. SYMBOLS

In the LR-CLSC, a multitude of symbols will be utilized. To enhance the readability for readers, we have organized these symbols into Table 1.

IV. SYNTAX AND ADVERSARY MODEL FOR LR-CLSC

A leakage-resilient certificateless signcryption (LR-CLSC) scheme consists of two roles: a trusted key generating centre (KGC) and entities. The KGC employs a system master key SMK to compute the partial public key KPK_{ID} and partial secret key KSK_{ID} for an entity with identity ID. Then these keys are transmitted to the entity through a secure channel. Meanwhile, the entity generates the entity secret key ESK_{ID} and entity public key EPK_{ID}. Based on the concept of leakage-resilient property [42], we divide every secret key used in the LR-CLSC scheme (including SMK, KSK_{ID} and ESK_{ID}) into two parts. In addition, an update process is required when using the divided two parts above in each session. For example, in the i-th session, if the

TABLE 1. Symbols.

Symbol	Meaning
PSP	The public system parameters
SPK	The system public key
SMK	The system master key
SMK_0	The beginning system master key
SMK_i	The system master key in the i-th session
KPK_{ID}	The partial public key of an entity ID
KSK_{ID}	The partial secret key of an entity ID
$KSK_{ID,0}$	The beginning partial secret key of an entity ID
$KSK_{ID,j}$	The partial secret key of an entity ID in the j-th session
EPK_{ID}	The entity public key
ESK_{ID}	The entity secret key
$ESK_{ID,0}$	The beginning entity secret key
$ESK_{ID,j}$	The entity secret key of an entity ID in the j-th session
M	The message
CT	The ciphertext

KGC wants to generate the entity's partial secret key KSK_{ID}, the KGC needs to update the latest system master key pair $(SMK_{i-1,A}, SMK_{i-1,B})$ to $(SMK_{i,A}, SMK_{i,B})$. We call this procedure as the key update process, but their essence remains unchanged due to $SMK = SMK_{0,A} \cdot SMK_{0,B} = SMK_{1,A} \cdot SMK_{1,B} = \ldots = SMK_{i,A} \cdot SMK_{i,B}$. Both KSK_{ID} and ESK_{ID} must also execute this key update process in each session. During the j-th session of the *Signcryption* algorithm's execution, the sender with identity ID_s first updates the latest partial secret key pair $(KSK_{ID_s,j-1,A}, KSK_{ID_s,j-1,B})$ to $(KSK_{ID_s,j,A}, KSK_{ID_s,j,B})$ and the latest entity secret key pair $(ESK_{ID_s,j-1,A}, ESK_{ID_s,j-1,B})$ to $(ESK_{ID_s,j,A}, ESK_{ID_s,j,B})$ while a ciphertext CT can be generated by this algorithm. During the k-th session of the *Unsigncryption* algorithm's execution, the receiver with identity ID_r first updates the latest partial secret key pair $(KSK_{ID_r,k-1,A}, KSK_{ID_r,k-1,B})$ to $(KSK_{ID_r,k,A}, KSK_{ID_r,k,B})$ and the latest entity secret key pair $(ESK_{ID_r,k-1,A}, ESK_{ID_r,k-1,B})$ to $(ESK_{ID_r,k,A}, ESK_{ID_r,k,B})$ while a message M can be generated by this algorithm. In the following, we formally define the syntax and adversary model for LR-CLSC scheme.

A. SYNTAX FOR LR-CLSC

According to the syntaxes in LR-CLE [43] and LR-CLS [44], we define a new syntax for LR-CLSC as follows.

Definition 3: An LR-CLSC scheme involves the utilization of five algorithms, namely, *Setup*, *Entity partial keys generation*, *Entity keys generation*, *Signcryption* and *Unsigncryption*, as presented below.

- *Setup* : The KGC executes the algorithm with a security parameter ω as input. The output consists of the system master key SMK and the public system parameters PSP. The KGC then computes the beginning system master key $SMK_0 = (SMK_{0,A}, SMK_{0,B})$ by using SMK.
- *Entity partial keys generation* : During the i-th session of the algorithm's execution, for given public system parameters PSP, the latest system master key $SMK_{i-1} = (SMK_{i-1,A}, SMK_{i-1,B})$, and an entity with identity ID, the KGC produces the partial public key KPK_{ID} and partial secret key KSK_{ID} for the entity. The KGC transmits the partial secret key KSK_{ID} to the entity through a secure communication channel. Then, the entity creates her/his beginning partial secret key $KSK_{ID,0} = (KSK_{ID,0,A}, KSK_{ID,0,B})$ by using KSK_{ID}.
- *Entity keys generation* : An entity with identity ID executes the algorithm with the public system parameters PSP as input. The output consists of the entity secret key ESK_{ID} and entity public key EPK_{ID}. The entity then computes the beginning entity secret key $ESK_{ID,0} = (ESK_{ID,0,A}, ESK_{ID,0,B})$ by using ESK_{ID}.
- *Signcryption* : During the j-th session of the algorithm's execution, an entity regarded as the sender with identity ID_s executes the algorithm with, as input, the public system parameters PSP, the latest partial secret key $KSK_{ID_s,j-1} = (KSK_{ID_s,j-1,A}, KSK_{ID_s,j-1,B})$, the latest entity secret key $ESK_{ID_s,j-1} = (ESK_{ID_s,j-1,A}, ESK_{ID_s,j-1,B})$, a receiver's partial public key KPK_{ID_r}, public key EPK_{ID_r} and a message M. The output is a ciphertext CT.
- *Unsigncryption* : During the k-th session of the algorithm's execution, an entity regarded as the receiver with identity ID_r executes the algorithm with, as input, the public system parameters PSP, the latest partial secret key $KSK_{ID_r,k-1} = (KSK_{ID_r,k-1,A}, KSK_{ID_r,k-1,B})$, the latest entity secret key $ESK_{ID_r,k-1} = (ESK_{ID_r,k-1,A}, ESK_{ID_r,k-1,B})$, a sender's partial public key KPK_{ID_s}, public key EPK_{ID_s} and a ciphertext $CT = (CT_0, CT_1, CT_2, ID_s, ID_r)$. The output is a message M.

B. ADVERSARY MODEL FOR LR-CLSC

Following the notions of leakage-resilient property [42], we first define six leakage functions: $LF^{I}_{EPGK,i}$, $LF^{II}_{EPGK,i}$, $LF^{I}_{SC,j}$, $LF^{II}_{SC,j}$, $LF^{I}_{USC,k}$ and $LF^{II}_{USC,k}$. The two leakage functions $LF^{I}_{EPGK,i}$ and $LF^{II}_{EPGK,i}$ are utilized to simulate the adversary's leakage ability in the *Entity partial keys generation* algorithm, where the adversary is allowed to obtain partial bits of the system master key $SMK_i = (SMK_{i,A}, SMK_{i,B})$ during the i-th session of this algorithm's execution. The two leakage functions $LF^{I}_{SC,j}$ and $LF^{II}_{SC,j}$ are utilized to simulate an adversary's leakage ability in the *Signcryption* algorithm, where the adversary is allowed to obtain partial bits of the partial secret key $KSK_{ID_s,j} = (KSK_{ID_s,j,A}, KSK_{ID_s,j,B})$ and the entity secret key $ESK_{ID_s,j} = (ESK_{ID_s,j,A}, ESK_{ID_s,j,B})$ during the j-th session of this algorithm's execution. The two leakage functions $LF^{I}_{USC,k}$ and $LF^{II}_{USC,k}$ are utilized to simulate an adversary's leakage ability in the *Unsigncryption* algorithm, where the adversary is allowed to obtain partial bits of the partial secret key $KSK_{ID_r,k} = (KSK_{ID_r,k,A}, KSK_{ID_r,k,B})$ and entity secret key $ESK_{ID_s,k} = (ESK_{ID_s,k,A}, ESK_{ID_s,k,B})$ during the k-th session of this algorithm's execution. Let λ represent the maximum length of leaked bits of these secret keys. The output lengths of these leakage functions, represented by $|LF^{I}_{EPGK,i}|$, $|LF^{II}_{EPGK,i}|$, $|L^{I}_{SC,j}|$, $|LF^{II}_{SC,j}|$, $|LF^{I}_{USC,k}|$ and $|LF^{II}_{USC,k}|$, are less than or equal to λ. Next, we present the outputs of these six leakage functions as follows.

- $\Lambda LF^{I}_{EPGK,i} = LF^{I}_{EPGK,i}(SMK_{i,A})$.
- $\Lambda LF^{II}_{EPGK,i} = LF^{II}_{EPGK,i}(SMK_{i,B})$.
- $\Lambda LF^{I}_{SC,j} = LF^{I}_{SC,j}(KSK_{ID_s,j,A}, ESK_{ID_s,j,A})$.
- $\Lambda LF^{II}_{SC,j} = LF^{II}_{SC,j}(KSK_{ID_s,j,B}, ESK_{ID_s,j,B})$.
- $\Lambda LF^{I}_{USC,k} = LF^{I}_{USC,k}(KSK_{ID_r,k,A}, ESK_{ID_r,k,A})$.
- $\Lambda LF^{II}_{USC,k} = LF^{II}_{USC,k}(KSK_{ID_r,k,B}, ESK_{ID_r,k,B})$.

According to the adversary models in LR-CLE [43] and LR-CLS [44], we define new adversary models for LR-CLSC. We first introduce the two types of adversaries.

- The type I adversary $\mathcal{A}_I$ runs the *Entity keys generation* algorithm to obtain the entity secret key ESK_{ID} and the entity public key EPK_{ID}. However, $\mathcal{A}_I$ cannot obtain the system master key SMK or partial secret key KSK_{ID}. Due to the leakage-resilient property, $\mathcal{A}_I$ is able to obtain partial bits of SMK and KSK_{ID}.
- The type II adversary $\mathcal{A}_{II}$ possesses the system master key SMK. However, $\mathcal{A}_{II}$ cannot obtain the entity secret key ESK_{ID}. Due to the leakage-resilient property, $\mathcal{A}_{II}$ is able to obtain partial bits of ESK_{ID}.

One of the new adversary models is employed to represent the authentication of the signature (AoS) and the other is employed to represent the confidentiality of encryption (CoE) which are presented as follows.

Definition 4: Assume that an adversary $\mathcal{A}$ ($\mathcal{A}_I$ or $\mathcal{A}_{II}$) has the attacking abilities with both side-channel and adaptive chosen-message attacks. We say that an LR-CLSC scheme is secure for the existential unforgeability against this adversary if there is no non-negligible advantage for the adversary $\mathcal{A}$ to win a security game G_{AoS} as defined below.

