INFORMATICA, 2023, Vol. 34, No. 1, 199–222

DOI: https://doi.org/10.15388/22-INFOR506

Fully Continuous Leakage-Resilient Certificate-Based Signcryption Scheme for Mobile Communications

Yuh-Min TSENG[1,*], Tung-Tso TSAI[2], Sen-Shan HUANG[1]

[1] *Department of Mathematics, National Changhua University of Education, Changhua 500, Taiwan*
[2] *Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan*
e-mail: ymtseng@cc.ncue.edu.tw

Received: September 2022; accepted: December 2022

Abstract. Due to the popularity of mobile communication, many computing devices are exposed to remote environments without physical protection so that these devices easily suffer from leakage attacks (e.g., side-channel attacks). Under such leakage attacks, when a computing device performs some cryptographic algorithm, an adversary may acquire partial bits of secret keys participated in this cryptographic algorithm. To resist leakage attacks, researchers offer leakage-resilient cryptography as a solution. A signcryption scheme combines signing and encrypting processes to simultaneously provide both authentication and confidentiality, which is an important cryptographic primitive. Indeed, many leakage-resilient signcryption schemes under various public key system (PKS) settings were proposed. Unfortunately, these schemes still have two shortcomings, namely, bounded leakage resilience and conditionally continuous leakage resilience. In this paper, a "fully" continuous leakage-resilient certificate-based signcryption (FCLR-CBSC) scheme is proposed. Security analysis is formally proved to show that our scheme possesses both authentication and confidentiality against two types of adversaries in the certificate-based PKS setting. Performance analysis and simulation experience show that our scheme is suited to run on both a PC and a mobile device.

Key words: leakage attacks, signcryption, certificate-based public key system, leakage resilience.

1. Introduction

In the traditional public key system (PKS) setting (Rivest *et al.*, 1978), a public-key infrastructure (PKI) needs to be established to create and manage each member's certificate, which is used to validate the member's public key. To lighten the PKI establishment cost, Boneh and Franklin (2001) presented a practical identity-based PKS (ID-PKS) setting with bilinear pairings, in which a member's identity is regarded as the member's public key and no certificate is needed. However, the ID-PKS setting suffers from a constitutional key escrow problem. In 2003, the certificateless PKS (CL-PKS) (Al-Riyami and Paterson,

*Corresponding author.

2003) and the certificate-based PKS (CB-PKS) (Gentry, 2003) settings were constructed respectively to clear up the key escrow problem. Afterwards, the research of various cryptographic mechanisms under the CB-PKS and the CL-PKS settings have been thoroughly studied.

Typically, the security of these cryptographic mechanisms under these PKS settings mentioned above is dependent on the security of secret keys participated in these cryptographic mechanisms, and so these secret keys must be entirely concealed to adversaries. However, due to the popularity of mobile communication, many computing devices are exposed to remote environments without physical protection so that these devices easily suffer from leakage attacks (e.g. side-channel attacks) (Kocher *et al.*, 1999; Brumley and Boneh, 2005; Biham *et al.*, 2008). By leakage attacks, when a computing device performs some cryptographic algorithm, an adversary may acquire partial bits of secret keys participated in this algorithm. To resist such leakage attacks, researchers offer leakage-resilient cryptography as a solution. In the past, numerous leakage-resilient signature (LRS) schemes (Galindo and Virek, 2013; Wu *et al.*, 2019; Tseng *et al.*, 2020; Wu *et al.*, 2020b), leakage-resilient encryption (LRE) schemes (Kiltz and Pietrzak, 2010; Galindo *et al.*, 2016; Wu *et al.*, 2018, 2020a; Tseng *et al.*, 2022), and leakage-resilient authenticated key agreement protocols (Tseng *et al.*, 2021; Peng *et al.*, 2021; Tsai *et al.*, 2022) under various PKS settings have been published in the literature.

For reducing communication and computation costs, a signcryption scheme (Zheng, 1997) combines signing and encrypting processes in a mechanism to simultaneously provide both authentication and confidentiality, which is an important cryptographic primitive. In the past, some signcryption schemes under various PKS settings have been proposed that include PKI-based signcryption (PKI-SC) schemes (Ullah *et al.*, 2020; Ali *et al.*, 2020), certificateless signcryption (CLSC) schemes (Khan *et al.*, 2020; Wu *et al.*, 2022) and certificate-based signcryption (CBSC) schemes (Ullah *et al.*, 2019; Hussain *et al.*, 2020). Nevertheless, these signcryption schemes mentioned above are unable to resist leakage attacks. Indeed, several leakage-resilient (LR) signcryption schemes under the CL-PKS and the CB-PKS settings have been proposed. However, these LR signcryption schemes still have two shortcomings, that is, bounded leakage resilience and conditionally continuous leakage resilience, which will be discussed later. Hence, in this paper, we aim to propose a "fully" continuous leakage-resilient certificate-based signcryption (FCLR-CBSC) scheme to remove the shortcomings of the previously proposed schemes.

1.1. *Related Work*

Here, let's introduce two types of leakage attack models, that is, bounded and continuous (i.e. unbounded). The bounded leakage attack model has an impractical property that entire leaked bits of a secret key are bounded in a fractional proportion of the secret key during the usage life of an LR algorithm (Alwen *et al.*, 2009; Katz and Vaikuntanathan, 2009). In such a case, when the leaked bit number of the secret key exceeds the proportion, the secret key can no longer be used. Contrarily, the continuous leakage attack model allows adversaries to continuously obtain a secret key's partial bits in each usage of the

secret key during the usage life of the associated LR algorithm. Therefore, an LR cryptographic scheme against continuous leakage attacks has the unbounded leakage property and is more suitable for real practical environments (Kiltz and Pietrzak, 2010; Galindo and Virek, 2013).

As mentioned earlier, several LR signcryption schemes under the CL-PKS and the CB-PKS settings have been proposed to remove the key escrow problem. Here, let's review these LR certificateless signcryption (LR-CLSC) (Zhou *et al.*, 2016; Yang *et al.*, 2019) and LR certificate-based signcryption (LR-CBSC) (Zhou *et al.*, 2021) schemes. Zhou *et al.* (2016) adopted a non-interactive zero-knowledge mechanism to propose an LR-CLSC scheme. As we know, the usage of the non-interactive zero-knowledge mechanism is very time-consuming so that Zhou *et al.*'s scheme is unsuitable for mobile devices. Subsequently, Yang *et al.* (2019) presented an improvement on Zhou *et al.*'s scheme to remove the usage of the non-interactive zero-knowledge mechanism to achieve better performance. However, both schemes (Zhou *et al.*, 2016; Yang *et al.*, 2019) are only secure against bounded leakage attacks and cannot resist the attacks of adversaries with continuous leakage abilities.

To achieve continuous leakage-resilient property, Zhou *et al.* (2021) first presented a bounded LR-CBSC scheme and adopted the secret key update method proposed by Dodis *et al.* (2010) to obtain a "conditionally" continuous LR-CBSC (CCLR-CBSC) scheme. That is, by Dodis *et al.*'s secret key update method, a continuous version is constructed from the associated bounded LR cryptographic scheme. However, Kiltz and Pietrzak (2010) have previously shown that Dodis *et al.*'s secret key update method has a shortcoming in the sense that the key update process itself does not allow adversaries to leak any bits of the secret key even if the secret key actually participates in the computation of the key update method. Therefore, Zhou *et al.*'s scheme only possesses the "conditionally" continuous leakage-resilient property.

1.2. *Contributions*

As mentioned earlier, the LR-CLSC schemes in Zhou *et al.* (2016), Yang *et al.* (2019) are only secure against bounded leakage attacks and the CCLR-CBSC scheme in Zhou *et al.* (2021) only possesses the "conditionally" continuous leakage-resilient property. In this paper, a "fully" CLR-CBSC (FCLR-CBSC) scheme is proposed. In our FCLR-CBSC scheme, there are two roles, namely, a trusted certificate authority (CA) and members. A member ID_m sets the associated member secret key MSK_m. The CA uses its own secret key CSK to compute the member's certificate CTF_m using the member ID_m's identity information and public key, and returns it back to the member ID_m. By combining the adversary models of both the LR certificate-based signature (LR-CBS) scheme (Wu *et al.*, 2019) and the LR certificate-based encryption (LR-CBE) scheme (Wu *et al.*, 2020a), we define the adversary model of the FCLR-CBSC scheme. In this adversary model, there are two types of adversaries that include an uncertified member and the honest-but-curious CA.

To realize the "fully" continuous leakage-resilient property, our scheme adopts the key update method proposed by Kiltz and Pietrzak (2010) to update the member ID_m's secret

Table 1
Comparisons between the related schemes and our scheme.

Scheme	Zhou *et al.*'s LR-CLSC scheme (2016)	Yang *et al.*'s LR-CLSC scheme (2019)	Zhou *et al.*'s CCLR-CBSC scheme (2021)	Our proposed FCLR-CBSC scheme
PKS setting	CL-PKS	CL-PKS	CB-PKS	CB-PKS
Leakage of a member's secret key	Allowed	Allowed	Allowed	Allowed
Leakage of the system's secret key	Not allowed	Not allowed	Allowed	Allowed
Leakage model	Bounded	Bounded	Conditionally continuous	Fully continuous

key MSK_m and certificate CTF_m participated in both the signcryption and the unsigncryption algorithms, and the CA's secret key CSK participated in the certificate generation algorithm. In the process of the key update method, these secret keys or certificates are allowed to be leaked by an adversary so that our scheme has the fully continuous leakage-resilient property. Table 1 lists the comparisons between the LR-CLSC schemes in Zhou *et al.* (2016), Yang *et al.* (2019), the CCLR-CBSC scheme in Zhou *et al.* (2021) and our FCLR-CBSC scheme in terms of PKS setting, leakage of a member's secret key, leakage of the system's secret key and leakage model. It is obvious that only our scheme achieves the fully continuous leakage-resilient property and tolerates the leakages of both the system's secret key and a member's secret key. Finally, we employ the security proving method of the generic bilinear group (GBG) model (Boneh *et al.*, 2005) to show that our scheme possesses both authentication and confidentiality against two types of adversaries in the CB-PKS setting. Also, performance analysis and simulation experience demonstrate that our scheme is suited to run on both a PC and a mobile device.

1.3. *Paper Structure*

The remainder of this paper comprises six parts. Four preliminaries are introduced in Section 2. We define a new framework and security model of FCLR-CBSC schemes in Section 3. In Section 4, our FCLR-CBSC scheme is demonstrated. The security analysis of our scheme is given in Section 5. Performance analysis and conclusions are given in Sections 6 and 7, respectively.

2. Preliminaries

2.1. *Bilinear Pairing Set*

Let $\{g_1, g_2, G_1, G_2, p, \hat{e}\}$ be a bilinear pairing set. The reader can refer to Boneh and Franklin (2001) for the parameter selections of the bilinear pairing set. g_1 and g_2 are, respectively, generators of the multiplicative groups G_1 and G_2 with the same prime order p. The bilinear pairing function $\hat{e} : G_1 \times G_1 \rightarrow G_2$ satisfies three properties as presented below:

– *Bilinear property*: $\hat{e}(g_1^x, g_1^y) = \hat{e}(g_1, g_1)^{xy}$, for any $x, y \in Z_p^*$.
– *Non-degenerate property*: $\hat{e}(g_1, g_1) = g_2 \neq 1$.
– *Computable property*: $\hat{e}(X, Y)$ can be effectively computed for any $X, Y \in G_1$.

