INFORMATICA, 2024, Vol. 35, No. 1, 131–154

DOI: https://doi.org/10.15388/24-INFOR546

Leakage-Resilient Hybrid Signcryption in Heterogeneous Public-key Systems

Ting-Chieh HO, Yuh-Min TSENG*, Sen-Shan HUANG

Department of Mathematics, National Changhua University of Education, Changhua 500, Taiwan
e-mail: ymtseng@cc.ncue.edu.tw

Received: September 2023; accepted: February 2024

Abstract. Signcryption integrates both signature and encryption schemes into single scheme to ensure both content unforgeability (authentication) and message confidentiality while reducing computational complexity. Typically, both signers (senders) and decrypters (receivers) in a signcryption scheme belong to the same public-key systems. When signers and decrypters in a signcryption scheme belong to heterogeneous public-key systems, this scheme is called a hybrid signcryption scheme which provides more elastic usage than typical signcryption schemes. In recent years, a new kind of attack, named side-channel attack, allows adversaries to learn a portion of the secret keys used in cryptographic algorithms. To resist such an attack, leakage-resilient cryptography has been widely discussed and studied while a large number of leakage-resilient schemes have been proposed. Also, numerous hybrid signcryption schemes under heterogeneous public-key systems were proposed, but none of them possesses leakage-resilient property. In this paper, we propose the first hybrid signcryption scheme with leakage resilience, called leakage-resilient hybrid signcryption scheme, in heterogeneous public-key systems (LR-HSC-HPKS). Security proofs are demonstrated to show that the proposed scheme provides both authentication and confidentiality against two types of adversaries in heterogeneous public-key systems.

Key words: heterogeneous public-key systems, side-channel attack, leakage-resilience, signcryption.

1. Introduction

Public key cryptography is the foundation of modern information security. So far, several famous public-key systems (PKSs) have been proposed, including public-key infrastructure PKS (PKI-PKS) (Rivest *et al.*, 1978), identity-based PKS (ID-PKS) (Boneh and Franklin, 2001) and certificateless PKS (CL-PKS) (Al-Riyami and Paterson, 2003). These PKSs have evolved in response to their advantages and disadvantages. In the PKI-PKS (Rivest *et al.*, 1978), a user with identity first generates a pair of (secret key, public key) randomly. Also, the user sends her/his identity and public key to a trusted certificate authority (CA) and then receives the associated certificate from the CA. The CA is responsible to respond the management issues of users' public keys and certificates that include the

*Corresponding author.

verification queries for expiration date or revoked users. Thus, a complex PKI architecture needs to be constructed.

To remove such a complex PKI architecture, an identity-based PKS (ID-PKS) was proposed by Boneh and Franklin (2001). In the ID-PKS, a trusted private key generator (PKG) is responsible for producing each member's secret key by taking each member's identity as input. Therefore, this ID-PKS encountered a key escrow problem because the PKG possesses all members' secret keys. To resolve the key escrow problem, a certificateless PKS (CL-PKS) was proposed by Al-Riyami and Paterson (2003). In the CL-PKS, each member holds two pairs of (secret key, public key). One pair is created by the member herself/himself and the other pair is generated by a semi-trusted key generation centre (KGC). Indeed, the CL-PKS possesses the advantages of both the PKI-PKS and the ID-PKS while avoiding their disadvantages. Therefore, this CL-PKS does not require the complex PKI construction and solves the key escrow problem.

In recent years, a new kind of attack, named side-channel attack, has been realized (Brumley and Boneh, 2005; Biham *et al.*, 2008), in the sense that adversaries can learn a portion of these secret keys used in cryptographic algorithms by timing, power analysis or fault attack. By repeatedly using the side-channel attack, adversaries could eventually learn the entire secret keys. Therefore, public-key cryptography failing to resist such side-channel attack is insecure. To resist this attack, leakage-resilient cryptography has been widely discussed and studied by researchers who have also presented a large number of leakage-resilient protocol or schemes (Alwen *et al.*, 2009; Akavia *et al.*, 2009; Kiltz and Pietrzak, 2010; Galindo and Virek, 2013; Galindo *et al.*, 2016; Wu *et al.*, 2018, 2019; Tseng *et al.*, 2020; Peng *et al.*, 2021; Tseng *et al.*, 2022a,b; Xie *et al.*, 2023; Tseng *et al.*, 2023; Tsai *et al.*, 2023). Based on adversaries' leakage ability, leakage-resilient cryptography is secure in two different leakage models, including the bounded leakage model (Alwen *et al.*, 2009; Akavia *et al.*, 2009) and the unbounded leakage model (Kiltz and Pietrzak, 2010; Galindo and Virek, 2013). Indeed, the unbounded leakage model is considered a more practical and widely accepted leakage model since it only limits the amount of leaked information per round and has overall unbounded characteristics.