- *Setup phase*: Let $\mathcal{C}$ be a challenger who executes the *Setup* algorithm with a security parameter ω. The output consists of the system master key SMK and the public system parameters PSP. The challenger $\mathcal{C}$ computes the beginning system master key $SMK_0 = (SMK_{0,A}, SMK_{0,B})$, and sends SMK_0 to the adversary if the adversary is $\mathcal{A}_{II}$. Notice that, an adversary $\mathcal{A}_I$ knows nothing about SMK_0. Also, both $\mathcal{A}_I$ and $\mathcal{A}_{II}$ have the public system parameters PSP.
- *Query phase*: The adversary can adaptively issue different types of queries to the challenger $\mathcal{C}$ as follows.
    - *Entity partial keys generation query*(ID): An identity ID is used as input for this query. With ID, the challenger $\mathcal{C}$ runs the *Entity partial keys generation* algorithm to generate the partial public key KPK_{ID} and partial secret key KSK_{ID}, and sends them to the adversary $\mathcal{A}$.
    - *Entity partial keys generation leak query*(i, $LF^{I}_{EPGK,i}$, $LF^{II}_{EPGK,i}$): A session index i, two leakage functions $LF^{I}_{EPGK,i}$ and $LF^{II}_{EPGK,i}$ are used as input for this query. The challenger $\mathcal{C}$ computes $\Lambda LF^{I}_{EPGK,i} = LF^{I}_{EPGK,i}(SMK_{i,A})$ and $\Lambda LF^{II}_{EPGK,i} = LF^{II}_{EPGK,i}(SMK_{i,B})$, and returns $\Lambda LF^{I}_{EPGK,i}$ and $\Lambda LF^{II}_{EPGK,i}$ to the adversary $\mathcal{A}$.
    - *Entity keys generation query*(ID): An identity ID is used as input for this query. With ID, the challenger $\mathcal{C}$ runs the *Entity keys generation* algorithm to generate the entity secret key ESK_{ID} and entity public key EPK_{ID}, and sends them to the adversary $\mathcal{A}$.
    - *Entity Public key replace query*(ID, KPK'_{ID}, EPK'_{ID}): An identity ID, two replace public keys KPK'_{ID} and EPK'_{ID} are used as input for this query. The challenger $\mathcal{C}$ records the replacement.
    - *Signcryption query*(M, ID_s, ID_r): A message M, two identities ID_s and ID_r are used as input for this query. The challenger $\mathcal{C}$ sets the partial secret key $KSK_{ID_s,j} = (KSK_{ID_s,j,A}, KSK_{ID_s,j,B})$, the entity secret key $ESK_{ID_s,j} = (ESK_{ID_s,j,A}, ESK_{ID_s,j,B})$, a receiver's partial public key KPK_{ID_r} and public key EPK_{ID_r}, and runs the *Signcryption* algorithm to generate a ciphertext CT.
    - *Signcryption leak query*(ID_s, j, $LF^{I}_{SC,j}$, $LF^{II}_{SC,j}$): A session index j, two leakage function $LF^{I}_{SC,j}$ and $LF^{II}_{SC,j}$ are used as input for this query. The challenger $\mathcal{C}$ computes $\Lambda LF^{I}_{SC,j} = LF^{I}_{SC,j}(KSK_{ID_s,j,A}, ESK_{ID_s,j,A})$ and $\Lambda LF^{II}_{SC,j} = LF^{II}_{SC,j}(KSK_{ID_s,j,B}, ESK_{ID_s,j,B})$, and returns $\Lambda LF^{I}_{SC,j}$ and $\Lambda LF^{II}_{SC,j}$ to the adversary $\mathcal{A}$.
    - *Unsigncryption*(CT, ID_s, ID_r): A message CT, two identities ID_s and ID_r are used as input for this query. The challenger $\mathcal{C}$ sets the partial secret key $KSK_{ID_r,k} = (KSK_{ID_r,k,A}, KSK_{ID_r,k,B})$, the entity secret key $ESK_{ID_r,k} = (ESK_{ID_r,k,A}, ESK_{ID_r,k,B})$, a sender's partial public key KPK_{ID_s} and public key EPK_{ID_s}, and runs the *Unsigncryption* algorithm to generate the message M.
    - *Unsigncryption leak query*(ID_r, k, $LF^{I}_{USC,k}$, $LF^{II}_{USC,k}$): A session index k, two leakage function $LF^{I}_{USC,k}$ and $LF^{II}_{USC,k}$ are used as input for this query. The challenger $\mathcal{C}$ computes $\Lambda LF^{I}_{USC,k} = LF^{I}_{USC,k}(KSK_{ID_r,k,A}, ESK_{ID_r,k,A})$ and $\Lambda LF^{II}_{USC,k} = LF^{II}_{USC,k}(KSK_{ID_r,k,B}, ESK_{ID_r,k,B})$, and returns $\Lambda LF^{I}_{USC,k}$ and $\Lambda LF^{II}_{USC,k}$ to the adversary $\mathcal{A}$.
- *Forgery*: A ciphertext $CT' = (CT'_0, CT'_1, CT'_2, ID'_s, ID'_r)$ for a message M' is forged by the adversary $\mathcal{A}$. We say that $\mathcal{A}$ wins the security game G_{AoS} if the following four conditions are met.
    - The message M' can be generated by the *Unsigncryption* algorithm.
    - The message M' as well as two identities ID'_s and ID'_r never appear in the *Signcryption query*.
    - If the adversary is $\mathcal{A}_I$, the identity ID'_s never appears in the *Entity partial keys generation query*.
    - If the adversary is $\mathcal{A}_{II}$, the identity ID'_s never appears neither in the *Entity keys generation query* nor *Entity Public key replace query*.

Definition 5: Assume that an adversary $\mathcal{A}$ (including $\mathcal{A}_I$ and $\mathcal{A}_{II}$) has the attacking abilities with side-channel and adaptive chosen-ciphertext attacks. We say that an LR-CLSC scheme is secure for the ciphertexts indistinguishability against this adversary if there is no non-negligible advantage for the adversary $\mathcal{A}$ to win a security game G_{CoE} as defined below.

- *Setup phase*: This phase is the same as the *Setup phase* in Definition 4.
- *Query phase*: This phase is the same as the *Query phase* in Definition 4.
- *Challenge phase*: The adversary $\mathcal{A}$ picks an identity ID'_r, two message M_0 and M_1 as a challenge objective. The challenger $\mathcal{C}$ randomly chooses a $coin \in \{0, 1\}$, and generates a challenge ciphertext CT' by running the *Signcryption* algorithm with (M_{coin}, ID_s, ID'_r). The challenge ciphertext CT' is sent to the adversary $\mathcal{A}$, and $\mathcal{A}$ wins the security game G_{AoS} if the following conditions are met.
  - If the adversary is $\mathcal{A}_I$, the identity ID'_s never appears in the *Entity partial keys generation query*.
  - If the adversary is $\mathcal{A}_{II}$, the identity ID'_s never appears neither in the *Entity keys generation query* nor *Entity Public key replace query*.
- *Guess phase*: A guess $coin' \in \{0, 1\}$ is output by the adversary $\mathcal{A}$. We say that $\mathcal{A}$ wins this the game if $coin' = coin$. The winning advantage is defined as $Adv(\mathcal{A}) = |Pr[coin' = coin] - 1/2|$.

V. A CONCRETE LR-CLSC SCHEME

In this section, we show a leakage-resilient certificateless signcryption (LR-CLSC) scheme. We can refer to Fig. 1 for a visual representation of the LR-CLSC scheme, which involves the following five algorithms and two roles: a trusted KGC and entities.

- *Setup*: The KGC executes the algorithm with a security parameter ω as input. The output consists of the system master key SMK and the public system parameters PSP. The detailed processes are shown as follows.
  - Generate the bilinear parameters q, $\hat{e}$, g, G_1, G_2 as described in Section III.
  - Pick a random value $s \in Z_q^*$, and compute the system master key $SMK = g^s$.
  - Set the system public key $SPK = \hat{e}(SMK, g) = \hat{e}(g^s, g)$.
  - Randomly choose a reset value $a \in Z_q^*$, and compute the beginning system master key $SMK_0 = (SMK_{0,A}, SMK_{0,B}) = (g^a, g^{-a} \cdot SMK)$.
  - Randomly pick four values $t, k, u, v \in Z_q^*$, and set $T = g^t$, $K = g^k$, $U = g^u$ and $V = g^v$.
  - Choose a hash functions $HF : \{0, 1\}^* \times G_1 \to \{0, 1\}^l$, where l is a fixed length.
  - Employ the encryption function $Enc()$ and decryption function $Dec()$ of a symmetric cryptosystem.
  - Set the public system parameters $PSP = \{q, \hat{e}, g, G_1, G_2, SPK, T, K, U, V, HF, Enc(), Dec()\}$.
- *Entity partial keys generation*: During the i-th session of the algorithm's execution, given the public system parameters PSP, the latest system master key $SMK_{i-1} = (SMK_{i-1,A}, SMK_{i-1,B})$, and an entity with identity ID, the KGC generates the partial public key KPK_{ID} and partial secret key KSK_{ID} for the entity through the following steps.
  - Generate a current system master key $SMK_i = (SMK_{i,A}, SMK_{i,B}) = (g^b \cdot SMK_{i-1,A}, g^{-b} \cdot SMK_{i-1,B})$ by randomly selecting a reset value $b \in Z_q^*$.
  - Randomly choose a value $r \in Z_q^*$, and compute the partial public key $KPK_{ID} = g^r$.
  - Use the value r to set temporary key $TK_{ID} = SMK_{i,A} \cdot (T \cdot K^{ID})^r$.
  - Compute the partial secret key $KSK_{ID} = SMK_{i,B} \cdot TK_{ID}$.

  The KGC sends the partial secret key KSK_{ID} to the entity through a secure communication channel. Then, the entity picks a random reset value $c \in Z_q^*$ and creates her/his beginning partial secret key $KSK_{ID,0} = (KSK_{ID,0,A}, KSK_{ID,0,B}) = (g^c, g^{-c} \cdot KSK_{ID})$.
- *Entity keys generation*: An entity with identity ID executes the algorithm with the public system parameters PSP as input. The output consists of the entity secret key ESK_{ID} and entity public key EPK_{ID}. The detailed processes are shown as follows.
  - Pick a random value $e \in Z_q^*$, and compute the entity secret key $ESK_{ID} = g^e$.
  - Set the entity public key $EPK_{ID} = \hat{e}(ESK_{ID}, g) = \hat{e}(g^e, g)$.

  Then, the entity picks a random reset value $d \in Z_q^*$ and creates her/his beginning entity secret key $ESK_{ID,0} = (ESK_{ID,0,A}, ESK_{ID,0,B}) = (g^d, g^{-d} \cdot ESK_{ID})$.
- *Signcryption*: During the j-th session of the algorithm's execution, an entity regarded as the sender with identity ID_s executes the algorithm with, as input, the public system parameters PSP, the latest partial secret key $KSK_{ID_s,j-1} = (KSK_{ID_s,j-1,A}, KSK_{ID_s,j-1,B})$, the latest entity secret key $ESK_{ID_s,j-1} = (ESK_{ID_s,j-1,A}, ESK_{ID_s,j-1,B})$, a receiver's partial public key KPK_{ID_r}, public key EPK_{ID_r} and a message M. The output is a ciphertext CT. The detailed processes are shown as follows.
  - Generate a current partial secret key $KSK_{ID_s,j} = (KSK_{ID_s,j,A}, KSK_{ID_s,j,B}) = (g^h \cdot KSK_{ID_s,j-1,A}, g^{-h} \cdot KSK_{ID_s,j-1,B})$ and a current entity secret key $ESK_{ID_s,j} = (ESK_{ID_s,j,A}, ESK_{ID_s,j,B}) = (g^h \cdot ESK_{ID_s,j-1,A}, g^{-h} \cdot ESK_{ID_s,j-1,B})$ by randomly selecting a reset value $h \in Z_q^*$.
  - Randomly choose a value $\alpha \in Z_q^*$, and compute $CT_1 = g^\alpha$, $SK_1 = (EPK_{ID_r})^\alpha$ and $SK_2 = (SPK \cdot \hat{e}(KPK_{ID_r}, T \cdot K^{ID}))^\alpha$.