2.2. *Security Assumptions*

Our proposed FCLR-CBSC scheme is based on two security assumptions as presented below:

– Strong-collision-resistant hash (SCRH) assumption: Let a hash function $H : \{0, 1\}^* \rightarrow \{0, 1\}^l$, l is a large integer, be strong-collision-resistant. Namely, it is difficult to get two different strings $S_1, S_2 \in \{0, 1\}^*$ such that $H(S_1) = H(S_2)$.
– Discrete logarithm (DL) assumption: In the bilinear pairing set $\{g_1, g_2, G_1, G_2, p, \hat{e}\}$ presented earlier, it is difficult to compute $x \in Z_p^*$ for given $g_1^x \in G_1$ or $g_2^x \in G_2$.

2.3. *Generic Bilinear Group Model*

The generic bilinear group (GBG) model (Boneh *et al.*, 2005) is a security proving technique of cryptographic schemes. This GBG model is combined into the security game of a cryptographic scheme. In the security game played by an adversary and a challenger, the challenger first creates a bilinear pairing set $\{g_1, g_2, G_1, G_2, p, \hat{e}\}$. When the adversary performs operations in the bilinear pairing set, it must request the associated queries to the challenger that include the multiplicative query Q_1 of G_1, the multiplicative query Q_2 of G_2 and the bilinear pairing query $Q_{\hat{e}}$. Additionally, the challenger sets two injective random mappings to respectively encode every element of G_1 and G_2 to a distinct bit string, namely, $\zeta_1 : Z_p^* \rightarrow \Psi G_1$ and $\zeta_2 : Z_p^* \rightarrow \Psi G_2$ that satisfy $\Psi G_1 \cap \Psi G_2 = \emptyset$ and $|\Psi G_1| = |\Psi G_2| = p$. The behaviours of three associated queries Q_1, Q_2 and $Q_{\hat{e}}$, for $x, y \in Z_p^*$, are defined as follows:

– $Q_1(\zeta_1(x), \zeta_1(y)) \rightarrow \zeta_1(x + y \bmod p)$.
– $Q_2(\zeta_2(x), \zeta_2(y)) \rightarrow \zeta_2(x + y \bmod p)$.
– $Q_{\hat{e}}(\zeta_1(x), \zeta_1(y)) \rightarrow \zeta_2(x \cdot y \bmod p)$.

Note that $\zeta_1(1)$ and $\zeta_2(1)$ equal g_1 and g_2, respectively. Finally, the adversary would answer the DL problem on G_1/G_2 if it found any collision on G_1/G_2 after finishing the security game.

2.4. *Entropy Evaluation of Secret Keys*

Later, we will employ the entropy evaluation of secret keys with partial leakage to establish the security theorems of the proposed scheme. Here, we first introduce two previous consequences. In 2008, Dodis *et al.* (2008) presented a result (Lemma 1 below) about the entropy evaluation of a secret key K under the leakage function $F(K)$, where $F : K \rightarrow \{0, 1\}^\omega$ and ω is the leakage bit size. Moreover, Galindo and Virek (2013) discussed the entropy evaluation of multiple secret keys to obtain the other result (Lemma 2 below).

Lemma 1. *Let K be a secret key and $F : K \to \{0,1\}^{\omega}$ be its associated leakage function, where ω is the leakage bit size. Under the leakage function F, we have $\widetilde{H}_{\infty}(K|F(K)) \geqq H_{\infty}(K) - \omega$, where H_{∞} and $\widetilde{H}_{\infty}$ denote the min-entropy and the average conditional min-entropy, respectively.*

Lemma 2. *Let $K_1, K_2, \ldots, K_n$ be multiple secret keys participated in a computation formula. Let $MVP \in Z_p[K_1, K_2, \ldots, K_n]$ denote a multiple-variable polynomial that has the degree d. Assume that PD_i denotes the probability distribution of $K_i = k_i \leftarrow Z_p$ under a leakage function F_i with the leakage bit size ω. Thus, we have $H_{\infty}(PD_i) \geqq \log p - \omega$, for $i = 1, 2, \ldots, n$. When all PD_i are mutually independent, we have $\Pr[MVP(K_1 = k_1, K_2 = k_2, \ldots, K_n = k_n) = 0] \leqq (d/p)2^{\omega}$, which is negligible if $\omega < (1-\varepsilon)\log p$, where ε is a positive fraction.*

3. Notations, Framework and Adversary Model

An FCLR-CBSC scheme composes of two roles, namely, a trusted certificate authority (CA) and members. A member ID_m (a sender ID_s or a receiver ID_r) first sets the member's secret key MSK_m and first public key MPK_m, and transmits (ID_m, MPK_m) to the CA. The CA uses a secret key CSK to compute and return the member's certificate CTF_m and second public key UPK_m to the member ID_m via a secure channel. By taking as input a message msg and the public key pair (MPK_r, UPK_r) of the receiver ID_r, the sender ID_s uses her/his certificate and secret key to compute a ciphertext CT and send (ID_s, CT) to the receiver ID_r. By taking as input a ciphertext CT and the public key pair (MPK_s, UPK_s) of the sender ID_s, the receiver ID_r returns msg if CT is "Valid"; otherwise returns "Invalid". The system model of the FCLR-CBSC scheme is depicted in Fig. 1.

To achieve fully continuous leakage-resilient property (Kiltz and Pietrzak, 2010; Galindo and Virek, 2013), every secret key or certificate in the system is partitioned into two parts, which must be updated before being participated in each computation round. Assume that the CA's secret key CSK is initially partitioned into the beginning secret key pair $(CSK_{1,0}, CSK_{2,0})$. Additionally, let the CA's current secret key pair be $(CSK_{1,i-1}, CSK_{2,i-1})$, which must be updated to $(CSK_{1,i}, CSK_{2,i})$ when it participates in the i-th invocation of the certificate generation algorithm. Note that we have $CSK = CSK_{1,0} \cdot CSK_{2,0} = \cdots = CSK_{1,i-1} \cdot CSK_{2,i-1} = CSK_{1,i} \cdot CSK_{2,i}$. For the same reason, the member ID_m's secret key MSK_m and certificate CTF_m are initially partitioned into the beginning secret key pair $(MSK_{m,1,0}, MSK_{m,2,0})$ and the certificate pair $(CTF_{m,1,0}, CTF_{m,2,0})$, respectively. Let the member ID_m's current secret key and certificate pairs be, respectively, $(CTF_{m,1,j-1}, CTF_{m,2,j-1})$ and $(MSK_{m,1,j-1}, MSK_{m,2,j-1})$, which must be updated to $(CTF_{m,1,j}, CTF_{m,2,j})$ and $(MSK_{m,1,j}, MSK_{m,2,j})$ when they participate in the j-th invocation of the signcryption or unsigncryption algorithm. Note that we have $MSK_m = MSK_{m,1,0} \cdot MSK_{m,2,0} = \cdots = MSK_{m,1,j-1} \cdot MSK_{m,2,j-1} = MSK_{m,1,j} \cdot MSK_{m,2,j}$ and $CTF_m = CTF_{m,1,0} \cdot CTF_{m,2,0} = \cdots = CTF_{m,1,j-1} \cdot CTF_{m,2,j-1} = CTF_{m,1,j} \cdot CTF_{m,2,j}$.

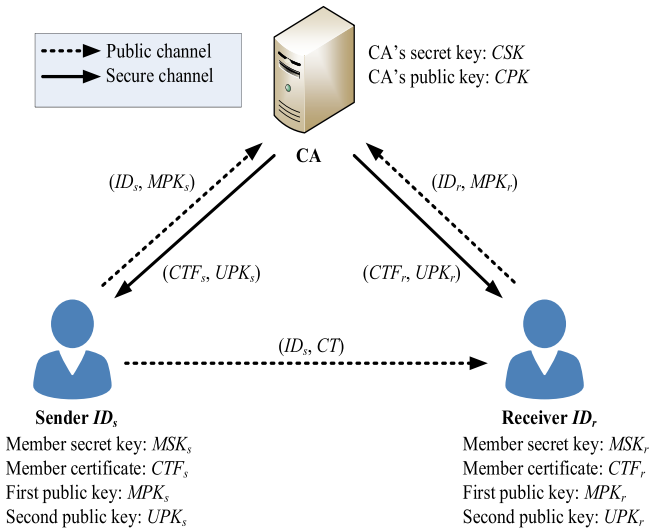

Fig. 1. The system model of an FCLR-CBSC scheme.

In the following, we first present some symbols and notations used in the proposed FCLR-CBSC scheme. Subsequently, a new framework and adversary model of FCLR-CBSC schemes are defined.

3.1. *Symbols and Notations*

For reference, the symbols and notations used in the proposed scheme are summarized in Table 2.

3.2. *Framework*

Based on the frameworks of both the LR-CBS (Wu *et al.*, 2019) and the LR-CBE (Wu *et al.*, 2020a) schemes, a new framework of FCLR-CBSC schemes is defined as follows.

Definition 1. An FCLR-CBSC scheme comprises five algorithms as presented below:

- *Initialization*: The CA runs this algorithm to compute the CA's secret key CSK and system public key CPK while generating and publishing a public parameter set PPS. Additionally, the CA also partitions CSK to set the beginning secret key pair $(CSK_{1,0}, CSK_{2,0})$.
- *Member secret key generation*: A member ID_m (a sender ID_s or a receiver ID_r) runs this algorithm to compute her/his secret key MSK_m and first public key MPK_m. Additionally, the member ID_m partitions MSK_m to set the beginning secret key pair $(MSK_{m,1,0}, MSK_{m,2,0})$. Finally, the member ID_m sends (ID_m, MPK_m) to the CA.
- *Certificate generation*: Assume that the CA's current secret key pair is $(CSK_{1,i-1}, CSK_{2,i-1})$. The CA first obtains a new current secret key pair $(CSK_{1,i}, CSK_{2,i})$ by up-

Table 2
Symbols and notations.

Symbols/notations	Meanings
PPS	a public parameter set
CPK	the CA's public key
CSK	the CA's secret key
$(CSK_{1,0}, CSK_{2,0})$	the CA's beginning secret key pair
$(CSK_{1,i}, CSK_{2,i})$	the CA's i-th secret key pair
ID_m	the identity of a member (a sender ID_s or a receiver ID_r)
(MPK_m, UPK_m)	the public key pair of ID_m
MSK_m	the certificate of ID_m
$(MSK_{m,1,0}, MSK_{m,2,0})$	the beginning secret key pair of ID_m
$(MSK_{m,1,j}, MSK_{m,2,j})$	the j-th secret key pair of ID_m
CTF_m	the certificate of ID_m
$(CTF_{m,1,0}, CTF_{m,2,0})$	the beginning certificate pair of ID_m
$(CTF_{m,1,j}, CTF_{m,2,j})$	the j-th certificate pair of ID_m
$H()$	a hash function
$SKE()/SKD()$	symmetric-key encryption/decryption functions
ID_s	the identity of a sender (it is also a member)
ID_r	the identity of a receiver (it is also a member)
msg	a message
CT	a ciphertext

dating $(CSK_{1,i-1}, CSK_{2,i-1})$. Upon receiving (ID_m, MPK_m) from the member ID_m, the CA creates and returns the certificate CTF_m and second public key UPK_m to the member ID_m. Upon receiving (CTF_m, UPK_m), the member ID_m sets her/his beginning certificate pair $(CTF_{m,1,0}, CTF_{m,2,0})$ and public key pair (MPK_m, UPK_m).