1.1. *Motivation*

Encryption and signature are two important foundations in public-key cryptography. Signcryption integrates both signature and encryption schemes into single scheme to ensure both content unforgeability (authentication) and message confidentiality while reducing computational complexity. Signcryption is also an important foundation in public-key cryptography which is used in many applications, such as secure email, data sharing, etc. Very recently, several leakage-resilient signcryption schemes with the unbounded leakage property have been proposed (Tseng *et al.*, 2022a, 2023; Tsai *et al.*, 2023) which are based on several public-key systems that include the PKI-PKS, the CL-PKS and certificate-based PKS. In these leakage-resilient signcryption (LRSC) schemes mentioned above, both signers (senders) and decrypters (receivers) belong to the same public-key systems.

Moreover, when signers and decrypters in a signcryption scheme belong to heterogeneous public-key systems, such as signers in the PKI-PKS and decrypters in the CL-

PKS, such a scheme is called as a hybrid signcryption scheme in heterogeneous public-key systems which provides more elastic usage than typical signcryption schemes. In the past, numerous hybrid signcryption schemes in heterogeneous PKSs (including PKI-PKS, ID-PKS and CL-PKS) were proposed, which will be reviewed later. However, until now, there exists no hybrid signcryption scheme with leakage-resilient property. In this paper, our goal is to design the first hybrid signcryption scheme with leakage resilience, called leakage-resilient hybrid signcryption scheme, in heterogeneous public-key systems (LR-HSC-HPKS) from the PKI-PKS to the CL-PKS.

1.2. *Related Work*

In this section, let's review the evolution and development about signcryption schemes and hybrid signcryption schemes in heterogeneous public-key systems.

Based on the PKI-PKS, Zheng (1997) proposed the first signcryption scheme to integrate both signature and encryption schemes into a single scheme to ensure both content authentication and message confidentiality while reducing computational complexity. In 2007, Baek *et al.* (2007) furthermore defined a formal adversary model of signcryption schemes. Indeed, until now, the research on signcryption schemes is still essential for several issues, namely, various public-key systems, security, communication cost and computational complexity. In the past, some signcryption schemes based on various PKSs (PKI-PKS, ID-PKS and CL-PKS) have been proposed, such as PKI-PKS-based (Li *et al.*, 2010), ID-PKS-based (Wei *et al.*, 2015; Karati *et al.*, 2018) and CL-PKS-based (Barbosa and Farshim, 2008; Li *et al.*, 2013a) signcryption schemes.

When signers and decrypters in a signcryption scheme belong to heterogeneous public-key systems, this scheme is called a hybrid signcryption scheme which provides more elastic usage than typical signcryption schemes. In 2010, Sun and Li (2010) proposed the first hybrid signcryption scheme from the PKI-PKS to the ID-PKS. However, Huang *et al.* (2011) pointed out several security drawbacks on Sun and Li's scheme, and proposed an improvement. In the past decade, a large number of hybrid signcryption schemes were proposed, such as hybrid signcryption schemes between the PKI-PKS and the ID-PKS (Li *et al.*, 2013b; Li and Xiong, 2013), hybrid signcryption schemes between the ID-PKS and the CL-PKS (Li *et al.*, 2016a), as well as hybrid signcryption schemes between the PKI-PKS and the CL-PKS (Li *et al.*, 2016b; Liu *et al.*, 2018).

To provide additional properties, several hybrid signcryption schemes were also proposed. Three hybrid signcryption schemes with equality test functionality were proposed, that include Xiong *et al.*'s scheme from the PKI-PKS to the ID-PKS (Xiong *et al.*, 2021), Hou *et al.*'s scheme from the PKI-PKS to the CLC-PKS (Hou *et al.*, 2021) and Xiong *et al.*'s scheme from the ID-PKS to the PKI-PKS (Xiong *et al.*, 2022). A hybrid signcryption schemes with equality test functionality allows users to perform comparative searches on ciphertexts encrypted under different public keys without revealing sensitive data. For the vehicular ad-hoc network (VANET) or Industrial Internet of Things (IIoT) environments, there are four hybrid signcryption schemes that include Ali *et al.*'s scheme from the ID-PKS to the PKI-PKS (Ali *et al.*, 2020), Elkhalil *et al.*'s scheme from the CL-PKS to the

Table 1
Comparisons among the recently proposed hybrid signcryption schemes and our scheme.