- Set a symmetric key $SK = SK_1 \oplus SK_2$, and compute $CT_2 = Enc_{SK}(M)$ and $f = HF(M, CT_1, CT_2, ID_s, ID_r)$.
- Set a temporary signature $TS = KSK_{IDs,j,A} \cdot ESK_{ID_s,j,A} \cdot (U \cdot V^f)^\alpha$.
- Generate a signature $CT_0 = KSK_{ID_s,j,B} \cdot ESK_{ID_s,j,B} \cdot TS$.
- Set a ciphertext $CT = (CT_0, CT_1, CT_2, ID_s, ID_r)$.

- *Unsigncryption*: During the k-th session of the algorithm's execution, an entity regarded as the receiver with identity ID_r executes the algorithm with, as input, the public system parameters PSP, the latest partial secret key $KSK_{ID_r,k-1} = (KSK_{ID_r,k-1,A}, KSK_{ID_r,k-1,B})$, the latest entity secret key $ESK_{ID_r,k-1} = (ESK_{ID_r,k-1,A}, ESK_{ID_r,k-1,B})$, a sender's partial public key KPK_{ID_s}, public key EPK_{ID_s} and a ciphertext $CT = (CT_0, CT_1, CT_2, ID_s, ID_r)$. The output is a message M. The detailed processes are shown as follows.
  - Generate a current partial secret key $KSK_{ID_r,k} = (KSK_{ID_r,k,A}, KSK_{ID_r,k,B}) = (g^w \cdot KSK_{ID_r,k-1,A}, g^{-w} \cdot KSK_{ID_r,k-1,B})$ and a current entity secret key $ESK_{ID_r,k} = (ESK_{ID_r,k,A}, ESK_{ID_r,k,B}) = (g^w \cdot ESK_{ID_r,k-1,A}, g^{-w} \cdot ESK_{ID_r,k-1,B})$ by randomly selecting a reset value $w \in Z_q^*$.
  - Compute two temporary keys $TSK_1 = \hat{e}(CT_1, ESK_{ID_r,k,A})$ and $TSK_2 = \hat{e}(CT_1, KSK_{ID_r,k,A})$.
  - Compute $SK_1' = TSK_1 \cdot \hat{e}(CT_1, ESK_{ID_r,k,B})$ and $SK_2' = TSK_2 \cdot \hat{e}(CT_1, KSK_{ID_r,k,B})$.
  - Set a symmetric key $SK' = SK_1' \oplus SK_2'$, and generate a message $M = Dec'_{SK}(CT_2)$.
  - Compute $f' = HF(M, CT_1, CT_2, ID_s, ID_r)$ and verify $\hat{e}(g, CT_0) = SPK \cdot EPK_{ID_s} \cdot \hat{e}(KPK_{ID_s}, T \cdot K_{ID}) \cdot \hat{e}(CT_1, U \cdot V^{f'})$.

Let's prove the correctness of two entities $SK' = SK_1' \oplus SK_2' = SK_1 \oplus SK_2 = SK$ and $\hat{e}(g, CT_0) = SPK \cdot EPK_{ID_s} \cdot \hat{e}(KPK_{ID_s}, T \cdot K^{ID}) \cdot \hat{e}(CT_1, U \cdot V^{f'})$.

✓
$$\begin{aligned} SK' &= SK_1' \oplus SK_2' = TSK_1 \cdot \hat{e}(CT_1, ESK_{ID_r,k,B}) \\ &\quad \oplus TSK_2 \cdot \hat{e}(CT_1, KSK_{ID_r,k,B}) \\ &= \hat{e}(CT_1, ESK_{ID_r,k,A}) \cdot \hat{e}(CT_1, ESK_{ID_r,k,B}) \\ &\quad \oplus \hat{e}(CT_1, KSK_{ID_r,k,A}) \cdot \hat{e}(CT_1, KSK_{ID_r,k,B}) \\ &= \hat{e}(CT_1, ESK_{ID_r}) \oplus \hat{e}(CT_1, KSK_{ID_r}) \\ &= \hat{e}(g^\alpha, ESK_{ID_r}) \oplus \hat{e}(g^\alpha, KSK_{ID_r}) \\ &= \hat{e}(g, ESK_{ID_r})^\alpha \oplus \hat{e}(g^\alpha, SMK \cdot (T \cdot K^{ID})^r) \\ &= \hat{e}(g, ESK_{ID_r})^\alpha \oplus \hat{e}(g^\alpha, SMK) \cdot \hat{e}(g^\alpha, (T \cdot K^{ID})^r) \\ &= \hat{e}(g, ESK_{ID_r})^\alpha \oplus \hat{e}(g, SMK)^\alpha \cdot \hat{e}(g^r, (T \cdot K^{ID})^\alpha) \\ &= (EPK_{ID_r})^\alpha \oplus (SPK \cdot \hat{e}(KPK_{ID_r}, T \cdot K^{ID}))^\alpha \\ &= SK_1 \oplus SK_2 \end{aligned}$$

✓
$$\begin{aligned} \hat{e}(g, CT_0) &= \hat{e}(g, KSK_{ID_s,j,B} \cdot ESK_{ID_s,j,B} \cdot TS) \\ &= \hat{e}(g, KSK_{ID_s,j,B} \cdot ESK_{ID_s,j,B} \cdot KSK_{ID_s,j,A} \\ &\quad \cdot ESK_{ID_s,j,A} \cdot (U \cdot V^f)^\alpha) \\ &= \hat{e}(g, KSKID_s \cdot ESK_{ID_s} \cdot (U \cdot V^f)^\alpha) \\ &= \hat{e}(g, KSK_{ID_s}) \cdot \hat{e}(g, ESK_{ID_s}) \cdot \hat{e}(g, (U \cdot V^f)^\alpha) \\ &= \hat{e}(g, SMK \cdot (T \cdot K^{ID})^r) \cdot \hat{e}(g, ESK_{ID_s}) \\ &\quad \cdot \hat{e}(g, (U \cdot V^f)^\alpha) \\ &= \hat{e}(g, SMK) \cdot \hat{e}(g, (T \cdot K^{ID})^r) \\ &\quad \cdot \hat{e}(g, ESK_{ID_s}) \cdot \hat{e}(g, (U \cdot V^f)^\alpha) \\ &= SPK \cdot \hat{e}(g^r, T \cdot K^{ID}) \cdot EPK_{ID_s} \cdot \hat{e}(g^\alpha, U \cdot V^f) \\ &= SPK \cdot EPK_{ID_s} \cdot \hat{e}(KPK_{ID_s}, T \cdot K^{ID}) \\ &\quad \cdot \hat{e}(CT_1, U \cdot V^{f'}) \end{aligned}$$

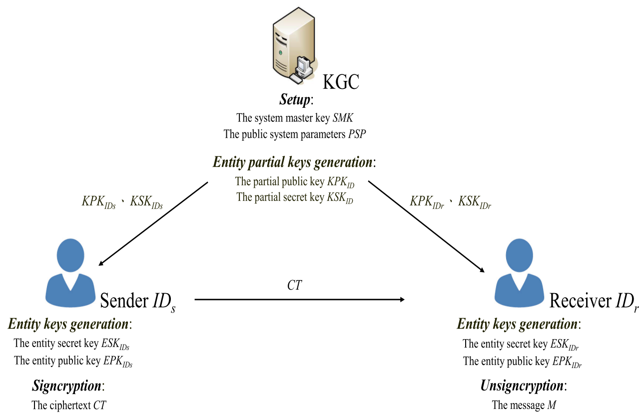


FIGURE 1. Visual representation of the LR-CLSC scheme.

VI. SECURITY ANALYSIS

As mentioned earlier, the adversary models are employed to represent the authentication of the signature (AoS) and the confidentiality of encryption (CoE). In the following, we give four theorems to complete the security analysis of the proposed LR-CLSC scheme.

Theorem 1: Assume that the HF and DL assumptions hold. Under the GBG model, the proposed LR-CLSC scheme is secure for the existential unforgeability against the adversary $\mathcal{A}_I$ in the security game G_{AoS}.

Proof: Let $\mathcal{C}$ be the challenger and interact with the adversary $\mathcal{A}_I$ in the following security game G_{AoS}.

- *Setup phase*: The challenger $\mathcal{C}$ executes the *Setup* algorithm to generate the system master key SMK and the public system parameters $PSP = \{q, \hat{e}, g, G_1, G_2, SPK, T, K, U, V, HF, Enc(), Dec()\}$. In addition, $\mathcal{C}$ creates the following six lists $List_{G_1}$, $List_{G_2}$, $List_{KSK}$, $List_{ESK}$, $List_{SC}$ and $List_{HF}$.
  - $List_{G_1}$ records items related to elements of G_1. Each item in $List_{G_1}$ is presented as $(\mathbb{P}G_{1,\zeta,\eta,\theta}, \mathbb{B}G_{1,\zeta,\eta,\theta})$, where $\mathbb{P}G_{1,\zeta,\eta,\theta}$ and $\mathbb{B}G_{1,\zeta,\eta,\theta}$ are, respectively, a multivariate polynomial and bit string of the element in G_1. And, three symbols ζ, η and θ denote the query type, the query number and the item number, respectively. Initially, six items $(\mathbb{P}g, \mathbb{B}G_{1,s,0,1})$, $(\mathbb{P}SMK, \mathbb{B}G_{1,s,0,2})$, $(\mathbb{P}T, \mathbb{B}G_{1,s,0,3})$, $(\mathbb{P}K, \mathbb{B}G_{1,s,0,4})$, $(\mathbb{P}U, \mathbb{B}G_{1,s,0,5})$ and $(\mathbb{P}V, \mathbb{B}G_{1,s,0,6})$ are recorded in $ListG_1$.
  - $List_{G_2}$ records items related to elements of G_2. Each item in $List_{G_2}$ is presented as $(\mathbb{P}G_{2,\zeta,\eta,\theta}, \mathbb{B}G_{2,\zeta,\eta,\theta})$, where $\mathbb{P}G_{2,\zeta,\eta,\theta}$ and $\mathbb{B}G_{2,\zeta,\eta,\theta}$ are, respectively, a multivariate polynomial and bit string of the element in G_2. And, three symbols ζ, η and θ

are the same as those in $ListG_1$. Initially, the item $(\mathbb{P}SPK, \mathbb{B}G_{2,s,0,1})$ is recorded in $List_{G_2}$.

It is noticed that in $List_{G_1}$ and $List_{G_2}$, each item is represented as both a multivariate polynomial and a bit string. Hence, we provide two conversion rules, CR-1 and CR-2, to explain the transformation between a multivariate polynomial and its bit string.

✓ CR-1: Convert $\mathbb{P}G_{1,\zeta,\eta,\theta}$ / $\mathbb{P}G_{2,\zeta,\eta,\theta}$ to $\mathbb{B}G_{1,\zeta,\eta,\theta}$ / $\mathbb{B}G_{2,\zeta,\eta,\theta}$ and return $\mathbb{B}G_{1,\zeta,\eta,\theta}$ / $\mathbb{B}G_{2,\zeta,\eta,\theta}$ if $\mathbb{P}G_{1,\zeta,\eta,\theta}$ / $\mathbb{P}G_{2,\zeta,\eta,\theta}$ exists in $List_{G_1}$ / $List_{G_2}$. Otherwise, a random bit string $\mathbb{B}G_{1,\zeta,\eta,\theta}$ / $\mathbb{B}G_{2,\zeta,\eta,\theta}$ related to $\mathbb{P}G_{1,\zeta,\eta,\theta}$ / $\mathbb{P}G_{2,\zeta,\eta,\theta}$ is chosen and returned while the bit string $\mathbb{B}G_{1,\zeta,\eta,\theta}$ / $\mathbb{B}G_{2,\zeta,\eta,\theta}$ is added in $List_{G_1}$ / $List_{G_2}$.