– *Signcryption*: Assume that the sender ID_s's current certificate and secret key pairs are, respectively, $(CTF_{s,1,j-1}, CTF_{s,2,j-1})$ and $(MSK_{s,1,j-1}, MSK_{s,2,j-1})$. The sender ID_s first obtains a new current certificate pair $(CTF_{s,1,j}, CTF_{s,2,j})$ and secret key pair $(MSK_{s,1,j}, MSK_{s,2,j})$ by updating $(CTF_{s,1,j-1}, CTF_{s,2,j-1})$ and $(MSK_{s,1,j-1}, MSK_{s,2,j-1})$, respectively. By taking as input a message msg and the public key pair (MPK_r, UPK_r) of the receiver ID_r, the sender ID_s runs this algorithm to return a ciphertext CT.
– *Unsigncryption*: Assume that the receiver ID_r's current certificate and secret key pairs are, respectively, $(CTF_{r,1,k-1}, CTF_{r,2,k-1})$ and $(MSK_{r,1,k-1}, MSK_{r,2,k-1})$. The receiver ID_r first obtains a new current certificate pair $(CTF_{r,1,k}, CTF_{r,2,k})$ and secret key pair $(MSK_{r,1,k}, MSK_{r,2,k})$ by updating $(CTF_{r,1,k-1}, CTF_{r,2,k-1})$ and $(MSK_{r,1,k-1}, MSK_{r,2,k-1})$, respectively. By taking as input a ciphertext CT and the public key pair (MPK_s, UPK_s) of the sender ID_s, the receiver ID_r returns msg if CT is "Valid"; otherwise returns "Invalid".

3.3. *Adversary Model*

Here, six continuous leakage functions $f_{CA,i}, h_{CA,i}, f_{SC,j}, h_{SC,j}, f_{US,k}$ and $h_{US,k}$ are used to simulate the leakage abilities of adversaries. In the i-th invocation of the *Certificate generation* algorithm, an adversary could obtain partial bits of the CA's current secret key pair

$(CSK_{1,i}, CSK_{2,i})$ by $f_{CA,i}$ and $h_{CA,i}$. In the j-th invocation of the *Signcryption* algorithm, an adversary could obtain partial bits of the sender ID_s's current certificate pair $(CTF_{s,1,j}, CTF_{s,2,j})$ and secret key pair $(MSK_{s,1,j}, MSK_{s,2,j})$ by $f_{SC,j}$ and $h_{SC,j}$. In the k-th invocation of the *Unsigncryption* algorithm, an adversary could obtain partial bits of the receiver ID_r's current certificate pair $(CTF_{r,1,k}, CTF_{r,2,k})$ and secret key pair $(MSK_{r,1,k}, MSK_{r,2,k})$ by $f_{US,k}$ and $h_{US,k}$. Let ω be the maximal leakage bit length for each leakage function. Let $\Delta f_{CA,i}$, $\Delta h_{CA,i}$, $\Delta f_{SC,j}$, $\Delta h_{SC,j}$, $\Delta f_{US,k}$ and $\Delta h_{US,k}$, respectively, denote their outputs of the six leakage functions. Therefore, we have $|\Delta f_{CA,i}|$, $|\Delta h_{CA,i}|$, $|\Delta f_{SC,j}|$, $|\Delta h_{SC,j}|$, $|\Delta f_{US,k}|$, $|\Delta h_{US,k}| \leqq \omega$ and their inputs/outputs are presented as follows:

- $\Delta f_{CA,i} = f_{CA,i}(CSK_{1,i})$.
- $\Delta h_{CA,i} = h_{CA,i}(CSK_{2,i})$.
- $\Delta f_{SC,j} = f_{SC,j}(CTF_{s,1,j}, MSK_{s,1,j})$.
- $\Delta h_{SC,j} = h_{SC,j}(CTF_{s,2,j}, MSK_{s,2,j})$.
- $\Delta f_{US,k} = f_{US,k}(CTF_{r,1,k}, MSK_{r,1,k})$.
- $\Delta h_{US,k} = h_{US,k}(CTF_{r,2,k}, MSK_{r,2,k})$.

Based on the adversary models of both the LR-CBS (Wu *et al.*, 2019) and the LR-CBE (Wu *et al.*, 2020a) schemes, we define a new adversary model of FCLR-CBSC schemes. In the new adversary model, there are two types of adversaries (A_I and A_{II}) as presented below:

- A_I: A_I simulates an "uncertified member" who can set any member ID_m's secret key MSK_m and first public key MPK_m, but can obtain neither ID_m's certificate CTF_m nor second public key UPK_m. Indeed, A_I can get partial bits of the CA's current secret key pair $(CSK_{1,i}, CSK_{2,i})$ in the i-th invocation of the *Certificate generation* algorithm. Also, A_I could obtain partial bits of the sender ID_s's current certificate pair $(CTF_{s,1,j}, CTF_{s,2,j})$ in the j-th invocation of the *Signcryption* algorithm. In the k-th invocation of the *Unsigncryption* algorithm, A_I could obtain partial bits of the receiver ID_r's current certificate pair $(CTF_{r,1,k}, CTF_{r,2,k})$.
- A_{II}: A_{II} simulates an "honest-but-curious CA" who has the CA's secret key CSK and produces any member ID_m's certificate CTF_m and second public key UPK_m, but A_{II} can obtain neither the member ID_m's secret key MSK_m nor first public key MPK_m. Also, A_{II} could obtain partial bits of the sender ID_s's current secret key pair $(MSK_{s,1,j}, MSK_{s,2,j})$ in the j-th invocation of the *Signcryption* algorithm. In the k-th invocation of the *Unsigncryption* algorithm, A_{II} could obtain partial bits of the receiver ID_r's current secret key pair $(MSK_{r,1,k}, MSK_{r,2,k})$.

An FCLR-CBSC scheme must possess two security properties, namely, authentication of signing process and confidentiality of encrypting process, that are modelled by two security games as defined below.

Definition 2 ($\mathbf{G_{auth}}$). The authentication property is modelled by the security game G_{auth} that is played by an adversary A (A_I or A_{II}) and a challenger B. We say that an FCLR-CBSC scheme has existential unforgeability against both continuous leakage and

adaptive chosen-message attacks (EXUF-CLRACMA) if no probabilistic polynomial time (PPT) adversary A has a non-negligible advantage to win the following game G_{auth}.

- *Setup*. The challenger B runs the *Initialization* algorithm in Definition 1 to get the CA's secret key CSK and public key CPK while generating and publishing a public parameter set PPS. Also, B partitions CSK to set the beginning secret key pair $(CSK_{1,0}, CSK_{2,0})$. Additionally, if A is of type A_{II}, B sends CSK to A_{II}.
- *Queries*. A may adaptively request the following queries to B at most η times.
  - *Member key generation query* (ID_m): B produces and returns the member ID_m's secret key MSK_m and first public key MPK_m.
  - *Member secret key query* (ID_m): B returns the member ID_m's secret key MSK_m.
  - *Certificate generation query* (ID_m, MPK_m): B returns the member ID_m's certificate CTF_m and second public key UPK_m.
  - *Certificate generation leak query* $(i, f_{CA,i}, h_{CA,i})$: A may request this query only once. B returns $\Delta f_{CA,i} = f_{CA,i}(CSK_{1,i})$ and $\Delta h_{CA,i} = h_{CA,i}(CSK_{2,i})$.
  - *Public key retrieve query* (ID_m): B returns the member ID_m's public key pair (MPK_m, UPK_m).
  - *Public key replace query* $(ID_m, (MPK'_m, UPK'_m))$: B records the replacement.
  - *Signcryption query* (msg, ID_s, ID_r): The sender ID_s first obtains a new current certificate pair $(CTF_{s,1,j}, CTF_{s,2,j})$ and secret key pair $(MSK_{s,1,j}, MSK_{s,2,j})$ by updating $(CTF_{s,1,j-1}, CTF_{s,2,j-1})$ and $(MSK_{s,1,j-1}, MSK_{s,2,j-1})$, respectively. B returns a ciphertext CT.
  - *Signcryption leak query* $(ID_s, j, f_{SC,j}, h_{SC,j})$: A may request this query only once. B returns $\Delta f_{SC,j} = f_{SC,j}(CTF_{s,1,j}, MSK_{s,1,j})$ and $\Delta h_{SC,j} = h_{SC,j}(CTF_{s,2,j}, MSK_{s,2,j})$.
  - *Unsigncryption query* (CT, ID_r): The receiver ID_r first obtains a new current certificate pair $(CTF_{r,1,k}, CTF_{r,2,k})$ and secret key pair $(MSK_{r,1,k}, MSK_{r,2,k})$ by updating $(CTF_{r,1,k-1}, CTF_{r,2,k-1})$ and $(MSK_{r,1,k-1}, MSK_{r,2,k-1})$, respectively. B returns the message msg.
  - *Unsigncryption leak query* $(ID_r, k, f_{US,k}, h_{US,k})$: A may request this query only once. B returns $\Delta f_{US,k} = f_{US,k}(CTF_{r,1,k}, MSK_{r,1,k})$ and $\Delta h_{US,k} = h_{US,k}(CTF_{r,2,k}, MSK_{r,2,k})$.
- *Forgery*. A creates a tuple $(msg', CT' = (\sigma', C', U', ID'_s, ID'_r))$. It is said that A wins G_{auth} if the following four conditions hold.
  (1) For $(msg', CT' = (\sigma', C', U', ID'_s, ID'_r))$, the *Unsigncryption* algorithm returns "Valid".
  (2) The *Signcryption query* (msg', ID'_s, ID'_r) has never been requested.
  (3) If A is of type A_I, the *Certificate generation query* (ID'_s, MPK'_s) has never been requested.
  (4) If A is of type A_{II}, neither the *Member secret key query* (ID'_s), nor the *Public key replace query* $(ID'_s, (MPK'_s, UPK'_s))$ have never been requested.

Definition 3 ($\mathbf{G_{conf}}$). The confidentiality property is modelled by the security game G_{conf} that is played by an adversary A (A_I or A_{II}) and a challenger B. We say that

an FCLR-CBSC scheme has indistinguishability of encryptions against both continuous leakage and chosen-ciphertext attacks (INDEN-CLCCA) if no PPT adversary A has a non-negligible advantage to win the following game G_{conf}.

- *Setup*. It is identical to the *Setup* in Definition 2.
- *Queries*. It is identical to the *Queries* in Definition 2.
- *Challenge*. A sends a receiver's identity ID'_r and a message pair (msg'_0, msg'_1) to B. Then B randomly chooses a bit $b' \in \{0, 1\}$ and runs the *Signcryption* algorithm with (msg'_b, ID_s, ID'_r) to create and return a ciphertext $CT = (\sigma, C, U, ID_s, ID'_r)$ to A. Additionally, the following two conditions must hold.
  (1) If A is of type A_I, the *Certificate generation query* (ID'_r, MPK'_r) has never been requested.
  (2) If A is of type A_{II}, neither the *Member secret key query* (ID'_r), nor the *Public key replace query* $(ID'_r, (MPK'_r, UPK'_r))$ have been requested.
- *Guess*. A returns a bit $b \in \{0, 1\}$. If $b = b'$, we say that A wins G_{conf} and its advantage is defined as $\mathrm{Adv}_A = |\Pr[b = b'] - 1/2|$.