Schemes	Signers	Decrypters	Additional property
Xiong *et al.*'s scheme (Xiong *et al.*, 2021)	PKI-PKS	ID-PKS	Equality test functionality
Hou *et al.*'s scheme (Hou *et al.*, 2021)	PKI-PKS	CL-PKS	Equality test functionality
Xiong *et al.*'s scheme (Xiong *et al.*, 2022)	ID-PKS	PKI-PKS	Equality test functionality
Ali *et al.*'s scheme (Ali *et al.*, 2020)	ID-PKS	PKI-PKS	Suitable for VANET environments
Elkhalil *et al.*'s scheme (Elkhalil *et al.*, 2021)	CL-PKS	PKI-PKS	Suitable for VANET environments
Pan *et al.*'s scheme (Pan *et al.*, 2022)	ID-PKS	PKI-PKS	Suitable for VANET environments
Niu *et al.*'s scheme (Niu *et al.*, 2023)	ID-PKS	CL-PKS	Suitable for IIoT environments
Our scheme	PKI-PKS	CL-PKS	Leakage-resilient property

PKI-PKS (Elkhalil *et al.*, 2021) and Pan *et al.*'s scheme from the ID-PKS to the PKI-PKS (Pan *et al.*, 2022) and Niu *et al.*'s scheme from the ID-PKS to the CL-PKS (Niu *et al.*, 2023). Table 1 lists the comparisons among the recently proposed hybrid signcryption schemes and our scheme in terms of the PKS of signers, the PKS of decrypters, and additional properties. We emphasize that our scheme is the first hybrid signcryption scheme with leakage resilience.

1.3. *Contribution*

As mentioned earlier, Tseng *et al.* (2022a) have proposed a PKI-PKS-based leakage-resilient signcryption (LRSC) scheme and Tsai *et al.* (2023) have also proposed a CL-PKS-based LRSC scheme. Based on Tseng *et al.*'s and Tsai *et al.*'s schemes, a new framework of the LR-HSC-HPKS scheme from the PKI-PKS to the CL-PKS is defined. For achieving leakage resilient property of the LR-HSC-HPKS scheme, we employ the key updating process with the multiplicative blinding technique (Kiltz and Pietrzak, 2010; Galindo and Virek, 2013) while partitioning each secret key into two parts. Namely, in the PKI-PKS, the CA's secret key SK_{CA} and the signer ID_{PKI}'s secret key $PKISK_{ID}$ are initially partitioned into $(SK_{CA,0,0}, SK_{CA,0,1})$ and $(PKISK_{ID,0,0}, PKISK_{ID,0,1})$, respectively. In the CL-PKS, the KGC's secret key SK_{KGC} is partitioned into $(SK_{KGC,0,0}, SK_{KGC,0,1})$. Also, the decrypter ID_{CL}'s secret key $CLSK_{ID}$ and identity secret key $CLISK_{ID}$ are initially partitioned into $(CLSK_{ID,0,0}, CLSK_{ID,0,1})$ and $(CLISK_{ID,0,0}, CLISK_{ID,0,1})$, respectively. Meanwhile, each secret key pair must be updated before it is used in each cryptographic computation, namely, the key updating process.

Moreover, two new adversary games of the LR-HSC-HPKS scheme are defined by extending the adversary games of both Tseng *et al.*'s scheme (Tseng *et al.*, 2022a) and Tsai *et al.*'s scheme (Tsai *et al.*, 2023). Based on these two new adversary games under the generic bilinear group (GBG) model (Boneh *et al.*, 2005), security proofs are demonstrated to show that the proposed LR-HSC-HPKS scheme provides both authentication and confidentiality against two types of adversaries in heterogeneous public-key systems. Furthermore, by comparing with several previously proposed hybrid signcryption schemes, the proposed scheme has the following merits: (1) It is the first hybrid signcryption scheme resisting to side-channel attacks. (2) It possesses the unbounded leakage-resilient property, namely, allowing adversaries to repeatedly learn a portion of the secret key used in

each computation. (3) All secret keys of the proposed scheme, (including the CA's secret key SK_{CA}, the signer ID_{PKI}'s secret key $PKISK_{ID}$, the KGC's secret key SK_{KGC}, and the decrypter ID_{CL}'s secret key $CLSK_{ID}$ and identity secret key $CLISK_{ID}$), are allowed to be leaked to adversaries while remaining the security of the proposed scheme. Finally, by the performance experiences on both a PDA and a PC, performance analysis is demonstrated to show that our scheme is well suitable for running on both a PDA and a PC.