✓ CR-2: Convert $\mathbb{B}G_{1,\zeta,\eta,\theta}$ / $\mathbb{B}G_{2,\zeta,\eta,\theta}$ to $\mathbb{P}G_{1,\zeta,\eta,\theta}$ / $\mathbb{P}G_{2,\zeta,\eta,\theta}$ and return $\mathbb{P}G_{1,\zeta,\eta,\theta}$ / $\mathbb{P}G_{2,\zeta,\eta,\theta}$ if $\mathbb{B}G_{1,\zeta,\eta,\theta}$ / $\mathbb{B}G_{2,\zeta,\eta,\theta}$ exists in $List_{G_1}$ / $List_{G_2}$. Otherwise, a random multivariate polynomial $\mathbb{P}G_{1,\zeta,\eta,\theta}$ / $\mathbb{P}G_{2,\zeta,\eta,\theta}$ related to $\mathbb{B}G_{1,\zeta,\eta,\theta}$ / $\mathbb{B}G_{2,\zeta,\eta,\theta}$ is chosen and returned while the multivariate polynomial $\mathbb{P}G_{1,\zeta,\eta,\theta}$ / $\mathbb{P}G_{2,\zeta,\eta,\theta}$ is added in $List_{G_1}$ / $List_{G_2}$.

- $List_{KSK}$ records an entity's identity ID, partial public key KPK_{ID} and partial secret key KSK_{ID}. Each item in $List_{KSK}$ is presented as $(ID, \mathbb{P}KPK_{ID}, \mathbb{P}KSK_{ID})$.
- $List_{ESK}$ records an entity's identity ID, entity public key EPK_{ID} and entity secret key ESK_{ID}. Each item in $List_{ESK}$ is presented as $(ID, \mathbb{P}EPK_{ID}, \mathbb{P}ESK_{ID})$.
- $List_{SC}$ records the information of executing the *Signcryption* algorithm. Each item in $List_{SC}$ is presented as $(M, \mathbb{P}CT_0, \mathbb{P}CT_1, CT_2, \mathbb{P}SK_1, \mathbb{P}SK_2, \mathbb{P}f, ID_s, ID_r)$.
- $List_{HF}$ records the information of executing the hash function $HF()$. Each item in $List_{HF}$ is presented as $(M||CT_1||\mathbb{B}CT_2||ID_s||ID_r, \mathbb{P}f)$.

- *Query phase*: The adversary $\mathcal{A}_I$ can adaptively issue the following different types of queries at most ϕ times totally to the challenger $\mathcal{C}$.
  - PO_{G_1} *query* $(\mathbb{B}G_{1,q,r,a}, \mathbb{B}G_{1,q,r,b}, ORER)$: $\mathbb{B}G_{1,q,r,a}, \mathbb{B}G_{1,q,r,b}$ and $ORER$ are used as input for this query. The challenger $\mathcal{C}$ performs the following steps and returns $\mathbb{B}G_{1,q,r,c}$.

    ✓ Convert $(\mathbb{B}G_{1,q,r,a}, \mathbb{B}G_{1,q,r,b})$ to $(\mathbb{P}G_{1,q,r,a}, \mathbb{P}G_{1,q,r,b})$ by the rule CR-2.

    ✓ Compute $\mathbb{P}G_{1,q,r,c} = \mathbb{P}G_{1,q,r,a} + \mathbb{P}G_{1,q,r,b}$ if $ORER$ = "multiplication" and $\mathbb{P}G_{1,q,r,c} = \mathbb{P}G_{1,q,r,a}$ - $\mathbb{P}G_{1,q,r,b}$ if $ORER$ = "division".

    ✓ Convert $\mathbb{P}G_{1,q,r,c}$ to $\mathbb{B}G_{1,q,r,c}$ by the rule CR-1.
  - PO_{G_2} *query* $(\mathbb{B}G_{2,q,r,a}, \mathbb{B}G_{2,q,r,b}, ORER)$: $\mathbb{B}G_{2,q,r,a}, \mathbb{B}G_{2,q,r,b}$ and $ORER$ are used as input for this query. The challenger $\mathcal{C}$ performs the following steps and returns $\mathbb{B}G_{2,q,r,c}$.

    ✓ Convert $(\mathbb{B}G_{2,q,r,a}, \mathbb{B}G_{2,q,r,b})$ to $(\mathbb{P}G_{2,q,r,a}, \mathbb{P}G_{2,q,r,b})$ by the rule CR-2.

    ✓ Compute $\mathbb{P}G_{2,q,r,c} = \mathbb{P}G_{2,q,r,a} + \mathbb{P}G_{2,q,r,b}$ if $ORER$ = "multiplication" and $\mathbb{P}G_{2,q,r,c} = \mathbb{P}G_{2,q,r,a}$ - $\mathbb{P}G_{2,q,r,b}$ if $ORER$ = "division".

    ✓ Convert $\mathbb{P}G_{2,q,r,c}$ to $\mathbb{B}G_{2,q,r,c}$ by the rule CR-1.
  - $PO_{\hat{e}}$ *query* $(\mathbb{B}G_{1,q,r,a}, \mathbb{B}G_{1,q,r,b})$: $\mathbb{B}G_{1,q,r,a}$ and $\mathbb{B}G_{1,q,r,b}$ are used as input for this query. The challenger $\mathcal{C}$ performs the following steps and returns $\mathbb{B}G_{2,q,r,c}$.

    ✓ Convert $(\mathbb{B}G_{1,q,r,a}, \mathbb{B}G_{1,q,r,b})$ to $(\mathbb{P}G_{1,q,r,a}, \mathbb{P}G_{1,q,r,b})$ by the rule CR-2.

    ✓ Compute $\mathbb{P}G_{2,q,r,c} = \mathbb{P}G_{1,q,r,a} \cdot \mathbb{P}G_{1,q,r,b}$.

    ✓ Convert $\mathbb{P}G_{2,q,r,c}$ to $\mathbb{B}G_{2,q,r,c}$ by the rule CR-1.
  - *Entity partial keys generation query*(ID): An identity ID is used as input for this query. The challenger $\mathcal{C}$ performs the following steps and returns $(\mathbb{B}KPK_{ID}, \mathbb{B}KSK_{ID})$.

    ✓ Set a random variate $\mathbb{P}KPK_{ID}$ as the partial public key.

    ✓ Set $\mathbb{P}KSK_{ID} = \mathbb{P}SMK + (\mathbb{P}T + ID \cdot \mathbb{P}K) \cdot \mathbb{P}KPK_{ID}$ as the partial secret key.

    ✓ Convert $(\mathbb{P}KPK_{ID}, \mathbb{P}KSK_{ID})$ to $(\mathbb{B}KPK_{ID}, \mathbb{B}KSK_{ID})$ by the rule CR-1.
  - *Entity partial keys generation leak query* $(i, LF^{I}_{EPGK,i}, LF^{II}_{EPGK,i})$: A session index i, two leakage functions $LF^{I}_{EPGK,i}$ and $LF^{II}_{EPGK,i}$ are used as input for this query. The challenger $\mathcal{C}$ returns $\Lambda LF^{I}_{EPGK,i} = LF^{I}_{EPGK,i}(\mathbb{P}SMK_{i,A})$ and $\Lambda LF^{II}_{EPGK,i} = LF^{II}_{EPGK,i}(\mathbb{P}SMK_{i,B})$.
  - *Entity keys generation query*(ID): An identity ID is used as input for this query. The challenger $\mathcal{C}$ converts $(\mathbb{P}EPK_{ID}, \mathbb{P}ESK_{ID})$ to $(\mathbb{B}EPK_{ID}, \mathbb{B}ESK_{ID})$ by the rule CR-1, where $(\mathbb{P}EPK_{ID}, \mathbb{P}ESK_{ID})$ can be found in $List_{ESK}$. Then, $\mathcal{C}$ returns the entity public key $\mathbb{B}EPK_{ID}$ and entity secret key $\mathbb{B}ESK_{ID}$.
  - *Entity Public key replace query*$(ID, \mathbb{B}KPK'_{ID}, \mathbb{B}EPK'_{ID})$: An identity ID, two replace public keys $\mathbb{B}KPK'_{ID}$ and $\mathbb{B}EPK'_{ID}$ are used as input for this query. The challenger $\mathcal{C}$ first converts $(\mathbb{B}KPK'_{ID}, \mathbb{B}EPK'_{ID})$ to $(\mathbb{P}KPK'_{ID}, \mathbb{P}EPK'_{ID})$ by the rule CR-2. Then, $\mathcal{C}$ records $(ID, \mathbb{P}KPK'_{ID}, -)$ and $(ID, \mathbb{P}EPK'_{ID}, -)$ in $List_{KSK}$ and $List_{ESK}$, respectively.
  - *Signcryption query*(M, ID_s, ID_r): A message M, two identities ID_s and ID_r are used as input for this query. The challenger $\mathcal{C}$ uses the partial secret key $\mathbb{P}KSK_{ID_s,j} = (\mathbb{P}KSK_{ID_s,j,A}, \mathbb{P}KSK_{ID_s,j,B})$, the entity secret key $\mathbb{P}ESK_{ID_s,j} = (\mathbb{P}ESK_{ID_s,j,A}, \mathbb{P}ESK_{ID_s,j,B})$, a receiver's partial public key $\mathbb{P}KPK_{ID_r}$ and public key $\mathbb{P}EPK_{ID_r}$ to generate a ciphertext CT as the output. The detailed processes are shown as follows.

    ✓ With respect to ID_r, search $(ID_r, \mathbb{P}KPK_{ID_r}, \mathbb{P}KSK_{ID_r})$ in $List_{KSK}$ and $(ID_r, \mathbb{P}EPK_{ID_r}, \mathbb{P}ESK_{ID_r})$ in $List_{ESK}$.

- ✓ Set $\mathbb{P}SK_1 = \mathbb{P}EPK_{ID_r} \cdot \mathbb{P}\alpha$ and $\mathbb{P}SK_2 = (\mathbb{P}SPK + \mathbb{P}KPK_{ID_r} \cdot (\mathbb{P}T + ID \cdot \mathbb{P}K)) \cdot \mathbb{P}\alpha$, where $\mathbb{P}\alpha$ is a random variate.
- ✓ Convert $\mathbb{P}\alpha$, $\mathbb{P}SK_1$ and $\mathbb{P}SK_2$ to $\mathbb{B}\alpha$, $\mathbb{B}SK_1$ and $\mathbb{B}SK_2$ by the rule CR-1.
- ✓ Set $\mathbb{B}SK = \mathbb{B}SK_1 \oplus \mathbb{B}SK_2$ and $CT_2 = Enc_{\mathbb{B}SK}(M)$.
- ✓ Set $\mathbb{B}f = HF(M, \mathbb{B}\alpha, CT_2, ID_s, ID_r)$.
- ✓ Pick a new variate $\mathbb{P}f$ in G_1, and put $(\mathbb{P}f, \mathbb{B}f)$ in $List_{G_1}$.
- ✓ Set $\mathbb{P}CT_0 = \mathbb{P}KSK_{ID_s} + \mathbb{P}ESK_{ID_s} + (\mathbb{P}U + \mathbb{P}f \cdot \mathbb{P}V) \cdot \mathbb{P}\alpha$.
- ✓ Convert $\mathbb{P}CT_0$ to $\mathbb{B}CT_0$ by the rule CR-1.
- ✓ Put $(M, \mathbb{P}CT_0, \mathbb{P}\alpha, CT_2, \mathbb{P}SK_1, \mathbb{P}SK_2, \mathbb{P}f, ID_s, ID_r)$ in $List_{SC}$.
- ✓ Return $CT = (\mathbb{B}CT_0, \mathbb{B}\alpha, CT_2, ID_s, ID_r)$.