4. The Proposed FCLR-CBSC Scheme

By the framework defined in Section 3, a fully continuous leakage-resilient certificate-based signcryption (FCLR-CBSC) scheme is proposed below that comprises five algorithms.

- **Initialization:** The CA first create a bilinear pairing set $\{g_1, g_2, G_1, G_2, p, \hat{e}\}$ described in Section 2. By running the following procedure, the CA computes her/his secret key CSK and public key CPK while generating and publishing a public parameter set PPS.
  (1) Choose a random value $s \in Z_p^*$, and compute the CA's secret key $CSK = g_1^s$ and public key $CPK = \hat{e}(CSK, g_1)$.
  (2) Choose a random value $t \in Z_p^*$, and set the CA's beginning secret key pair $(CSK_{1,0}, CSK_{2,0}) = (CSK \cdot g_1^{-t}, g_1^t)$, where $CSK = CSK_{1,0} \cdot CSK_{2,0}$ and the public key CPK is kept unchanged.
  (3) Choose four random values $w, x, y, z \in Z_p^*$, and compute $W = g_1^w$, $X = g_1^x$, $Y = g_1^y$ and $Z = g_1^z$.
  (4) Pick symmetric-key encryption and decryption functions, denoted by SKE and SKD.
  (5) Pick a hash function H: $\{0, 1\}^* \times G_1 \rightarrow \{0, 1\}^l$, where l is a large integer.
  (6) Publish $PPS = \{g_1, g_2, G_1, G_2, p, \hat{e}, CPK, W, X, Y, Z, SKE, SKD, H\}$.
- **Member secret key generation:** A member ID_m (a sender ID_s or a receiver ID_r) first chooses a random value $a \in Z_p^*$, and computes the member's secret key $MSK_m = g_1^a$ and first public key $MPK_m = \hat{e}(MSK_m, g_1)$. The member chooses a random value $b \in Z_p^*$ and computes the member ID_m's beginning secret key pair $(MSK_{m,1,0}, MSK_{m,2,0}) = (MSK_m \cdot g_1^{-b}, g_1^b)$, where $MSK_m = MSK_{m,1,0} \cdot MSK_{m,2,0}$. Meanwhile, the member ID_m sends (ID_m, MPK_m) to the CA.

- **Certificate generation:** Assume that the CA's current secret key pair is $(CSK_{1,i-1}, CSK_{2,i-1})$. Upon receiving (ID_m, MPK_m) from the member ID_m, the CA runs the following procedure.
  (1) Choose a random value $u \in Z_p^*$, and update the CA's current secret key pair as $(CSK_{1,i}, CSK_{2,i}) = (CSK_{1,i-1} \cdot g_1^{-u}, CSK_{2,i-1} \cdot g_1^u)$, where $CSK = CSK_{1,i} \cdot CSK_{2,i} = CSK_{1,i-1} \cdot CSK_{2,i-1}$ and the public key CPK is kept unchanged.
  (2) Choose a random value $v \in Z_p^*$, compute the member ID_m's second public key $UPK_m = g_1^v$ and set the member ID_m's public key pair (MPK_m, UPK_m).
  (3) Set $\alpha = ID_m || MPK_m || UPK_m$, and compute a temporary value $TV_m = CSK_{1,i} \cdot (W \cdot X^\alpha)^v$ and the member ID_m's certificate $CTF_m = CSK_{2,i} \cdot TV_m$.
  (4) Return CTF_m to the member ID_m via a secure channel.
  Upon receiving CTF_m, the member ID_m chooses a random value $c \in Z_p^*$ and sets her/his beginning certificate pair $(CTF_{m,1,0}, CTF_{m,2,0}) = (CTF_m \cdot g_1^{-c}, g_1^c)$.
- **Signcryption:** Assume that the sender ID_s's current certificate and secret key pairs are, respectively, $(CTF_{s,1,j-1}, CTF_{s,2,j-1})$ and $(MSK_{s,1,j-1}, MSK_{s,2,j-1})$. The sender ID_s would like to signcrypt a message msg to the receiver ID_r with public key pair (MPK_r, UPK_r) by running the following procedure.
  (1) Choose a random value $d \in Z_p^*$, and update the two pairs above as $(CTF_{s,1,j}, CTF_{s,2,j}) = (CTF_{s,1,j-1} \cdot g_1^d, CTF_{s,2,j-1} \cdot g_1^{-d})$ and $(MSK_{s,1,j}, MSK_{s,2,j}) = (MSK_{s,1,j-1} \cdot g_1^d, MSK_{s,2,j-1} \cdot g_1^{-d})$, where $CTF_s = CTF_{s,1,j} \cdot CTF_{s,2,j} = CTF_{s,1,j-1} \cdot CTF_{s,2,j-1}$ and $MSK_s = MSK_{s,1,j} \cdot MSK_{s,2,j} = MSK_{s,1,j-1} \cdot MSK_{s,2,j-1}$. Note that the member ID_s's public key pair (MPK_s, UPK_s) is kept unchanged.
  (2) Set $\alpha = ID_r || MPK_r || UPK_r$, select a random value $\beta \in Z_p^*$, and compute $U = g_1^\beta$, $K_1 = (MPK_r)^\beta$ and $K_2 = (CPK \cdot \hat{e}(UPK_r, W \cdot X^\alpha))^\beta$.
  (3) Set an encryption key $K = K_1 \oplus K_2$, and compute $C = SKE_K(msg)$ and $\delta = H(msg, C, U, ID_s, ID_r)$.
  (4) Compute a temporary value $TV_S = CTF_{s,1,j} \cdot MSK_{s,1,j} \cdot (Y \cdot Z^\delta)^\beta$.
  (5) Compute a signature $\sigma = CTF_{s,2,j} \cdot MSK_{s,2,j} \cdot TV_S$.
  (6) Produce a ciphertext $CT = (\sigma, C, U, ID_s, ID_r)$.
- **Unsigncryption**: Assume that the receiver ID_r's current certificate and secret key pairs are, respectively, $(CTF_{r,1,k-1}, CTF_{r,2,k-1})$ and $(MSK_{r,1,k-1}, MSK_{r,2,k-1})$. Given $CT = (\sigma, C, U, ID_s, ID_r)$, the receiver ID_r unsigncrypts CT to obtain the message msg and verify the signature σ by running the following procedure.
  (1) Choose a random value $f \in Z_p^*$, and update the two pairs above as $(CTF_{r,1,k}, CTF_{r,2,k}) = (CTF_{r,1,k-1} \cdot g_1^f, CTF_{r,2,k-1} \cdot g_1^{-f})$ and $(MSK_{r,1,k}, MSK_{r,2,k}) = (MSK_{r,1,k-1} \cdot g_1^f, MSK_{r,2,k-1} \cdot g_1^{-f})$, where $CTF_r = CTF_{r,1,k} \cdot CTF_{r,2,k} = CTF_{r,1,k-1} \cdot CTF_{r,2,k-1}$ and $MSK_r = MSK_{r,1,k} \cdot MSK_{r,2,k} = MSK_{r,1,k-1} \cdot MSK_{r,2,k-1}$. Note that the member ID_r's public key pair (MPK_r, UPK_r) is kept unchanged.
  (2) Compute two temporary values $TV_{K1} = \hat{e}(U, MSK_{r,1,k})$ and $TV_{K2} = \hat{e}(U, CTF_{r,1,k})$.
  (3) Compute $K_1' = TV_{K1} \cdot \hat{e}(U, MSK_{r,2,k})$ and $K_2' = TV_{K2} \cdot \hat{e}(U, CTF_{r,2,k})$.

(4) Set $K' = K'_1 \oplus K'_2$, and decrypt the message $msg = SKD'_K(C)$.
(5) Compute $\delta = H(msg, C, U, ID_s, ID_r)$, and set $\alpha = ID_r || MPK_r || UPK_r$.
(6) Verify the equality $\hat{e}(g_1, \sigma) = CPK \cdot MPK_s \cdot \hat{e}(UPK_s, W \cdot X^{\alpha}) \cdot \hat{e}(U, Y \cdot Z^{\delta})$. If the equality holds, return *msg* and "Valid"; otherwise return "Invalid".

We can arrive at $K' = K'_1 \oplus K'_2 = K_1 \oplus K_2 = K$ and $\hat{e}(g_1, \sigma) = CPK \cdot MPK_s \cdot \hat{e}(UPK_s, W \cdot X^{\alpha}) \cdot \hat{e}(U, Y \cdot Z^{\delta})$ by the following equalities.

(1) $K'_1 = TV_{K1} \cdot \hat{e}(U, MSK_{r,2,k}) = \hat{e}(U, MSK_{r,1,k}) \cdot \hat{e}(U, MSK_{r,2,k}) = \hat{e}(U, MSK_{r,1,k} \cdot MSK_{r,2,k}) = \hat{e}(U, MSK_r) = \hat{e}(g_1^{\beta}, MSK_r) = \hat{e}(MSK_r, g_1)^{\beta} = (MPK_r)^{\beta} = K_1$.
(2) $K'_2 = TV_{K2} \cdot \hat{e}(U, CTF_{r,2,k}) = \hat{e}(U, CTF_{r,1,k}) \cdot \hat{e}(U, CTF_{r,2,k}) = \hat{e}(U, CTF_{r,1,k} \cdot CTF_{r,2,k}) = \hat{e}(U, CTF_r) = \hat{e}(g_1^{\beta}, CSK \cdot (W \cdot X^{\alpha})^{v}) = \hat{e}(g_1, CSK \cdot (W \cdot X^{\alpha})^{v})^{\beta} = \hat{e}(g_1, CSK) \cdot \hat{e}(g_1, (W \cdot X^{\alpha})^{v})^{\beta} = (CPK \cdot \hat{e}(g_1^{v}, (W \cdot X^{\alpha}))^{\beta} = (CPK \cdot \hat{e}(UPK_r, W \cdot X^{\alpha}))^{\beta} = K_2$.
(3) $\hat{e}(g_1, \sigma) = \hat{e}(g_1, CTF_{s,2,j} \cdot MSK_{s,2,j} \cdot TV_S) = \hat{e}(g_1, CTF_{s,2,j} \cdot MSK_{s,2,j} \cdot CTF_{s,1,j} \cdot MSK_{s,1,j} \cdot (Y \cdot Z^{\delta})^{\beta}) = \hat{e}(g_1, CTF_{s,1,j} \cdot CTF_{s,2,j} \cdot MSK_{s,1,j} \cdot MSK_{s,2,j} \cdot (Y \cdot Z^{\delta})^{\beta}) = \hat{e}(g_1, CTF_s \cdot MSK_s \cdot (Y \cdot Z^{\delta})^{\beta}) = \hat{e}(g_1, CSK \cdot (W \cdot X^{\alpha})^{v} \cdot MSK_s \cdot (Y \cdot Z^{\delta})^{\beta}) = \hat{e}(g_1, CSK) \cdot \hat{e}(g_1, (W \cdot X^{\alpha})^{v}) \cdot \hat{e}(g_1, MSK_s) \cdot \hat{e}(g_1, (Y \cdot Z^{\delta})^{\beta}) = CPK \cdot \hat{e}(g_1^{v}, (W \cdot X^{\alpha})) \cdot MPK_s \cdot \hat{e}(g_1^{\beta}, (Y \cdot Z^{\delta})) = CPK \cdot MPK_s \cdot \hat{e}(UPK_s, W \cdot X^{\alpha}) \cdot \hat{e}(U, Y \cdot Z^{\delta})$.