1.4. *Paper Structure*

The rest of this paper is structured as follows. In Section 2, several preliminary contents are introduced. In Section 3, we define a new framework and two new adversary games for the LR-HSC-HPKS scheme. The LR-HSC-HPKS scheme is presented in Section 4. The proofs of two security theorems are shown in Section 5. Section 6 conducts the performance analysis on a PC and a PDA. In Section 7, the conclusions and future work are given.

2. Preliminaries

2.1. *Bilinear Groups and GBG Model*

Let $G = \langle Q \rangle$ and $G_1 = \langle Q_1 \rangle$ be, respectively, an additive group and a multiplicative group with the same prime order q, where Q and Q_1 are generators of G and G_1, respectively. Meanwhile, the bilinear pairing operation $\hat{e} : G \times G \to G_1$ is admissible, if it satisfies three conditions below:

- Bilinearity: for $u, v \in Z_q^*$, $\hat{e}(u \cdot Q, v \cdot Q) = \hat{e}(Q, Q)^{uv}$.
- Non-degeneration: $Q_1 = \hat{e}(Q, Q) \neq 1$.
- Computation: for $u, v \in Z_q^*$, $\hat{e}(u \cdot Q, v \cdot Q)$ can be computed efficiently.

Finally, let $\{G, G_1, \hat{e}, Q, Q_1, q\}$ represent a bilinear group set. The reader can refer to [BF-01] for detailed parameter settings.

Boneh *et al.* (2005) introduced a method for security proof, called the generic bilinear group (GBG) model, which is operated on a bilinear group set $\{G, G_1, \hat{e}, Q, Q_1, q\}$. Meanwhile, the GBG model is combined into adversary games for security properties. In such adversary games, there is an adversary and a challenger who, respectively, are an oracle (query) requester and a replier. To run the operations on a bilinear group set $\{G, G_1, \hat{e}, Q, Q_1, q\}$, the adversary requests the corresponding oracles (queries) and receives the operation results from the challenger. Therefore, the adversary may request three oracles O_a, O_m and $O_{\hat{e}}$, which are, respectively, the additive operation on G, the multiplicative operation on G_1 and the operation $\hat{e} : G \times G \to G_1$. Two injective random encoding functions $\xi : Z_q^* \to \Omega G$ and $\xi_1 : Z_q^* \to \Omega G_1$, are used to map all the elements of G and G_1 to distinct bit strings, respectively, which satisfy both $\Omega G \cap \Omega G_1 = \phi$ and $|\Omega G| = |\Omega G_1| = q$. Additionally, for all $u, v \in Z_q^*$, three oracles O_a, O_m and $O_{\hat{e}}$ have the following operation properties;

– $O_a(\xi(u), \xi(v)) \rightarrow \xi(u + v \bmod q)$;
– $O_m(\xi_1(u), \xi_1(v)) \rightarrow \xi_1(u + v \bmod q)$;
– $O_{\hat{e}}(\xi(u), \xi(v) \rightarrow \xi_1(u \cdot v \bmod q)$.

Note that Q is represented by $\xi(1)$, whereas $\xi_1(1)$ represents $Q_1 = \hat{e}(Q, Q)$. When such an adversary game ends and the adversary finds collisions in G or G_1, the discrete logarithm problem in G or G_1 will be resolved, respectively.

2.2. *Security Assumptions and Entropy*

In this section, we define two security assumptions on which the proposed scheme is based as follows:

– **Discrete logarithm (DL) assumption**: In $\{G, G_1, \hat{e}, Q, Q_1, q\}$, for given $u \cdot Q \in G$ or $Q_1^u \in G_1$, without knowing $u \in Z_q^*$, it is hard to discover u.
– **Secure hash function (SH) assumption**: Let $SH : \{0, 1\}^* \rightarrow \{0, 1\}^t$ be a secure hash function, where t is a fixed length. Then it is hard to discover any two random bit strings RBS_1 and RBS_2 such that $SH(RBS_1) = SH(RBS_2)$.