- *Signcryption leak query*$(ID_s, j, LF^{I}_{SC,j}, LF^{II}_{SC,j})$: A session index j, two leakage functions $LF^{I}_{SC,j}$ and $LF^{II}_{SC,j}$ are used as input for this query. The challenger $\mathcal{C}$ computes $\Lambda LF^{I}_{SC,j} = LF^{I}_{SC,j}(\mathbb{P}KSK_{ID_s,j,A}, \mathbb{P}ESK_{ID_s,j,A})$ and $\Lambda LF^{II}_{SC,j} = LF^{II}_{SC,j}(\mathbb{P}KSK_{ID_s,j,B}, \mathbb{P}ESK_{ID_s,j,B})$, and returns $\Lambda LF^{I}_{SC,j}$ and $\Lambda LF^{II}_{SC,j}$ to the adversary $\mathcal{A}_I$.
- *Unsigncryption*(CT, ID_s, ID_r): A message CT, two identities ID_s and ID_r are used as input for this query. The challenger $\mathcal{C}$ uses the partial secret key $\mathbb{P}KSK_{ID_r,k} = (\mathbb{P}KSK_{ID_r,k,A}, \mathbb{P}KSK_{ID_r,k,B})$, the entity secret key $\mathbb{P}ESK_{ID_r,k} = (\mathbb{P}ESK_{ID_r,k,A}, \mathbb{P}ESK_{ID_r,k,B})$, a sender's partial public key $\mathbb{P}KPK_{ID_s}$ and public key $\mathbb{P}EPK_{ID_s}$ to generate the message M as the output. The detailed processes are shown as follows.
  - ✓ With respect to ID_s, search $(ID_s, \mathbb{P}KPK_{ID_s}, \mathbb{P}KSK_{ID_s})$ in $List_{KSK}$ and $(ID_s, \mathbb{P}EPK_{ID_s}, \mathbb{P}ESK_{ID_s})$ in $List_{ESK}$.
  - ✓ Convert $\mathbb{P}KPK_{ID_s}$ and $\mathbb{P}EPK_{ID_s}$ to $\mathbb{B}KPK_{ID_s}$ and $\mathbb{B}EPK_{ID_s}$ by the rule CR-1.
  - ✓ Convert $\mathbb{B}CT_0$ and $\mathbb{B}\alpha$ to $\mathbb{P}CT_0$ and $\mathbb{P}\alpha$ by the rule CR-2.
  - ✓ Set $\mathbb{B}SK_1 = \mathbb{P}\alpha \cdot \mathbb{P}ESK_{ID_r}$ and $\mathbb{B}SK_2 = \mathbb{P}\alpha \cdot \mathbb{P}KSK_{ID_r}$.
  - ✓ Set $\mathbb{B}f = HF(M, \mathbb{B}\alpha, CT_2, ID_s, ID_r)$ and convert $\mathbb{B}f$ to $\mathbb{P}f$.
  - ✓ Use $\mathbb{P}CT_0, \mathbb{P}\alpha, CT_2, \mathbb{P}SK_1, \mathbb{P}SK_2, \mathbb{P}f, ID_s$ and ID_r to find $(M, \mathbb{P}CT_0, \mathbb{P}CT_1, CT_2, \mathbb{P}SK_1, \mathbb{P}SK_2, \mathbb{P}_f, ID_s, ID_r)$ in $List_{SC}$.
  - ✓ Output the message M if it is found. Otherwise, return "invalid".
- *Unsigncryption leak query*$(ID_r, k, LF^{I}_{USC,k}, LF^{II}_{USC,k})$: A session index k, two leakage functions $LF^{I}_{USC,k}$ and $LF^{II}_{USC,k}$ are used as input for this query. The challenger $\mathcal{C}$ computes $\Lambda LF^{I}_{USC,k} = LF^{I}_{USC,k}(\mathbb{P}KSK_{ID_r,k,A}, \mathbb{P}ESK_{ID_r,k,A})$ and $\Lambda LF^{II}_{USC,k} = LF^{II}_{USC,k}(\mathbb{P}KSK_{ID_r,k,B}, \mathbb{P}ESK_{ID_r,k,B})$, and returns $\Lambda LF^{I}_{USC,k}$ and $\Lambda LF^{II}_{USC,k}$ to the adversary $\mathcal{A}_I$.

- *Forgery*: A ciphertext $CT' = (CT'_0, CT'_1, CT'_2, ID'_s, ID'_r)$ for a message M' is forged by the adversary $\mathcal{A}_I$. We say that $\mathcal{A}_I$ wins the security game G_{AoS} if the message M' can be generated by the *Unsigncryption* algorithm.

Next, we discuss the advantage of $\mathcal{A}_I$ in two parts. In the first part, we consider the situation that the adversary $\mathcal{A}_I$ does not use any leak queries during the security game G_{AoS} and denote this advantage as $Adv^{nolq}_{\mathcal{A}_I}$. In the other part, we consider the situation that the adversary $\mathcal{A}_I$ uses leak queries, namely *Entity partial keys generation leak query*, *Signcryption leak query* and *Unsigncryption leak query*, during the security game G_{AoS}. We denote the advantage of this part as $Adv^{lq}_{\mathcal{A}_I}$. Both $Adv^{nolq}_{\mathcal{A}_I}$ and $Adv^{lq}_{\mathcal{A}_I}$ are analyzed as follows.

- $Adv^{nolq}_{\mathcal{A}_I} = \Pr[S_A] + \Pr[S_B] \leqq 216\phi^2/q + 3/q = O(\phi^2/q)$, and so can be negligible if $\phi = poly(logq)$, where $\Pr[S_A]$ and $\Pr[S_B]$ are computed as follows.
  - ✓ S_A is the situation under which a collision in $List_{G_1}$ or $List_{G_2}$ occurs. In $List_{G_1}$, a collision occurs when one polynomial $\mathbb{P}G_{1,i}$ is identical to another polynomial $\mathbb{P}G_{1,j}$. More specifically, $\mathbb{P}G_1(\mu_1, \mu_2, \ldots, \mu_v) = \mathbb{P}G_{1,i} - \mathbb{P}G_{1,j} = 0$ must be true. Here, $\mu_1, \mu_2, \ldots, \mu_v$ are random variables. For all queries in the *Query phase*, we observe all polynomials in $List_{G_1}$ and obtain a result that these polynomials are at most of degree 3. Hence, we employ Lemma 2 to obtain the probability of collision in $List_{G_1}$ is $(3/q)\binom{|List_{G_1}|}{2}$. Similarly, we can obtain the probability of collision in $List_{G_2}$ is $(6/q)\binom{|List_{G_2}|}{2}$. Due to $|List_{G_1}| + |List_{G_2}| \leqq 6\phi$, we have
$$\begin{aligned} Pr[S_A] &\leqq (3/q)\binom{|List_{G_1}|}{2} + (6/q)\binom{|List_{G_2}|}{2} \\ &\leqq (6/q)(|List_{G_1}| + |List_{G_2}|)^2 \\ &\leqq 216\phi^2/q. \end{aligned}$$
  - ✓ S_B is the situation of forging a valid tuple $(M', CT = (\mathbb{B}CT'_0, \mathbb{B}'_\alpha, CT'_2, ID_s, ID_r))$. As in the *Unsigncryption* algorithm, the identity $\hat{e}(g, CT_0) = SPK \cdot EPK_{ID_s} \cdot \hat{e}(KPK_{ID_s}, T \cdot K^{ID}) \cdot \hat{e}(CT_1, U \cdot V^{f'})$ holds and then the identity $\mathbb{P}g \cdot \mathbb{P}CT'_0 = \mathbb{P}SPK + \mathbb{P}EPK_{ID_s} + \mathbb{P}KPK_{ID_s} \cdot (\mathbb{P}T + ID \cdot \mathbb{P}K) + \mathbb{P}\alpha' \cdot (\mathbb{P}U + \mathbb{P}f \cdot \mathbb{P}V)$ also holds. Let $\mathbb{P}\delta = \mathbb{P}g \cdot \mathbb{P}CT'_0 - (\mathbb{P}SPK + \mathbb{P}EPK_{ID_s} + \mathbb{P}KPK_{ID_s} \cdot (\mathbb{P}T + ID \cdot \mathbb{P}K) + \mathbb{P}\alpha' \cdot (\mathbb{P}U + \mathbb{P}f \cdot \mathbb{P}V))$. Since 3 is the largest degree of $\mathbb{P}\delta$, we employ Lemma 2 to obtain the probability of $\mathbb{P}\delta = 0$ is $3/q$, namely $\Pr[S_B] = 3/q$.
- $Adv^{lq}_{\mathcal{A}_I} \leqq O((\phi^2/q) \cdot 2^{2\lambda}) + O(\phi^2/q) = O((\phi^2/q) \cdot 2^{2\lambda})$, and so can be negligible if $\lambda = poly(logq)$ according to Lemma 2. The arguments are shown as follows.
  - ✓ $\mathcal{A}_I$ issues the *Entity partial keys generation leak query*: Two leak results $\Lambda LF^{I}_{EPGK,i} = LF^{I}_{EPGK,i}$

$(SMK_{i,A})$ and $\Lambda LF^{II}_{EPGK,i} = LF^{II}_{EPGK,i}(SMK_{i,B})$ can be obtained by $\mathcal{A}_I$. Here, the system master key SMK can be obtained from $SMK_{0,A} \cdot SMK_{0,B} = SMK_{1,A} \cdot SMK_{1,B} = \ldots = SMK_{i-1,A} \cdot SMK_{i-1,B} = SMK_{i,A} \cdot SMK_{i,B}$. According to the techniques of key update [42] and $|LF^{I}_{EPGK,i}|$, $|\Lambda LF^{II}_{EPGK,i}| \leqq \lambda$, the adversary $\mathcal{A}_I$ can obtain at most 2λ bits of SMK.

✓ $\mathcal{A}_I$ issues the *Signcryption leak query*: Two leak results $\Lambda LF^{I}_{SC,j} = LF^{I}_{SC,j}(KSK_{ID_s,j,A}, ESK_{ID_s,j,A})$ and $\Lambda LF^{II}_{SC,j} = LF^{II}_{SC,j}(KSK_{ID_s,j,B}, ESK_{ID_s,j,B})$ can be obtained by $\mathcal{A}_I$. Here, the partial secret key KSK_{ID_s} and the entity secret key ESK_{ID_s} can be respectively obtained from $KSK_{ID_s,0,A} \cdot KSK_{ID_s,0,B} = KSK_{IDs,1,A} \cdot KSK_{ID_s,1,B} = \ldots = KSK_{ID_s,j-1,A} \cdot KSK_{ID_s,j-1,B} = KSK_{ID_s,j,A} \cdot KSK_{ID_s,j,B}$ and $ESK_{ID_s,0,A} \cdot ESK_{ID_s,0,B} = ESK_{ID_s,1,A} \cdot ESK_{ID_s,1,B} = \ldots = ESK_{ID_s,j-1,A} \cdot ESK_{ID_s,j-1,B} = ESK_{IDs,j,A} \cdot ESK_{ID_s,j,B}$. According to the techniques of key update and $|LF^{I}_{SC,j}|$, $|LF^{II}_{SC,j}| \leqq \lambda$, the adversary $\mathcal{A}_I$ can obtain at most 2λ bits of KSK_{ID_s} and ESK_{ID_s} totally.