5. Security Analysis

An FCLR-CBSC scheme must possess two security properties, namely, authentication of signing process and confidentiality of encrypting process, that are modelled by two security games G_{auth} and G_{conf} defined in Section 3.3. Both games are played by an adversary A (A_I or A_{II}) and a challenger B. Theorems 1 and 2 below show that our FCLR-CBSC scheme is EXUF-CLRACMA-secure against A_I and A_{II} in G_{auth}, respectively. Also, Theorems 3 and 4 show that the FCLR-CBSC scheme is INDEN-CLCCA-secure against A_I and A_{II} in G_{conf}, respectively.

Theorem 1. *Under the SCRH and DL assumptions in the GBG model, our FCLR-CBSC scheme is EXUF-CLRACMA-secure against A_I in G_{auth}.*

Proof. A_I and B play G_{auth} that comprises three phases as presented below.

– *Setup*. B runs the *Initialization* algorithm of the FCLR-CBSC scheme to create the CA's secret key *CSK* and public key *CPK* while setting the public parameter set $PPS = \{g_1, g_2, G_1, G_2, p, \hat{e}, CPK, W, X, Y, Z, SKE, SKD, H\}$. B also establishes six lists L_1, L_2, L_{MS}, L_{MC}, L_{SC} and L_H as defined below.
- L_1 is used to record all elements $(\Theta G_{1,a,b,c}, \Psi G_{1,a,b,c})$ of G_1, where $\Theta G_{1,a,b,c}$ and $\Psi G_{1,a,b,c}$ denote a multivariate polynomial and its associated bit string, respectively. The indices a, b, c mean the query-type a, b-th query and c-th element, respectively. Initially, six elements $(\Theta g_1, \Psi G_{1,s,0,1})$, $(\Theta CSK, \Psi G_{1,s,0,2})$, $(\Theta W, \Psi G_{1,s,0,3})$, $(\Theta X, \Psi G_{1,s,0,4})$, $(\Theta Y, \Psi G_{1,s,0,5})$ and $(\Theta Z, \Psi G_{1,s,0,6})$ are put in L_1.

- L_2 is used to record all elements ($\Theta G_{2,a,b,c}$, $\Psi G_{2,a,b,c}$) of G_2, where $\Theta G_{2,a,b,c}$ and $\Psi G_{2,a,b,c}$ denote a multivariate polynomial and the associated bit string, respectively. The indices a, b, c have the identical meanings as in L_1. Initially, two elements (Θg_2, $\Psi G_{2,s,0,1}$) and (ΘCPK, $\Psi G_{2,s,0,2}$) are put in L_2.

Note that A_I can apply the following two rules to make converting between a bit string $\Psi G_{1,a,b,c}/\Psi G_{2,a,b,c}$ and a multivariate polynomial $\Theta G_{1,a,b,c}/\Theta G_{2,a,b,c}$.

(1) Converting-1 ($\Theta G_{1,a,b,c}/\Theta G_{2,a,b,c}$): B returns $\Psi G_{1,a,b,c}/\Psi G_{2,a,b,c}$ if $\Theta G_{1,a,b,c}/\Theta G_{2,a,b,c}$ is found in L_1/L_2. Otherwise, B chooses a distinct bit string $\Psi G_{1,a,b,c}/\Psi G_{2,a,b,c}$ and puts $(\Theta G_{1,a,b,c}, \Psi G_{1,a,b,c})/(\Theta G_{2,a,b,c}, \Psi G_{2,a,b,c})$ in L_1/L_2.

(2) Converting-2 ($\Psi G_{1,a,b,c}/\Psi G_{2,a,b,c}$): B returns $\Theta G_{1,a,b,c}/\Theta G_{2,a,b,c}$ if $\Psi G_{1,a,b,c}/\Psi G_{2,a,b,c}$ is found in L_1/L_2. Otherwise, B terminates the game.

- L_{MS} is used to record a member ID_m's secret key MSK_m and her/his first public key MPK_m by (ID_m, ΘMSK_m, ΘMPK_m, *replace*), for $m = 1, \ldots, n$. Initially, B sets $replace = 0$ to indicate that the member's public key has never been replaced by the adversary.
- L_{MC} is used to record a member ID_m's certificate CTF_m and her/his second public key UPK_m by (ID_m, ΘCTF_m, ΘUPK_m, *replace*), for $m = 1, \ldots, n$. Also, B sets $replace = 0$.
- L_{SC} is used to record the content of running the *Signcryption* algorithm by (ID_s, ID_r, msg, ΘU, ΘK_1, ΘK_2, $\Theta\sigma$, C, $\Theta\delta$).
- L_H is used to record the input/output of the hash function H by ($msg||C||\Psi U||ID_s||ID_r$, $\Psi\delta$).

– *Query*: A_I may issue the following queries to B at most η times.

- Q_1 *query* ($\Psi G_{1,q,k,r}$, $\Psi G_{1,q,k,s}$, *ORER*): For the k-th Q_1, B runs the following procedure.
  (1) Convert ($\Psi G_{1,q,k,r}$, $\Psi G_{1,q,k,s}$) to ($\Theta G_{1,q,k,r}$, $\Theta G_{1,q,k,s}$) in L_1.
  (2) Set $\Theta G_{1,q,k,t} = \Theta G_{1,q,k,r} + \Theta G_{1,q,k,s}$ if $ORER =$ "$\times$", and $\Theta G_{1,q,k,t} = \Theta G_{1,q,k,r} - \Theta G_{1,q,k,s}$ if $ORER =$ "/".
  (3) Convert $\Theta G_{1,q,k,t}$ in L_1 to return $\Psi G_{1,q,k,t}$.
- Q_2 *query* ($\Psi G_{2,q,k,r}$, $\Psi G_{2,q,k,s}$, *ORER*): For the k-th Q_2, B runs the following procedure.
  (1) Convert ($\Psi G_{2,q,k,r}$, $\Psi G_{2,q,k,s}$) to ($\Theta G_{2,q,k,r}$, $\Theta G_{2,q,k,s}$) in L_2.
  (2) Set $\Theta G_{2,q,k,t} = \Theta G_{2,q,k,r} + \Theta G_{2,q,k,s}$ if $ORER =$ "$\times$", and $\Theta G_{2,q,k,t} = \Theta G_{2,q,k,r} - \Theta G_{2,q,k,s}$ if $ORER =$ "/".
  (3) Convert $\Theta G_{2,q,k,t}$ in L_2 to return $\Psi G_{2,q,k,t}$.
- $Q_{\hat{e}}$ *query* ($\Psi G_{1,q,k,r}$, $\Psi G_{1,q,k,s}$): For the k-th $Q_{\hat{e}}$, B runs the following procedure.
  (1) Convert ($\Psi G_{1,q,k,r}$, $\Psi G_{1,q,k,s}$) to ($\Theta G_{1,q,k,r}$, $\Theta G_{1,q,k,s}$) in L_1.
  (2) Set $\Theta G_{2,q,k,t} = \Theta G_{1,q,k,r} \cdot \Theta G_{1,q,k,s}$.
  (3) Convert $\Theta G_{2,q,k,t}$ in L_2 to return $\Psi G_{2,q,k,t}$.
- *Member key generation query* (ID_m): B produces the member ID_m's secret key ΘMSK_m and first public key ΘMPK_m. B then puts (ID_m, ΘMSK_m, ΘMPK_m, 0) in L_{MS}. Also, B converts (ΘMSK_m, ΘMPK_m) in L_1 to return (ΨMSK_m, ΨMPK_m).

- *Member secret key query* (ID_m): If (ID_m, ΘMSK_m, ΘMPK_m, 0) in L_{MS} is found, B converts ΘMSK_m in L_1 to return ΨMSK_m. Otherwise, B issues the *Member key generation query* (ID_m) to return ΨMSK_m.
- *Certificate generation query* (ID_m, ΨMPK_m): Assume that the CA has the i-th secret key pair ($\Theta CSK_{1,i}$, $\Theta CSK_{2,i}$). B runs the following procedure.
  (1) Pick a new variate ΘUPK_m in G_1 as the second public key of the member ID_m.
  (2) Pick a new variate $\Theta\alpha$ in G_1 and put ($\Theta\alpha$, $\Psi\alpha = ID_m || \Psi MPK_m || \Psi UPK_m$) in L_1.
  (3) Set $\Theta CTF_m = \Theta CSK + (\Theta W + \Theta X \cdot \Theta\alpha) \cdot \Theta UPK_m$ and put (ID_m, ΘCTF_m, ΘUPK_m, 0) in L_{MC}.
  (4) Convert (ΘCTF_m, ΘUPK_m) in L_1 to return (ΨCTF_m, ΨUPK_m).
- *Certificate generation leak query* (i, $f_{CA,i}$, $h_{CA,i}$): A_I can issue this query only once for the CA's i-th secret key pair ($\Theta CSK_{1,i}$, $\Theta CSK_{2,i}$). B returns $\Delta f_{CA,i} = f_{CA,i}(\Theta CSK_{1,i})$ and $\Delta h_{CA,i} = h_{CA,i}(\Theta CSK_{2,i})$.
- *Public key retrieve query* (ID_m): B finds ID_m's public key pair (ΘMPK_m, ΘUPK_m) by searching (ID_m, ΘMSK_m, ΘMPK_m, 0) in L_{MS} and (ID_m, ΘCTF_m, ΘUPK_m, 0) in L_{MC}. B converts (ΘMPK_m, ΘUPK_m) in L_1 to return (ΨMPK_m, ΨUPK_m).
- *Public key replace query* (ID_m, ($\Psi MPK'_m$, $\Psi UPK'_m$)): B converts ($\Psi MPK'_m$, $\Psi UPK'_m$) to ($\Theta MPK'_m$, $\Theta UPK'_m$). B modifies (ID_m, $-$, $\Theta MPK'_m$, 1) in L_{MS} and (ID_m, $-$, $\Theta UPK'_m$, 1) in L_{MC}.
- *Signcryption query* (msg, ID_s, ID_r): Assume that the sender ID_s has the j-th certificate pair ($\Theta CTF_{s,1,j}$, $\Theta CTF_{s,2,j}$) and secret key pair ($\Theta MSK_{s,1,j}$, $\Theta MSK_{s,2,j}$). B runs the following procedure.
  (1) Use ID_r to find (ID_r, ΘMSK_r, ΘMPK_r, *replace*) in L_{MS} and (ID_r, ΘCTF_r, ΘUPK_r, *replace*) in L_{MC}, and convert (ΘMPK_r, ΘUPK_r) in L_1 to (ΨMPK_r, ΨUPK_r).
  (2) Set $\Psi\alpha = ID_r || \Psi MPK_r || \Psi UPK_r$ and convert $\Psi\alpha$ in L_1 to $\Theta\alpha$.
  (3) Choose a new variate ΘU in G_1, and set $\Theta K_1 = \Theta MPK_r \cdot \Theta U$ and $\Theta K_2 = (\Theta CPK + \Theta UPK_r \cdot (\Theta W + \Theta X \cdot \Theta\alpha)) \cdot \Theta U$.
  (4) Convert both ΘU and ΘK_1 in L_1, and ΘK_2 in L_2 to obtain ΨU, ΨK_1 and ΨK_2.
  (5) Set $\Psi K = \Psi K_1 \oplus \Psi K_2$ and $C = SKE_{\Psi K}(msg)$.
  (6) Choose a new variate $\Theta\delta$ in G_1, and set $\Psi\delta = H(msg||C||\Psi U||ID_s||ID_r)$, and put ($\Theta\delta$, $\Psi\delta$) in L_1.
  (7) Set $\Theta\alpha = \Theta CTF_s + \Theta MSK_s + (\Theta Y + \Theta Z \cdot \Theta\delta) \cdot \Theta U$ and convert $\Theta\delta$ in L_1 to $\Psi\sigma$.
  (8) Put (ID_s, ID_r, msg, ΘU, ΘK_1, ΘK_2, $\Theta\sigma$, C, $\Theta\delta$) in L_{SC}.
  (9) Return $CT = (\Psi\sigma, C, \Psi U, ID_s, ID_r)$.
- *Signcryption leak query* (ID_s, j, $f_{SC,j}$, $h_{SC,j}$): A_I can issue this query only once for the member ID_s's j-th certificate pair ($\Theta CTF_{s,1,j}$, $\Theta CTF_{s,2,j}$) and secret key pair ($\Theta MSK_{s,1,j}$, $\Theta MSK_{s,2,j}$). B returns $\Delta f_{SC,j} = f_{SC,j}(\Theta CTF_{s,1,j}, \Theta MSK_{s,1,j})$ and $\Delta h_{SC,j} = h_{SC,j}(\Theta CTF_{s,2,j}, \Theta MSK_{s,2,j})$.
- *Unsigncryption query* (ID_r, $CT = (\Psi\sigma, C, \Psi U, ID_s, ID_r)$): Assume that the receiver ID_r has the k-th certificate pair ($\Theta CTF_{r,1,k}$, $\Theta CTF_{r,2,k}$) and secret key pair ($\Theta MSK_{r,1,k}$, $\Theta MSK_{r,2,k}$). B runs the following procedure.