For evaluating the leakage impact of secret keys incurred by side-channel attacks, we employ the entropy concept by which the secret keys are viewed as finite random variables. Also, two consequences below (Lemmas 1 and 2) have been conducted in the literature (Dodis *et al.*, 2008; Galindo and Virek, 2013).

Lemma 1. *Let SK and $LF : SK \rightarrow \{0, 1\}^\tau$, respectively, denote a secret key and the corresponding leak function, where τ is a fixed length. Under the leak function $LF()$, we have $\widetilde{H}_\infty(SK|LF(SK)) \geqq H_\infty(SK) - \tau$, where $\widetilde{H}$ and H_∞ are, respectively, the average conditional min-entropy and the min-entropy.*

Lemma 2. *Assume that there is a multiple-secret-key polynomial $MSKF \in Z_q[SK_0, SK_1, \ldots, SK_{n-1}]$ with degree d, where $SK_0, SK_1, \ldots, SK_{n-1}$ are secret keys. Let P_i (for $i = 0, 1, \ldots, n-1$) be n mutually independent probability distributions $SK_i = sk_i \leftarrow Z_q$, which satisfy $0 \leqq \tau \leqq \log q$ and $H_\infty(P_i) \geqq \log q - \tau$. Then the probability $\text{Pb}[MSKF(SK_0 = sk_0, SK_1 = sk_1, \ldots, SK_{n-1} = sk_{n-1}) = 0] \leqq 2^\tau(d/q)$ is negligible if $\tau < (1 - \omega) \log q$, where ω denotes a positive fraction.*

3. Framework and Adversary Games

In this section, we define the framework and adversary games of the LR-HSC-HPKS scheme. For readability, some notations used throughout this paper are first defined in Table 2.

3.1. *Framework*

Based on Tseng *et al.*'s scheme (Tseng *et al.*, 2022a) and Tsai *et al.*'s scheme (Tsai *et al.*, 2023), we define a new framework of the LR-HSC-HPKS scheme. In the heterogeneous

Table 2
Notations.

Notation	Meaning
CA	A certificate authority in the PKI-PKS
KGC	A key generation centre in the CL-PKS
SK_{CA}/PK_{CA}	CA's secret/public key pair
SK_{KGC}/PK_{KGC}	KGC's secret/public key pair
ID_{PKI}	The identity of a user in the PKI-PKS
$PKISK_{ID}/PKIPK_{ID}$	The secret/public key pair of the user ID_{PKI}
CRT_{ID}	The certificate of the user ID_{PKI}
ID_{CL}	The identity of a user in the CL-PKS
$CLSK_{ID}/CLPK_{ID}$	The secret/public key pair of the user ID_{CL}
$CLISK_{ID}/CLIPK_{ID}$	The identity secret/public key pair of the user ID_{CL}
M	A message
CT	A ciphertext
SP	The system parameters
HSE	The Hybrid signcryption in the LR-HSC-HPKS scheme
$HUSE$	The Hybrid unsigncryption in the LR-HSC-HPKS scheme

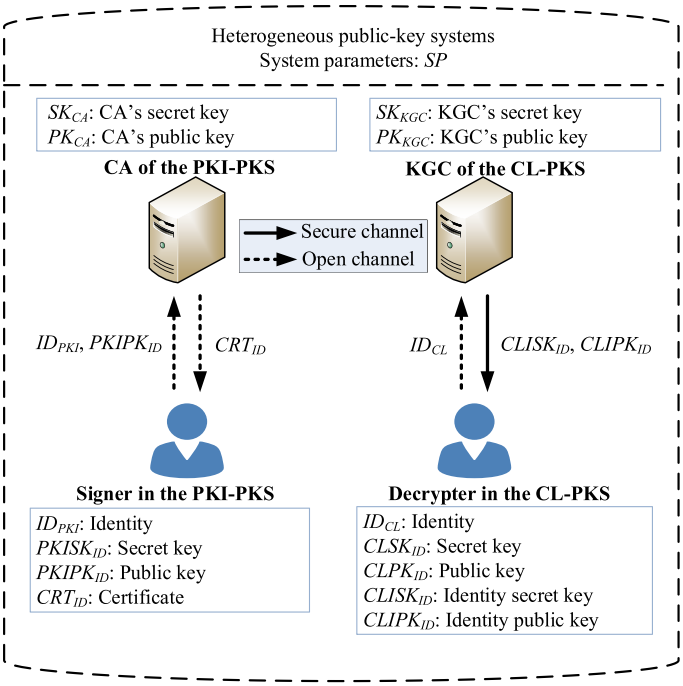

Fig. 1. Two key generating procedures of the LR-HSC-HPKS scheme.