✓ $\mathcal{A}_I$ issues the *Unsigncryption leak query*: Two leak results $\Lambda LF^{I}_{USC,k} = LF^{I}_{USC,k}(KSK_{ID_r,k,A}, ESK_{ID_r,k,A})$ and $\Lambda LF^{II}_{USC,k} = LF^{II}_{USC,k}(KSK_{ID_r,k,B}, ESK_{ID_r,k,B})$ can be obtained by $\mathcal{A}_I$. Here, the partial secret key KSK_{ID_r} and the entity secret key ESK_{ID_r} can be respectively obtained from $KSK_{ID_r,0,A} \cdot KSK_{ID_r,0,B} = KSK_{ID_r,1,A} \cdot KSK_{ID_r,1,B} = \ldots = KSK_{ID_r,k-1,A} \cdot KSK_{ID_r,k-1,B} = KSK_{ID_r,k,A} \cdot KSK_{ID_r,k,B}$ and $ESK_{ID_r,0,A} \cdot ESK_{ID_r,0,B} = ESK_{ID_r,1,A} \cdot ESK_{ID_r,1,B} = \ldots = ESK_{ID_r,k-1,A} \cdot ESK_{ID_r,k-1,B} = ESK_{ID_r,k,A} \cdot ESK_{ID_r,k,B}$. According to the techniques of key update and $|LF^{I}_{USC,k}|$, $|LF^{II}_{USC,k}| \leqq \lambda$, the adversary $\mathcal{A}_I$ can obtain at most 2λ bits of KSK_{ID_r} and ESK_{ID_r} totally.

Next, the advantage that $\mathcal{A}_I$ wins this game is the probability that the ciphertext $CT' = (CT'_0, CT'_1, CT'_2, ID'_s, ID'_r)$ for a message M' can be forged by using the system master key SMK, the partial secret key KSK_{ID} and the entity secret key ESK_{ID}. Hence, four events are discussed as follows.

(1) Event E_{SMK}: The system master key SMK can be obtained by $\mathcal{A}_I$ from $\Lambda LF^{I}_{EPGK,i}$ and $\Lambda LF^{II}_{EPGK,i}$. Meanwhile, let $\overline{E_{SMK}}$ denote E_{SMK}'s complement event.

(2) Event E_{KSK}: The partial secret key KSK can be obtained by $\mathcal{A}_I$ from $\Lambda LF^{I}_{SC,j}$, $\Lambda LF^{II}_{SC,j}$, $\Lambda LF^{I}_{USC,k}$ and $\Lambda LF^{II}_{USC,k}$. Meanwhile, let $\overline{E_{KSK}}$ denote E_{KSK}'s complement event.

(3) Event E_{ESK}: The entity secret key ESK can be obtained by $\mathcal{A}_I$ from $\Lambda LF^{I}_{SC,j}$, $\Lambda LF^{II}_{SC,j}$, $\Lambda LF^{I}_{USC,k}$ and $\Lambda LF^{II}_{USC,k}$. Meanwhile, the event $\overline{E_{ESK}}$ is E_{ESK}'s complement.

(4) Event E_{MSF}: the ciphertext $CT' = (CT'_0, CT'_1, CT'_2, ID'_s, ID'_r)$ for a message M' can be successfully forged.

Considering these events, we compute the probability $\Pr[\mathcal{A}_I]$ that $\mathcal{A}_I$ wins this game as follows.

$$\begin{aligned}
Pr[\mathcal{A}_I] &= Pr[E_{MSF}] \\
&= Pr[E_{MSF} \wedge (E_{SMK} \vee E_{KSK} \vee E_{ESK})] \\
&\quad + Pr[E_{MSF} \wedge (\overline{E_{SMK}} \wedge \overline{E_{KSK}} \wedge \overline{E_{ESK}})] \\
&\leqq Pr[E_{SMK} \vee E_{KSK} \vee E_{ESK}] \\
&\quad + Pr[E_{MSF} \wedge (\overline{E_{SMK}} \wedge \overline{E_{KSK}} \wedge \overline{E_{ESK}})].
\end{aligned}$$

According to Lemma 2, the *Entity partial keys generation leak query*, the *Signcryption leak query* and the *Unsigncryption leak query*, we have $\Pr[E_{SMK} \vee E_{KSK} \vee E_{ESK}] \leqq Adv^{nolq}_{\mathcal{A}_I} \cdot 2^{2\phi} \leqq O((\phi^2/q) \cdot 2^{2\lambda})$. Next, we discuss $\Pr[E_{MSF} \wedge (\overline{E_{SMK}} \wedge \overline{E_{KSK}} \wedge \overline{E_{ESK}})]$ which states the probability of successful forgery without the help of information of SMK, KSK and ESK. Hence, we have $\Pr[E_{MSF} \wedge (\overline{E_{SMK}} \wedge \overline{E_{KSK}} \wedge \overline{E_{ESK}})] = Adv^{nolq}_{\mathcal{A}_I} = O(\phi^2/q)$. Finally, we have

$$\begin{aligned}
Pr[\mathcal{A}_I] &= Pr[E_{MSF}] \\
&= Pr[E_{MSF} \wedge (E_{SMK} \vee E_{KSK} \vee E_{ESK})] \\
&\quad + Pr[E_{MSF} \wedge (\overline{E_{SMK}} \wedge \overline{E_{KSK}} \wedge \overline{E_{ESK}})] \\
&\leqq O((\phi^2/q) \cdot 2^{2\lambda}) + O(\phi^2/q) = O((\phi^2/q) \cdot 2^{2\lambda}).
\end{aligned}$$

Theorem 2: Assume that the HF and DL assumptions hold. Under the GBG model, the proposed LR-CLSC scheme is secure for the existential unforgeability against the adversary $\mathcal{A}_{II}$ in the security game G_{AoS}.

Proof: Let $\mathcal{C}$ be the challenger and interact with the adversary $\mathcal{A}_{II}$ in the following security game G_{AoS}.

- *Setup phase*: This phase is the same as that in the proof of Theorem 1, except that $\mathbb{B}SMK$ can be obtained by the adversary $\mathcal{A}_{II}$.
- *Query phase*: This phase is the same as that in the proof of Theorem 1, except that the *Entity partial keys generation query* and *Entity partial keys generation leak query* are not necessary anymore because $\mathcal{A}_{II}$ has $\mathbb{B}SMK$ and can execute the relevant algorithms to obtain the results.
- *Forgery*: A ciphertext $CT' = (CT'_0, CT'_1, CT'_2, ID'_s, ID'_r)$ for a message M' is forged by the adversary $\mathcal{A}_{II}$. Here, the *Entity keys generation query*(ID'), *Entity Public key replace query*$(ID, \mathbb{B}KPK'_{ID}, \mathbb{B}EPK'_{ID})$ and *Signcryption query*(M', ID'_s, ID'_r) cannot occur in this game. Then, we say that $\mathcal{A}_{II}$ wins the security game G_{AoS} if the message M' can be generated by the *Unsigncryption* algorithm.

Next, we discuss the advantage of the other type of adversary, $\mathcal{A}_{II}$. As same as the security analysis in the proof of Theorem 1, the advantage is divided into $Adv^{nolq}_{\mathcal{A}_{II}}$ and $Adv^{lq}_{\mathcal{A}_{II}}$. By a similar way, we have the advantage $Adv^{nolq}_{\mathcal{A}_{II}} = \Pr[S_A]+\Pr[S_B] \leqq 216\phi^2/q + 3/q = O(\phi^2/q)$, and so can be negligible if $\phi = poly(logq)$. Next, we consider the *Signcryption leak query* and *Unsigncryption leak query*, and

TABLE 2. Performance comparisons of our LR-CLSC with existing CLSC and two LR-CLSC.

	Rastegart *et al.*'s CLSC [27]	Zhou *et al.*'s LR-CLSC [35]	Yang *et al.*'s LR-CLSC [36]	Our LR-CLSC
Allowing entity's secret key to be leaked	No	Yes	Yes	Yes
Allowing system's secret key to be leaked	No	No	No	Yes
Leakage model	Not provided	Bounded	Bounded	Continual

TABLE 3. Cost required for computing a bilinear pairing and an exponentiation.

Operations	C_{pair}	C_{exp}
Computational cost	7.8351 ms	0.4746 ms

obtain the advantage $Adv^{lq}_{\mathcal{A}_{II}} \leqq O((\phi^2/q)\cdot 2^{2\lambda}) + O(\phi^2/q) = O((\phi^2/q)\cdot 2^{2\lambda})$ and so can be negligible if $\lambda = poly(logq)$ according to Lemma 2. The detailed processes are shown as follows.

✓ $\mathcal{A}_{II}$ issues the *Signcryption leak query*: Two leak results $\Lambda LF^{I}_{SC,j} = LF^{I}_{SC,j}(KSK_{ID_s,j,A}, ESK_{ID_s,j,A})$ and $\Lambda LF^{II}_{SC,j} = LF^{II}_{SC,j}(KSK_{ID_s,j,B}, ESK_{ID_s,j,B})$ can be obtained by $\mathcal{A}_{II}$. Here, the partial secret key KSK_{ID_s} and the entity secret key ESK_{ID_s} can be respectively obtained from $KSK_{ID_s,0,A}\cdot KSK_{ID_s,0,B} = KSK_{ID_s,1,A}\cdot KSK_{ID_s,1,B} = \ldots = KSK_{ID_s,j-1,A}\cdot KSK_{ID_s,j-1,B} = KSK_{ID_s,j,A}\cdot KSK_{ID_s,j,B}$ and $ESK_{ID_s,0,A}\cdot ESK_{ID_s,0,B} = ESK_{ID_s,1,A}\cdot ESK_{ID_s,1,B} = \ldots = ESK_{ID_s,j-1,A}\cdot ESK_{ID_s,j-1,B} = ESK_{ID_s,j,A}\cdot ESK_{ID_s,j,B}$. According to the techniques of key update and $|LF^{I}_{SC,j}|$, $|LF^{II}_{SC,j}| \leqq \lambda$, the adversary $\mathcal{A}_{II}$ can obtain at most 2λ bits of KSK_{ID_s} and ESK_{ID_s} totally.