(1) Use ID_s to find (ID_s, ΘMSK_s, $\Theta MPKs$, *replace*) in L_{MS} and (ID_s, ΘCTF_s, ΘUPK_s, *replace*) in L_{MC}, and convert (ΘMPK_s, ΘUPK_s) in L_1 to (ΨMPK_s, ΨUPK_s).
(2) Convert $\Psi\sigma$ and ΨU in L_1 to $\Theta\sigma$ and ΘU, respectively.
(3) Set $\Theta K_1 = \Theta U \cdot \Theta MSK_r$ and $\Theta K_2 = \Theta U \cdot \Theta CTF_r$.
(4) Set $\Psi\delta = H(msg||C||\Psi U||ID_s||ID_r)$ and convert $\Psi\delta$ in L_1 to $\Theta\delta$.
(5) Use $ID_s, ID_r, \Theta U, \Theta K_1, \Theta K_2, \Theta\sigma, C$ and $\Theta\delta$ to find (ID_s, ID_r, msg, ΘU, ΘK_1, ΘK_2, $\Theta\sigma$, C, $\Theta\delta$) in L_{SC}. If it is found, return msg and "Valid".

- *Unsigncryption leak query* (ID_r, k, $f_{US,k}$, $h_{US,k}$): A_I can issue this query only once for the member ID_r's k-th certificate pair ($\Theta CTF_{r,1,k}$, $\Theta CTF_{r,2,k}$) and secret key pair ($\Theta MSK_{r,1,k}$, $\Theta MSK_{r,2,k}$). B returns $\Delta f_{US,k} = f_{US,k}(\Theta CTF_{r,1,k}, \Theta MSK_{r,1,k})$ and $\Delta h_{US,k} = h_{US,k}(\Theta CTF_{r,2,k}, \Theta MSK_{r,2,k})$.

– *Forgery*. A_I produces a pair (msg', $CT' = (\Psi\sigma', C', \Psi U', ID'_s, ID'_r)$). Note that the *Signcryption query* (msg', ID'_s, ID'_r) and the *Certificate generation query* (ID'_s, MPK'_s) have never been requested. If the *Unsigncryption* algorithm with (ID'_r, $CT' = (\Psi\sigma', C', \Psi U', ID'_s, ID'_r)$) returns msg and "Valid", we say that A_I wins G_{auth}.

As mentioned in Section 2.2, if an adversary found any collision of G_1/G_2, it would solve the *discrete logarithm problem* on G_1/G_2. In such a case, we first compute the sum amount of elements in L_1 and L_2, termed as $|L_1| + |L_2|$. By counting the added elements in both the *Setup* and *Query* phases, we have the inequality $|L_1| + |L_2| \leqq 6\eta$ because A_I can request different queries to B η times and at most 6 elements are increased in L_1/L_2 after issuing a query. Additionally, for evaluating the entropy of secret keys with partial leakage, we must compute the maximal degrees of elements in L_1 and L_2, in which L_1 and L_2 have the maximal degrees 3 and 6, respectively.

Subsequently, let us compute the advantage Adv_{AI-N} that A_I wins G_{auth} without issuing leak queries. The advantage Adv_{AI-N} comprises two probabilities as discussed below.

■ Let Pr[*Collision*] be the probability that A_I finds a collision in L_1 or L_2. Assume that there are r different variates in L_1, which are denoted by r random values $v_i \in Z_p^*$ for $i = 1, 2, \ldots, r$. Let $\Theta G_{1,j}$ and $\Theta G_{1,k}$ be two distinct elements in L_1 and set $\Theta G_{1,l} = \Theta G_{1,j} - \Theta G_{1,k}$. If $\Theta G_{1,l}(v_1, v_2, \ldots, v_r) = 0$, there exists a collision in L_1. Because L_1 has $\binom{|L_1|}{2}$ pairs of ($\Theta G_{1,j}$, $\Theta G_{1,k}$) and the maximal polynomial degree is 3, the probability that A_I finds a collision in L_1 is $(3/p)\binom{|L_1|}{2}$ by Lemma 2. For the same reason, the probability that A_I finds a collision in L_2 is $(6/p)\binom{|L_2|}{2}$. Because of $|L_1| + |L_2| \leqq 6\eta$, we have the following inequality Pr[*Collision*] $\leqq (3/p)\binom{|L_1|}{2} + (6/p)\binom{|L_2|}{2} \leqq (6/p)(|L_0| + |L_1|)^2 \leqq 216\eta^2/p$.

■ Let Pr[*Forge*] be the probability that A_I forges a valid pair (msg', $CT' = (\Psi\sigma', C', \Psi U', ID'_s, ID'_r)$). The valid pair satisfies $\hat{e}(g_1, \sigma') = CPK \cdot MPK'_s \cdot \hat{e}(UPK'_s, W \cdot X^{\alpha}) \cdot \hat{e}(U', Y \cdot Z^{\delta})$ in the *Unsigncryption* algorithm, namely, $\Theta g_1 \cdot \Theta\sigma' = \Theta CPK + \Theta MPK'_s + \Theta UPK'_s \cdot (\Theta W + \Theta X \cdot \Theta\alpha) + \Theta U' \cdot (\Theta Y + \Theta Z \cdot \Theta\delta)$. Let $\Theta f = \Theta g_1 \cdot \Theta\sigma' - \Theta CPK + \Theta MPK'_s + \Theta UPK'_s \cdot (\Theta W + \Theta X \cdot \Theta\alpha) + \Theta U' \cdot (\Theta Y + \Theta Z \cdot \Theta\delta)$. Since Θf has degree 3, the probability of $\Theta f = 0$ is $3/p$ by Lemma 2 so that we have Pr[*Forge*] $= 3/p$.

By above, we have $Adv_{AI-N} = \Pr[Collision]+\Pr[Forge] \leqq 216\eta^2/p+3/p = O(\eta^2/p)$, which is negligible if $\eta = poly(\log p)$.

Here, we compute the advantage Adv_{AI} that AI wins G_{auth} when permitted to issue three types of leak queries, namely, *Certificate generation leak query*, *Signcryption leak query* and *Unsigncryption leak query*.

(1) By the *Certificate generation leak query* $(i, f_{CA,i}, h_{CA,i})$, A_I gets partial bits of the CA's i-th secret key pair $(\Theta CSK_{1,i}, \Theta CSK_{2,i})$, namely, $\Delta f_{CA,i} = f_{CA,i}(\Theta CSK_{1,i})$ and $\Delta h_{CA,i} = h_{CA,i}(\Theta CSK_{2,i})$ with $|\Delta f_{CA,i}|, |\Delta_{hCA,i}| \leqq \omega$. According to the key update process (Kiltz and Pietrzak, 2010; Galindo and Virek, 2013), the CA's i-th secret key pair satisfies the relations $CSK = CSK_{1,0} \cdot CSK_{2,0} = \cdots = CSK_{1,i-1} \cdot CSK_{2,i-1} = CSK_{1,i} \cdot CSK_{2,i}$. Meanwhile, the leakage bits of $(\Theta CSK_{1,i-1}, \Theta CSK_{2,i-1})$ and $(\Theta CSK_{1,i}, \Theta CSK_{2,i})$ are mutually independent, so that A_I gets at most 2ω bits of CSK.

(2) By the *Signcryption leak query* $(ID_s, j, f_{SC,j}, h_{SC,j})$, A_I gets partial bits of the ID_s's j-th certificate pair $(\Theta CTF_{s,1,j}, \Theta CTF_{s,2,j})$, namely, $\Delta f_{SC,j} = f_{SC,j}(\Theta CTF_{s,1,j})$ and $\Delta h_{SC,j} = h_{SC,j}(\Theta CTF_{s,2,j})$ with $|\Delta f_{SC,j}|, |\Delta h_{SC,j}| \leqq \omega$. According to the key update procedure, the ID_s's j-th certificate pair satisfies the relations $CTF_s = CTF_{s,1,0} \cdot CTF_{s,2,0} = \cdots = CTF_{s,1,j-1} \cdot CTF_{s,2,j-1} = CTF_{s,1,j} \cdot CTF_{s,2,j}$. Thus, A_I gets at most 2ω bits of CTF_s.

(3) By the *Unsigncryption leak query* $(ID_r, k, f_{US,k}, h_{US,k})$, A_I gets partial bits of the ID_r's k-th certificate pair $(\Theta CTF_{r,1,k}, \Theta CTF_{r,2,k})$, namely, $\Delta f_{US,k} = f_{US,k}(\Theta CTF_{r,1,k})$ and $\Delta h_{US,k} = h_{US,k}(\Theta CTF_{r,2,k})$ with $|\Delta f_{US,k}|, |\Delta h_{US,k}| \leqq \omega$. According to the key update procedure, the ID_r's k-th certificate pair satisfies the relations $CTF_r = CTF_{r,1,0} \cdot CTF_{r,2,0} = \cdots = CTF_{r,1,j-1} \cdot CTF_{r,2,j-1} = CTF_{r,1,j} \cdot CTF_{r,2,j}$. Thus, A_I gets at most 2ω bits of CTF_r.