✓ $\mathcal{A}_{II}$ issues the *Unsigncryption leak query*: Two leak results $\Lambda LF^{I}_{USC,k} = LF^{I}_{USC,k}(KSK_{ID_r,k,A}, ESK_{ID_r,k,A})$ and $\Lambda LF^{II}_{USC,k} = LF^{II}_{USC,k}(KSK_{ID_r,k,B}, ESK_{ID_r,k,B})$ can be obtained by $\mathcal{A}_{II}$. Here, the partial secret key KSK_{ID_r} and the entity secret key ESK_{ID_r} can be respectively obtained from $KSK_{ID_r,0,A}\cdot KSK_{ID_r,0,B} = KSK_{ID_r,1,A}\cdot KSK_{ID_r,1,B} = \ldots = KSK_{ID_r,k-1,A}\cdot KSK_{ID_r,k-1,B} = KSK_{ID_r,k,A}\cdot KSK_{ID_r,k,B}$ and $ESK_{ID_r,0,A}\cdot ESK_{ID_r,0,B} = ESK_{ID_r,1,A}\cdot ESK_{ID_r,1,B} = \ldots = ESK_{ID_r,k-1,A}\cdot ESK_{ID_r,k-1,B} = ESK_{ID_r,k,A}\cdot ESK_{ID_r,k,B}$. According to the techniques of key update and $|LF^{I}_{USC,k}|$, $|LF^{II}_{USC,k}| \leqq \lambda$, the adversary $\mathcal{A}_{II}$ can obtain at most 2λ bits of KSK_{ID_r} and ESK_{ID_r} totally.

Next, the advantage that $\mathcal{A}_{II}$ wins this game is the probability that the ciphertext $CT' = (CT'_0, CT'_1, CT'_2, ID'_s, ID'_r)$ for a message M' can be forged by using the partial secret key KSK_{ID} and the entity secret key ESK_{ID}. Hence, we discuss three events as follows.

(1) Event E_{KSK}: The partial secret key KSK can be obtained by $\mathcal{A}_{II}$ from $\Lambda LF^{I}_{SC,j}$, $\Lambda LF^{II}_{SC,j}$, $\Lambda LF^{I}_{USC,k}$ and $\Lambda LF^{II}_{USC,k}$. Meanwhile, let $\overline{E_{KSK}}$ denote E_{KSK}'s complement event.

(2) Event E_{ESK}: The entity secret key ESK can be obtained by $\mathcal{A}_{II}$ from $\Lambda LF^{I}_{SC,j}$, $\Lambda LF^{II}_{SC,j}$, $\Lambda LF^{I}_{USC,k}$ and $\Lambda LF^{II}_{USC,k}$. Meanwhile, let $\overline{E_{ESK}}$ denote E_{ESK}'s complement event.

(3) Event E_{MSF}: the ciphertext $CT' = (CT'_0, CT'_1, CT'_2, ID'_s, ID'_r)$ for a message M' can be successfully forged.

Considering these events, we compute the probability $\Pr[\mathcal{A}_{II}]$ that $\mathcal{A}_{II}$ wins this game as follows.

$$\begin{aligned} Pr[\mathcal{A}_{II}] &= Pr[E_{MSF}] \\ &= Pr[E_{MSF} \wedge (E_{KSK} \vee E_{ESK})] \\ &\quad + Pr[E_{MSF} \wedge (\overline{E_{KSK}} \wedge \overline{E_{ESK}})] \\ &\leq qPr[E_{KSK} \vee E_{ESK}] \\ &\quad + Pr[E_{MSF} \wedge (\overline{E_{KSK}} \wedge \overline{E_{ESK}})]. \end{aligned}$$

According to Lemma 2, the *Entity partial keys generation leak query*, the *Signcryption leak query* and the *Unsigncryption leak query*, we have $\Pr[E_{KSK} \vee E_{ESK}] \leqq Adv^{nolq}_{\mathcal{A}_{II}} \cdot 2^{2\lambda} \leqq O((\phi^2/q)\cdot 2^{2\lambda})$. Next, we discuss $\Pr[E_{MSF} \wedge (\overline{E_{KSK}} \wedge \overline{E_{ESK}})]$ which states the probability of successful forgery without the help of information of KSK and ESK. Hence, we have $\Pr[E_{MSF} \wedge (\overline{E_{KSK}} \wedge \overline{E_{ESK}})] = Adv^{nolq}_{\mathcal{A}_{II}} = O(\phi^2/q)$. Finally, we have

$$\begin{aligned} Pr[\mathcal{A}_{II}] &= Pr[E_{MSF}] \\ &= Pr[E_{MSF} \wedge (E_{KSK} \vee E_{ESK})] \\ &\quad + Pr[E_{MSF} \wedge (\overline{E_{KSK}} \wedge \overline{E_{ESK}})] \\ &\leqq O((\phi^2/q)\cdot 2^{2\lambda}) + O(\phi^2/q) = O((\phi^2/q)\cdot 2^{2\lambda}). \end{aligned}$$

Theorem 3: Assume that the HF and DL assumptions hold. Under the GBG model, the proposed LR-CLSC scheme is secure for the ciphertexts indistinguishability against the adversary $\mathcal{A}_I$ in the security game G_{CoE}.

Proof: Let $\mathcal{C}$ be the challenger and interact with the adversary $\mathcal{A}_I$ in the following security game G_{CoE}.

- *Setup phase*: This phase is the same as that in the proof of Theorem 1.
- *Query phase*: This phase is the same as that in the proof of Theorem 1.
- *Challenge phase*: The adversary $\mathcal{A}_I$ picks an identity ID'_r, two messages M'_0 and M'_1 as a challenge objective. Here, the identity ID'_r can never appear in the *Entity partial keys generation query*. The challenger $\mathcal{C}$ randomly chooses a $coin \in \{0, 1\}$, and generates a challenge ciphertext CT' by running the *Signcryption* algorithm with (M'_{coin}, ID_s, ID'_r). The challenge ciphertext CT' is sent to the adversary $\mathcal{A}_I$.

TABLE 4. Computational cost of our LR-CLSC.

	Setup	Entity partial keys generation	Entity keys generation	Signcryption	Unsigncryption
Our	$C_{pair} + 7C_{exp}$	$7C_{exp}$	$C_{pair} + 3C_{exp}$	$C_{pair} + 8C_{exp}$	$7C_{pair} + 2C_{exp}$
LR-CLSC	11.1573 ms	3.3222 ms	9.2589 ms	11.6319 ms	55.7949 ms

TABLE 5. Performance comparisons between our LR-CLSC and LR-CLS + LR-CLE schemes.

	Signcryption	Unsigncryption
Our LR-CLSC	$C_{pair} + 8C_{exp}$ (11.6319 ms)	$7C_{pair} + 2C_{exp}$ (55.7949 ms)
LR-CLS [45] + LR-CLE [44]	$4C_{pair} + 11C_{exp}$ (36.561 ms)	$7C_{pair} + 6C_{exp}$ (57.6933 ms)

- *Guess phase*: A guess $coin' \in \{0, 1\}$ is output by the adversary $\mathcal{A}_I$. We say that $\mathcal{A}_I$ wins the game if $coin' = coin$. The winning advantage is defined as $Adv(\mathcal{A}_I) = |\Pr[coin' = coin] - 1/2|$.

Next, we discuss the advantage of $Adv(\mathcal{A}_I)$. The advantage is divided into two parts. In the first part, we consider the situation that the adversary $Adv(\mathcal{A}_I)$ does not use any leak queries during the security game G_{CoE} and denote this advantage as $Adv^{nolq}_{\mathcal{A}_I}$. In the other part, we consider that the adversary $\mathcal{A}_I$ uses leak queries, namely *Entity partial keys generation leak query*, *Signcryption leak query* and *Unsigncryption leak query*, during the security game G_{CoE}. We denote the advantage of this part as $Adv^{lq}_{\mathcal{A}_I}$. Both $Adv^{nolq}_{\mathcal{A}_I}$ and $Adv^{lq}_{\mathcal{A}_I}$ are analyzed as follows.

- $Adv^{nolq}_{\mathcal{A}_I} = |\Pr[coin = coin'] - 1/2| = \Pr[S_A] + \Pr[S_B] \leqq 216\phi^2/q = O(\phi^2/q)$, and so can be negligible if $\phi = poly(logq)$, where $\Pr[S_A]$ and $\Pr[S_B]$ are computed as follows.
  - ✓ S_A is the situation under which a collision in $List_{G_1}$ or $List_{G_2}$ occurs. We obtain $\Pr[S_A] \leqq 216\phi^2/q$ by a similar way as in the proof of Theorem 1.
  - ✓ $\Pr[S_B]$ is the probability of guessing $coin = coin'$, and so $\Pr[S_B] = 1/2$.
- $Adv^{lq}_{\mathcal{A}_I} \leqq O((\phi^2/q) \cdot 2^{2\lambda})$, and so can be negligible if $\lambda = poly(logq)$ according to Lemma 2. The detailed processes are the same as those in the proof of Theorem 1.

Theorem 4: Assume that the HF and DL assumptions hold. Under the GBG model, the proposed LR-CLSC scheme is secure for the ciphertexts indistinguishability against the adversary $\mathcal{A}_{II}$ in the security game G_{CoE}.

Proof: Let $\mathcal{C}$ be the challenger and interact with the adversary $\mathcal{A}_{II}$ in the following security game G_{CoE}.

- *Setup phase*: This phase is the same as that in the proof of Theorem 2.
- *Query phase*: This phase is the same as that in the proof of Theorem 2.
- *Challenge phase*: The adversary $\mathcal{A}_{II}$ picks an identity ID'_r and two message pair M'_0 and M'_1 as a challenge objective. Here, the identity ID'_r can never appear in the *Entity keys generation query* nor *Entity Public key replace query*. The challenger $\mathcal{C}$ randomly chooses a $coin \in \{0, 1\}$, and generates a challenge ciphertext CT' by running the *Signcryption* algorithm with (M'_{coin}, ID_s, ID'_r). The challenge ciphertext CT' is sent to the adversary $\mathcal{A}_{II}$.
- *Guess phase*: A guess $coin' \in \{0, 1\}$ is output by the adversary $\mathcal{A}_{II}$. We say that $\mathcal{A}_{II}$ wins the game if $coin' = coin$. The winning advantage is defined as $Adv(\mathcal{A}_{II}) = |\Pr[coin' = coin] - 1/2|$.

Next, we discuss the advantage of $\mathcal{A}_{II}$. The advantage is divided into two parts. In the first part, we consider the situation that the adversary $\mathcal{A}_{II}$ does not use any leak queries during the security game G_{CoE} and denote the advantage as $Adv^{nolq}_{\mathcal{A}_{II}}$. In the other part, we consider the situation that the adversary $\mathcal{A}_{II}$ uses leak queries, namely *Signcryption leak query* and *Unsigncryption leak query*, during the security game G_{CoE}. We denote the advantage of this part as $Adv^{lq}_{\mathcal{A}_{II}}$. By a similar way as in the proof of Theorem 3, we have $Adv^{nolq}_{\mathcal{A}_{II}} \leqq 216\phi^2/q = O(\phi^2/q)$ and the advantage can be negligible if $\phi = poly(logq)$. By a similar way as in the proof of Theorem 2, we have $Adv^{lq}_{\mathcal{A}_{II}} \leqq O((\phi^2/q) \cdot 2^{2\lambda})$, and so can be negligible if $\lambda = poly(logq)$ according to Lemma 2.

VII. COMPARISONS

We provide a comparison of characteristics of the proposed LR-CLSC scheme with the existing CLSC scheme [26] and two LR-CLSC schemes [34], [35]. Table 2 lists the comparisons under three situations, namely, allowing entity secret key to be leaked, allowing system secret key to be leaked and leakage model. Although the two LR-CLSC schemes in [34] and [35] can resist side-channel attacks, there are two limitations. One limitation is that they only allow the entity's secret key to be leaked, but cannot allow the system's secret key to be leaked. The other limitation is that the leakage model of the two schemes is bounded which makes the model not practical.