Due to the discussions above, we define three events as follows:

(1) Let $EVCSK$ indicate the event that A_I obtains CSK by $\Delta f_{CA,i}$ and $\Delta h_{CA,i}$. Meanwhile, $\overline{EVCSK}$ means $EVCSK$'s complement.

(2) Let $EVCTF$ indicate the event that A_I gets CTF_m (i.e. CTF_s or CTF_r) by $\Delta f_{SC,j}$, $\Delta h_{SC,j}$, $\Delta f_{US,k}$ and $\Delta h_{US,k}$. Meanwhile, $\overline{EVCTF}$ means $EVCTF$'s complement.

(3) Let $ESFV$ indicate the event that A_I successfully forges a valid pair $(msg', CT' = (\Psi\sigma', C', \Psi U', ID'_s, ID'_r))$.

Hence, Adv_{AI} has the inequality

$$\begin{aligned} Adv_{AI} &= \Pr[ESFV] \\ &= \Pr\big[ESFV \wedge (EVCSK \vee EVCTF)\big] + \Pr\big[ESFV \wedge (\overline{EVCSK} \wedge \overline{EVCTF})\big] \\ &\leqq \Pr\big[(EVCSK \vee EVCTF)\big] + \Pr\big[ESFV \wedge (\overline{EVSSK} \wedge \overline{EVUDID})\big]. \end{aligned}$$

By the *Certificate generation leak query*, A_I gets 2ω bits of CSK. Also, by the *Signcryption leak query* or *Unsigncryption leak query*, A_I gets 2ω bits of CTF_m (i.e. CTF_s or CTF_r). By Lemma 2, we have $\Pr[(EVCSK \vee EVCTF)] \leqq Adv_{AI-N} \cdot 2^{2\omega} \leqq O((\eta^2/p) \cdot 2^{2\omega})$

because of $Adv_{AI-N} = O(\eta^2/p)$. Since $\Pr[ESFV \wedge (\overline{EVSSK} \wedge \overline{EVUDID})]$ denotes that A_I gets no information of both CSK and CTF_m, we have $\Pr[ESFV \wedge (\overline{EVSSK} \wedge \overline{EVUDID})] = Adv_{AI-N} = O(\eta^2/p)$. Therefore,

$$\begin{aligned} Adv_{AI} &\leqq \Pr\big[(EVCSK \vee EVCTF)\big] + \Pr\big[ESFV \wedge (\overline{EVSSK} \wedge \overline{EVUDID})\big] \\ &\leqq O\big((\eta^2/p) \cdot 2^{2\omega}\big) + O\big(\eta^2/p\big) = O\big((\eta^2/p) \cdot 2^{2\omega}\big). \end{aligned}$$

Finally, Adv_{AI} is negligible if $\omega = poly(\log p)$, by Lemma 2. □

Theorem 2. *Under the SCRH and DL assumptions in the GBG model, our FCLR-CBSC scheme is EXUF-CLRACMA-secure against A_{II} in G_{auth}.*

Proof. A_{II} and B play G_{auth} that comprises three phases as presented below.

- *Setup*. It is identical to the *Setup* in Theorem 1. Because the adversary is of A_{II}, B sends ΨCSK to A_{II}.
- *Queries*. It is identical to the *Queries* in Theorem 1. Additionally, since A_{II} possesses the CA's secret key CSK, so that it can create any member ID_m's certificate CTF_m and second public key UPK_m. Thus, A_{II} has no need to request the *Certificate generation query* and *Certificate generation leak query*.
- *Forgery*. A_{II} produces a pair $(msg', CT' = (\Psi\sigma', C', \Psi U', ID'_s, ID'_r))$. Note that the *Signcryption query* (msg', ID'_s, ID'_r), the *Member secret key query* (ID_m) and the *Public key replace query* $(ID_m, (\Psi MPK'_m, \Psi UPK'_m))$ have never been requested. If the *Unsigncryption* algorithm with $(ID'_r, CT' = (\Psi\sigma', C', \Psi U', ID'_s, ID'_r))$ returns msg and "Valid", we say that A_I wins G_{auth}.

Here, let's compute the advantage Adv_{AII-N} that A_{II} wins G_{auth} without issuing leak queries. By $\Pr[Collision]$ and $\Pr[Forge]$ defined in Theorem 1, we have $Adv_{AII-N} = \Pr[Collision] + \Pr[Forge] \leqq 216\eta^2/p + 3/p = O(\eta^2/p)$, which is negligible if $\eta = poly(\log p)$. Next, we compute the advantage Adv_{AII} that A_{II} wins G_{auth} when permitted to issue two types of leak queries, namely, *Signcryption leak query* and *Unsigncryption leak query*.

(1) By the *Signcryption leak query* $(ID_s, j, f_{SC,j}, h_{SC,j})$, A_{II} gets partial bits of the ID_s's j-th secret key pair $(MSK_{s,1,j}, MSK_{s,2,j})$, namely, $\Delta f_{SC,j} = f_{SC,j}(MSK_{s,1,j})$ and $\Delta h_{SC,j} = h_{SC,j}(MSK_{s,2,j})$ with $|\Delta f_{SC,j}|, |\Delta h_{SC,j}| \leqq \omega$. According to the key update procedure, the ID_s's j-th secret key pair satisfies the relations $MSK_s = MSK_{s,1,0} \cdot MSK_{s,2,0} = \cdots = MSK_{s,1,j-1} \cdot MSK_{s,2,j-1} = MSK_{s,1,j} \cdot MSK_{s,2,j}$. Thus, A_{II} gets at most 2ω bits of MSK_s.

(2) By the *Unsigncryption leak query* $(ID_r, k, f_{US,k}, h_{US,k})$, A_{II} gets partial bits of the ID_r's k-th secret key pair $(MSK_{r,1,k}, MSK_{r,2,k})$, namely, $\Delta f_{US,k} = f_{US,k}(MSK_{r,1,k})$ and $\Delta h_{US,k} = h_{US,k}(MSK_{r,2,k})$ with $|\Delta f_{US,k}|, |\Delta h_{US,k}| \leqq \omega$. According to the key update procedure, the ID_r's k-th secret key pair satisfies the relations $MSK_r = MSK_{r,1,0} \cdot MSK_{r,2,0} = \cdots = MSK_{r,1,j-1} \cdot MSK_{r,2,j-1} = MSK_{r,1,j} \cdot MSK_{r,2,j}$. Thus, A_{II} gets at most 2ω bits of MSK_r.

Due to the discussions above, we define two events as follows:

(1) Let *EVMSK* indicate the event that A_{II} obtains MSK_m (i.e. MSK_s or MSK_r) by $\Delta f_{SC,j}$, $\Delta h_{SC,j}$, $\Delta f_{US,k}$ and $\Delta h_{US,k}$. Meanwhile, $\overline{EVMSK}$ denotes *EVMSK*'s complement.
(2) Let *ESFV* indicate the event that A_{II} successfully forges a valid tuple $(msg', CT' = (\Psi\sigma', C', \Psi U', ID'_s, ID'_r))$.

Hence, Adv_{AII} has the inequality

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

By the *Signcryption leak query* or *Unsigncryption leak query*, A_{II} gets 2ω bits of MSK_m (i.e. MSK_s or MSK_r). By Lemma 2, we have $\Pr[EVMSK] \leqq Adv_{AII-N} \cdot 2^{2\omega} \leqq O((\eta^2/p) \cdot 2^{2\omega})$. Since $\Pr[ESFV \wedge \overline{EVMSK}]$ denotes that A_{II} gets no information of MSK_m, we have $\Pr[ESFV \wedge \overline{EVMSK}] = Adv_{AII-N} = O(\eta^2/p)$. Therefore,

$$\begin{aligned} Adv_{AII} &\leqq \Pr[EVMSK] + \Pr[ESFV \wedge \overline{EVMSK}] \\ &\leqq O\big((\eta^2/p) \cdot 2^{2\omega}\big) + O\big(\eta^2/p\big) = O\big((\eta^2/p) \cdot 2^{2\omega}\big). \end{aligned}$$

Finally, Adv_{AII} is negligible if $\omega = poly(\log p)$, by Lemma 2. □

Theorem 3. *Under the SCRH and DL assumptions in the GBG model, our FCLR-CBSC scheme is INDEN-CLCCA-secure against A_I in G_{conf}.*

Proof. A_I and B play G_{conf} that comprises four phases as presented below:

– *Setup*. It is identical to the *Setup* in Theorem 1.
– *Queries*. It is identical to the *Queries* in Theorem 1.
– *Challenge*. A_I sends a receiver's identity ID'_r and a message pair (msg'_0, msg'_1) to B. Note that the *Certificate generation query* (ID'_r, MPK'_r) has never been requested. B randomly chooses a bit $b' \in \{0, 1\}$ and runs the *Signcryption* algorithm with (msg'_b, ID_s, ID'_r) to produce and return a ciphertext $CT = (\sigma, C, U, ID_s, ID'_r)$ to A_I.
– *Guess*. A_I returns a bit $b \in \{0, 1\}$. We say that A_I wins G_{conf} if $b = b'$. The advantage Adv_{AI} is defined as $|\Pr[b = b'] - 1/2|$.

Let us compute the advantage Adv_{AI-N} that A_I wins G_{conf} without issuing leak queries. The advantage Adv_{AI-N} comprises two probabilities as discussed below:

■ Let $\Pr[Collision]$ be the probability that A_I finds a collision in L_1 or L_2, which is the same with the probability $\Pr[Collision]$ in Theorem 1. Thus, we have the inequality $\Pr[Collision] \leqq 216\eta^2/p$.
■ Let $\Pr[Guess]$ be the probability that A_I with no useful information outputs a correct bit b. Thus, we have $\Pr[Guess] = Pr[b = b'] = 1/2$.

Hence, we have the following inequality.

$$Adv_{AI-N} = |\Pr[b = b'] - 1/2| = |\Pr[Collision] + \Pr[Guess] - 1/2| \leqq 216\eta^2/p = O(\eta^2/p).$$

Here, let's compute the advantage Adv_{AI} that A_I wins G_{conf} when permitted to issue three types of leak queries, namely, *Certificate generation leak query*, *Signcryption leak query* and *Unsigncryption leak query*. As in the proof of Theorem 1, by the *Certificate generation leak query*, A_I gets 2ω bits of *CSK*. Also, by the *Signcryption leak query* or the *Unsigncryption leak query*, A_I gets 2ω bits of CTF_m (i.e. CTF_s or CTF_r). By Lemma 2, we have $Adv_{AI} \leqq Adv_{AI-N} \cdot 2^{2\omega} \leqq O((\eta^2/p) \cdot 2^{2\omega})$, which is negligible if $\omega = poly(\log p)$. □

Theorem 4. *Under the SCRH and DL assumptions in the GBG model, our FCLR-CBSC scheme is INDEN-CLCCA-secure against A_{II} in G_{conf}.*

Proof. A_{II} and B play G_{conf} that comprises four phases as presented below:

- *Setup*. It is identical to the *Setup* in Theorem 2.
- *Queries*. It is identical to the *Queries* in Theorem 2.
- *Challenge*. A_{II} sends a receiver's identity ID'_r and a message pair (msg'_0, msg'_1) to B. Note that neither the *Member secret key query* (ID'_r) nor the *Public key replace query* $(ID'_r, (MPK'_r, UPK'_r))$ has been requested. B randomly chooses a bit $b' \in \{0, 1\}$ and runs the *Signcryption* algorithm with (msg'_b, ID_s, ID'_r) to produce and return a ciphertext $CT = (\sigma, C, U, ID_s, ID'_r)$ to A_{II}.
- *Guess*. A_{II} returns a bit $b \in \{0, 1\}$. We say that A_{II} wins G_{conf} if $b = b'$. The advantage Adv_{AII} is defined as $|\Pr[b = b'] - 1/2|$.

Here, let's compute Adv_{AII-N} that A_{II} wins G_{conf} without request leak queries. By using $\Pr[Collision]$ and $\Pr[Guess]$ in the proof of Theorem 3, we get $Adv_{AII-N} = \Pr[Collision] + \Pr[Guess] \leqq 216\eta^2/p + 3/p = O(\eta^2/p)$ that is negligible if $\eta = poly(\log p)$. Subsequently, let us compute the advantage Adv_{AII} that A_{II} wins G_{conf} when permitted to issue two types of leak queries, namely, *Signcryption leak query* and *Unsigncryption leak query*. As in the proof of Theorem 2, A_{II} gets 2ω bits of MSK_m (i.e. MSK_s or MSK_r). By Lemma 2, we obtain $Adv_{AII} \leqq Adv_{AII-N} \cdot 2^{2\omega} \leqq O((\eta^2/p) \cdot 2^{2\omega})$ that is negligible if $\omega = poly(\log p)$. □

6. Performance Comparisons

Here, let's evaluate the computation time of our FCLR-CBSC scheme in terms of *Initialization*, *Member secret key generation*, *Certificate generation*, *Signcryption* and *Unsigncryption* algorithms. By the simulation results in Xiong and Qin (2015), the notations (i.e. T_{bpf} and T_{exp}) for two time-consuming computations and their running time are presented

Table 3
Notations and running time of two time-consuming operations.

Operations	Notations	Running time on a PC	Running time on a mobile device
Bilinear pairing function $\hat{e}$	T_{bpf}	≈20 ms	≈96 ms
Exponentiation in G_1 or G_2	T_{exp}	≈7 ms	≈31 ms

Table 4
Computation costs and running time of five algorithms.

Algorithms	Computation costs	Running time on a PC	Running time on a mobile device
Initialization	$T_{bpf} + 7T_{exp}$	69 ms	313 ms
Member secret key generation	$T_{bpf} + 3T_{exp}$	41 ms	189 ms
Certificate generation	$7T_{exp}$	49 ms	217 ms
Signcryption	$T_{bpf} + 8T_{exp}$	76 ms	344 ms
Unsigncryption	$7T_{bpf} + 4T_{exp}$	168 ms	796 ms

in Table 3. Additionally, the running time of the multiplication in G_1 or G_2 is negligible since it is more slighter than T_{bpf} and T_{exp}. The simulation results in Xiong and Qin (2015) are evaluated under a PC with an Intel 1.80-GHz i7 CPU and a mobile device with an Intel 624-MHz PXA270 CPU. Meanwhile, the order p of both G_1 and G_2 is a 512-bit prime security level. The computation costs and the running time of five algorithms in our FCLR-CBSC scheme are listed in Table 4. By Table 4, it is obvious that our scheme performs efficiently on both a PC and a mobile device.

7. Conclusions

A practical FCLR-CBSC scheme was proposed in the paper. As compared with the previously proposed LR-CLSC and CLR-CBSC schemes, our scheme possesses the fully continuous leakage-resilient property. In our scheme, by the key update method participated in the *Certificate generation, Signcryption* and *Unsigncryption* algorithms of our scheme, respectively, an adversary is permitted to obtain partial bits of the CA's secret key, and a sender/receiver's certificate and secret key. Based on the SCRH and DL assumptions in the GBG model, four security theorems were formally shown that our scheme is EXUF-CLRACMA-secure and INDEN-CLCCA-secure against two types of adversaries (A_I and A_{II}) in the CB-PKS setting so that our scheme possesses both authentication of and confidentiality. Finally, performance analysis demonstrated that our scheme is performs efficiently on both a PC and a mobile device.

References

Ali, I., Lawrence, T., Omala, A., Li, F. (2020). An efficient hybrid signcryption scheme with conditional privacy-preservation for heterogeneous vehicular communication in VANETs. *IEEE Transactions on Vehicular Technology*, 69(10), 11266–11280.

Al-Riyami, S., Paterson, K. (2003). Certificateless public key cryptography. In: *ASIACRYPT'03*, *LNCS*, Vol. 2894, pp. 452–473.

Alwen, J., Dodis, Y., Wichs, D. (2009). Leakage-resilient public-key cryptography in the bounded-retrieval model. In: *Crypto'09*, *LNCS*, Vol. 5677, pp. 36–54.

Biham, E., Carmeli, Y., Shamir, A. (2008). Bug attacks. In: *Crypto'08*, *LNCS*, Vol. 5157, pp. 221–240.

Boneh, D., Franklin, M. (2001). Identity-based encryption from the Weil pairing. In: *Crypto'01*, *LNCS*, Vol. 2139, pp. 213–229.

Boneh, D., Boyen, X., Goh, E. (2005). Hierarchical identity-based encryption with constant size ciphertext. In: *Eurocrypt'05*, *LNCS*, Vol. 3494, pp. 440–456.

Brumley, D., Boneh, D. (2005). Remote timing attacks are practical. *Computer Networks*, 48(5), 701–716.

Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A. (2008). Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1), 97–139.

Dodis, Y., Haralambiev, K., Lopez-Alt, A., Wichs, D. (2010). Cryptography resilient to continual memory leakage. In: *51st Annual IEEE Symposium on Foundations of Computer Science*, pp. 501–510.

Galindo, D., Virek, S. (2013). A practical leakage-resilient signature scheme in the generic group model. In: *SAC'12*, *LNCS*, Vol. 7707, pp. 50–65.

Galindo, D., Grobschadl, J., Liu, Z., Vadnala, P., Vivek, S. (2016). Implementation of a leakage-resilient ElGamal key encapsulation mechanism. *Journal of Cryptographic Engineering*, 6(3), 229–238.

Gentry, C. (2003). Certificate-based encryption and the certificate revocation problem. In: *EUROCRYPT'03*, *LNCS*, Vol. 2656, pp. 272–293.

Hussain, S., Ullah, I., Khattak, H., Adnan, M., Kumari, S., Ullah, S., Khan, M., Khattak, S. (2020). A lightweight and formally secure certificate based signcryption with proxy re-encryption (CBSRE) for internet of things enabled smart grid. *IEEE Access*, 8, 93230–93248.

Katz, J., Vaikuntanathan, V. (2009). Signature schemes with bounded leakage resilience. In: *Asiacrypt'09*, *LNCS*, Vol. 5912, pp. 703–720.

Khan, M., Ullah, I., Nisar, S., Noor, F., Qureshi, I., Khanzada, F., Amin, N. (2020). An efficient and provably secure certificateless key-encapsulated signcryption scheme for flying ad-hoc network. *IEEE Access*, 8, 36807–36828.

Kiltz, E., Pietrzak, K. (2010). Leakage resilient Elgamal encryption. In: *Asiacrypt'10*, *LNCS*, Vol. 6477, pp. 595–612.

Kocher, P., Jaffe, J., Jun, B. (1999). Differential power analysis. In: *Crypto'99*, *LNCS*, Vol. 1666, pp. 388–397.

Peng, A.-L., Tseng, Y.-M., Huang, S.-S. (2021). An efficient leakage-resilient authenticated key exchange protocol suitable for IoT devices. *IEEE Systems Journal*, 15(4), 5343–5354.

Rivest, R., Shamir, A., Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120–126.

Tsai, T.-T., Huang, S.-S., Tseng, Y.-M., Chuang, Y.-H., Hung, Y.-H. (2022). Leakage-resilient certificate-based authenticated key exchange protocol. *IEEE Open Journal of the Computer Society*, 3, 137–148.

Tseng, Y.-M., Wu, J.-D., Huang, S.-S., Tsai, T.-T. (2020). Leakage-resilient outsourced revocable certificateless signature with a cloud revocation server. *Information Technology and Control*, 49(4), 464–481.

Tseng, Y.-M., Chen, J.-L., Huang, S.-S. (2021). A lightweight leakage-resilient identity-based mutual authentication and key exchange protocol for resource-limited devices. *Computer Networks*, 196, 108246.

Tseng, Y.-M., Huang, S.-S., Tsai, T.-T., Chuang, Y.-H., Hung, Y.-H. (2022). Leakage-resilient revocable certificateless encryption with an outsourced revocation authority. *Informatica*, 33(1), 151–179.

Ullah, I., Alomari, A., Amin, N., Khan, M., Khattak, H. (2019). An energy efficient and formally secured certificate-based signcryption for wireless body area networks with the internet of things. *Electronics*, 8(10), 1171.

Ullah, S., Li, X., Lan, Z. (2020). A novel trusted third party based signcryption scheme. *Multimedia Tools and Applications*, 79, 22749–22769.

Wu, Y., Gong, B., Zhang, Y. (2022). An improved efficient certificateless hybrid signcryption scheme for internet of things. *Wireless Communications and Mobile Computing*, 2022, 6945004.

Wu, J.-D., Tseng, Y.-M., Huang, S.-S., Chou, W.-C. (2018). Leakage-resilient certificateless key encapsulation scheme. *Informatica*, 29(1), 125–155.

Wu, J.-D., Tseng, Y.-M., Huang, S.-S., Tsai, T.-T. (2019). Leakage-resilient certificate-based signature resistant to side-channel attacks. *IEEE Access*, 7(1), 19041–19053.

Wu, J.-D., Tseng, Y.-M., Huang, S.-S., Tsai, T.-T. (2020a). Leakage-resilient certificate-based key encapsulation scheme resistant to continual leakage. *IEEE Open Journal of the Computer Society*, 1, 131–144.

Wu, J.-D., Tseng, Y.-M., Huang, S.-S., Tsai, T.-T. (2020b). Leakage-resilient revocable identity-based signature with cloud revocation authority. *Informatica*, 31(3), 597–620.

Xiong, H., Qin, Z. (2015). Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks. *IEEE Transactions on Information Forensics and Security*, 10(7), 1442–1455.

Yang, Q., Zhou, Y., Yu, Y. (2019). Leakage-resilient certificateless signcryption scheme. In: *GLOBECOM Workshops*, pp. 1–6.

Zheng, Y. (1997). Digital signcryption or how to achieve cost (signature & encryption) cost (signature)+ cost (encryption). In: *Annual International Cryptology Conference*, *LNCS*, Vol. 1294, pp. 165–179.

Zhou, Y., Yang, B., Zhang, W. (2016). Provably secure and efficient leakage-resilient certificateless signcryption scheme without bilinear pairing. *Discrete Applied Mathematics*, 204, 185–202.

Zhou, Y., Xu, Y., Qiao, Z., Yang, B., Zhang, M. (2021). Continuous leakage-resilient certificate-based signcryption scheme and application in cloud computing. *Theoretical Computer Science*, 860, 1–22.

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

T.-T. Tsai is currently an assistant professor in the Department of Computer Science and Engineering, National Taiwan Ocean University, Taiwan. His research interests include applied cryptography, pairing-based cryptography and leakage-resilient cryptography. He received the PhD degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2014, under the supervision of professor Yuh-Min Tseng.

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