Next, we introduce two symbols to analyze the computational cost of all algorithms of our LR-CLSC scheme.

- C_{pair}: the cost required for computing a bilinear pairing $\hat{e} : G_1 \times G_1 \to G_2$.
- C_{exp}: the cost required for computing an exponentiation in G_1 or G_2.

Based on the simulation conducted in [45], C_{pair} is equal to 7.8351 ms and C_{exp} is equal to 0.4746 ms, as indicated in Table 3. The simulation was carried out by using an

Intel Core i7-8550U CPU 1.80 GHz processor and taking a finite field F_p, G_1, and G_2 as input parameters. The value of p is a prime number with 256 bits, while G_1 and G_2 are groups with a prime order of 224 bits over the finite field F_p. Table 4 lists the computational cost in terms of *Setup*, *Entity partial keys generation*, *Entity keys generation*, *Signcryption* and *Unsigncryption* algorithms. Table 4 indicates that this execution time of all algorithms of our LR-CLSC scheme is efficient. It is worth mentioning that there are two related schemes, namely leakage-resilient certificateless signature (LR-CLS) [44] and leakage-resilient certificateless encryption (LR-CLE) scheme [43], already in existence that satisfy continual leakage model. If we combine the two schemes, the *Signcryption* and *Unsigncryption* processes can also be achieved. By Table 5, we can see that the performance of our proposed scheme is better than the combination performance of the two schemes.

VIII. CONCLUSION

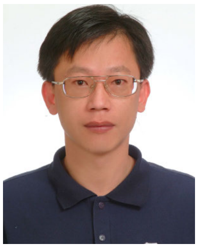
Our paper introduced the *first* LR-CLSC scheme designed to resist side-channel attacks under a continual leakage model. We presented the syntax of the LR-CLSC scheme and proposed a new security model of LR-CLSC. Assume that the DL and HF assumptions hold, the proposed scheme has been formally proven to be secure in the GBG model. In addition, the proposed scheme outperformed the previous LR-CLS, LR-CLE, and LR-CLSC schemes by achieving resistance against side-channel attacks in a continual leakage model.

REFERENCES

[1] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.
[2] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proc. CRYPTO*, vol. 196, 1984, pp. 10–18.
[3] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," *J. Cryptol.*, vol. 20, no. 3, pp. 265–294, Jul. 2007.
[4] F. Yu, X. Kong, A. A. M. Mokbel, W. Yao, and S. Cai, "Complex dynamics, hardware implementation and image encryption application of multiscroll memeristive Hopfield neural network with a novel local active memeristor," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 1, pp. 326–330, Jan. 2023.
[5] T.-Y. Wu, Q. Meng, Y.-C. Chen, S. Kumari, and C.-M. Chen, "Toward a secure smart-home IoT access control scheme based on home registration approach," *Mathematics*, vol. 11, no. 9, p. 2123, Apr. 2023.
[6] T.-Y. Wu, F. Kong, Q. Meng, S. Kumari, and C.-M. Chen, "Rotating behind security: An enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture," *EURASIP J. Wireless Commun. Netw.*, vol. 2023, no. 1, Apr. 2023, Art. no. 36.
[7] S. Wang and X. Zhou, "IoT data security authentication and key negotiation scheme based on edge computing and blockchain," *J. Netw. Intell.*, vol. 8, no. 2, pp. 363–379, May 2023.
[8] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost (encryption)," in *Proc. Annu. Int. Cryptol. Conf.*, 1997, pp. 165–179.
[9] Y. Zheng and H. Imai, "How to construct efficient signcryption schemes on elliptic curves," *Inf. Process. Lett.*, vol. 68, no. 5, pp. 227–233, Dec. 1998.
[10] B. H. Yum and P. J. Lee, "New signcryption schemes based on KCDSA," in *Proc. ICISC*, 2001, vol. 2288, pp. 305–317.
[11] J. H. An, Y. Dodis, and T. Rabin, "On the security of joint signature and encryption," in *Proc. EUROCRYPT*, vol. 2332, 2002, pp. 83–107.
[12] J. Malone-Lee and W. Mao, "Two birds one stone: Signcryption using RSA," in *Proc. CT-RSA*, vol. 2612, 2003, pp. 211–226.
[13] A. W. Dent, "Hybrid signcryption schemes with outsider security," in *Proc. ISC*, vol. 3650, 2005, pp. 203–217.
[14] J. Malone-Lee, "Identity-based signcryption," Cryptol. ePrint Arch., IACR, Pittsburgh, PA, USA, Tech. Rep. 098-2002, 2002.
[15] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. CRYPTO*, vol. 196, 1984, pp. 47–53.
[16] B. Libert and J. J. Quisquater, "A new identity based signcryption scheme from pairings," in *Proc. IEEE Inf. Theory Workshop*, Mar./Apr. 2003, pp. 155–158.
[17] X. Boyen, "Multipurpose identity-based signcryption," in *Proc. CRYPTO*, vol. 2729, 2003, pp. 383–399.
[18] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Proc. PKC*, vol. 3386, 2005, pp. 362–379.
[19] H. Xiong, Y. Hou, X. Huang, and Y. Zhao, "Secure message classification services through identity-based signcryption with equality test towards the Internet of Vehicles," *Veh. Commun.*, vol. 26, Dec. 2020, Art. no. 100264.
[20] H. Zhu, Y. Wang, C. Wang, and X. Cheng, "An efficient identity-based proxy signcryption using lattice," *Future Gener. Comput. Syst.*, vol. 117, pp. 321–327, Apr. 2021.
[21] Y. Zhao, Y. Wang, Y. Liang, H. Yu, and Y. Ren, "Identity-based broadcast signcryption scheme for vehicular platoon communication," *IEEE Trans. Ind. Informat.*, vol. 169, no. 6, pp. 7814–7824, Jun. 2023, doi: 10.1109/TII.2022.3203724.
[22] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. ASIACRYPT*, vol. 2894, 2003, pp. 452–473.
[23] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proc. ASIACCS*, 2008, pp. 369–372.
[24] Z. Liu, Y. Hu, X. Zhang, and H. Ma, "Certificateless signcryption scheme in the standard model," *Inf. Sci.*, vol. 180, no. 3, pp. 452–464, Feb. 2010.
[25] S. Miao, F. Zhang, S. Li, and Y. Mu, "On security of a certificateless signcryption scheme," *Inf. Sci.*, vol. 232, pp. 475–481, May 2013.
[26] P. Rastegari, W. Susilo, and M. Dakhlalian, "Efficient certificateless signcryption in the standard model: Revisiting Luo and Wan's scheme from wireless personal communications (2018)," *Comput. J.*, vol. 62, no. 8, pp. 1178–1193, Aug. 2019.
[27] J. Alwen, Y. Dodis, and D. Wichs, "Leakage-resilient public-key cryptography in the bounded-retrieval model," in *Proc. CRYPTO*, vol. 5677, 2009, pp. 36–54.
[28] J. Katz and V. Vaikuntanathan, "Signature schemes with bounded leakage resilience," in *Proc. ASIACRYPT*, vol. 5912, 2009, pp. 703–720.
[29] J.-D. Wu, Y.-M. Tseng, S.-S. Huang, and T.-T. Tsai, "Leakage-resilient certificate-based key encapsulation scheme resistant to continual leakage," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 131–144, 2020.
[30] T.-T. Tsai, Y.-H. Chuang, Y.-M. Tseng, S.-S. Huang, and Y.-H. Hung, "A leakage-resilient ID-based authenticated key exchange protocol with a revocation mechanism," *IEEE Access*, vol. 9, pp. 128633–128647, 2021.
[31] A. L. Peng, Y. M. Tseng, and S. S. Huang, "An efficient leakage-resilient authenticated key exchange protocol suitable for IoT devices," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5343–5354, Dec. 2021.
[32] T.-T. Tsai, S.-S. Huang, Y.-M. Tseng, Y.-H. Chuang, and Y.-H. Hung, "Leakage-resilient certificate-based authenticated key exchange protocol," *IEEE Open J. Comput. Soc.*, vol. 3, pp. 137–148, 2022.
[33] Q. Yu, J. Li, and S. Ji, "Fully secure ID-based signature scheme with continuous leakage resilience," *Secur. Commun. Netw.*, vol. 2022, pp. 1–12, Jan. 2022.
[34] Y. Zhou, B. Yang, and W. Zhang, "Provably secure and efficient leakage-resilient certificateless signcryption scheme without bilinear pairing," *Discrete Appl. Math.*, vol. 204, pp. 185–202, May 2016.
[35] Q. Yang, Y. Zhou, and Y. Yu, "Leakage-resilient certificateless signcryption scheme," in *Proc. GLOBECOM Workshops*, Dec. 2019, pp. 1–6.
[36] T. Kubota, K. Yoshida, M. Shiozaki, and T. Fujino, "Deep learning side-channel attack against hardware implementations of AES," *Microprocessors Microsyst.*, vol. 87, Nov. 2021, Art. no. 103383.
[37] K. Ngo, E. Dubrova, Q. Guo, and T. Johansson, "A side-channel attack on a masked IND-CCA secure saber KEM implementation," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, pp. 676–707, Aug. 2021.
[38] H. Xiong, T. H. Yuen, C. Zhang, S. M. Yiu, and Y.-J. He, "Leakage-resilient certificateless public key encryption," in *Proc. 1st ACM Workshop Asia Public-Key Cryptogr.*, May 2013, pp. 13–22.
[39] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. CRYPTO*, vol. 2139, 2001, pp. 213–229.
[40] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Proc. EUROCRYPT*, vol. 3494, 2005, pp. 440–456.

[41] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, Jan. 2008.
[42] D. Galindo and S. Virek, "A practical leakage-resilient signature scheme in the generic group model," in *Proc. SAC*, vol. 7707, 2013, pp. 50–65.
[43] J.-D. Wu, Y.-M. Tseng, S.-S. Huang, and W.-C. Chou, "Leakage-resilient certificateless key encapsulation scheme," *Informatica*, vol. 29, no. 1, pp. 125–155, Jan. 2018.
[44] J.-D. Wu, Y. Tseng, and S.-S. Huang, "Leakage-resilient certificateless signature under continual leakage model," *Inf. Technol. Control*, vol. 47, no. 2, pp. 363–386, Jun. 2018.
[45] Y. Li, Q. Cheng, X. Liu, and X. Li, "A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing," *IEEE Syst. J.*, vol. 15, no. 1, pp. 935–946, Mar. 2021.

TUNG-TSO TSAI received the Ph.D. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2014, under the supervision of Prof. Yuh-Min Tseng. He is currently an Assistant Professor with the Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan. His current research interests include applied cryptography, pairing-based cryptography, and leakage-resilient cryptography.



YUH-MIN TSENG (Member, IEEE) is currently the Vice President and a Professor with the Department of Mathematics, National Changhua University of Education, Taiwan. He has published over 100 scientific journal articles on various research areas of cryptography, security, and computer networks. His current research interests include cryptography, network security, computer networks, and leakage-resilient cryptography. He is a member of the IEEE Computer Society, the IEEE Communications Society, and the Chinese Cryptology and Information Security Association (CCISA). He serves as an editor for several international journals.

SEN-SHAN HUANG received the Ph.D. degree from the University of Illinois at Urbana–Champaign, in 1997, under the supervision of Prof. Bruce C. Berndt. He is currently a Professor with the Department of Mathematics, National Changhua University of Education, Taiwan. His current research interests include number theory, cryptography, and leakage-resilient cryptography.