public-key systems, there are two public-key systems (PKSs), namely, the public-key infrastructure PKS (PKI-PKS) and the certificateless PKS (CL-PKS). In the LR-HSC-HPKS scheme, signers and decrypeters belong to the PKI-PKS and the CL-PKS, respectively. Here, two key generating procedures of the LR-HSC-HPKS scheme are presented in Fig 1. In the PKI-PKS, a signer with identity ID_{PKI} randomly selects a secret key $PKISK_{ID}$ and

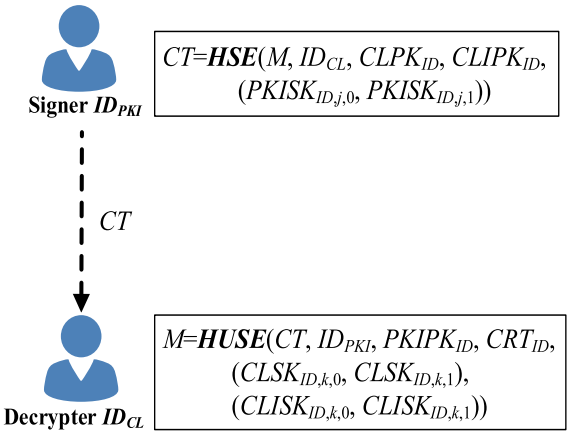

Fig. 2. The inputs/outputs of the *HSE* and the *HUSE* algorithms in the LR-HSC-HPKS scheme.

computes the associated public key $PKIPK_{ID}$. The signer sends both ID_{PKI} and $PKIPK_{ID}$ to a trusted certificate authority (CA) with a key pair of a secret key SK_{CA} and the associated public key PK_{CA}. Then, the CA uses SK_{CA} to compute and return the certificate CRT_{ID} to the signer ID_{PKI}. In the CL-PKS, a decrypter with identity ID_{CL} randomly selects a secret key $CLSK_{ID}$ and computes the associated public key $CLPK_{ID}$. The decrypter sends ID_{CL} to a key generation centre (KGC) with a key pair of a secret key SK_{KGC} and the associated public key PK_{KGC}. Then, the KGC uses SK_{KGC} to compute and return the decrypter ID_{CL}'s identity secret key $CLISK_{ID}$ and identity public key $CLIPK_{ID}$.

For achieving leakage resilient property of the LR-HSC-HPKS scheme, we employ the key updating process with the multiplicative blinding technique (Kiltz and Pietrzak, 2010; Galindo and Virek, 2013) while partitioning each secret key into two parts. Meanwhile, each secret key must be updated before it is used in each cryptographic computation, namely, the key updating process. In the PKI-PKS, the CA's secret key SK_{CA} and the signer ID_{PKI}'s secret key $PKISK_{ID}$ are initially partitioned into ($SK_{CA,0,0}$, $SK_{CA,0,1}$) and ($PKISK_{ID,0,0}$, $PKISK_{ID,0,1}$), respectively. In the CL-PKS, the KGC's secret key SK_{KGC} is partitioned into ($SK_{KGC,0,0}$, $SK_{KGC,0,1}$). Also, the decrypter ID_{CL}'s secret key $CLSK_{ID}$ and identity secret key $CLISK_{ID}$ are initially partitioned into ($CLSK_{ID,0,0}$, $CLSK_{ID,0,1}$) and ($CLISK_{ID,0,0}$, $CLISK_{ID,0,1}$), respectively.

In the LR-HSC-HPKS scheme, assume that a signer ID_{PKI} runs the *Hybrid signcryption* (*HSE*) algorithm to transmit a message M to a decrypter ID_{CL}. For the *HSE* algorithm's j-th running, the signer ID_{PKI} first updates the old secret key ($PKISK_{ID,j-1,0}$, $PKISK_{ID,j-1,1}$) to the new secret key ($PKISK_{ID,j,0}$, $PKISK_{ID,j,1}$) and sends a ciphertext $CT = HSE(M, ID_{CL}, CLPK_{ID}, CLIPK_{ID}, (PKISK_{ID,j,0}, PKISK_{ID,j,1}))$ to the decrypter ID_{CL}. For the *Hybrid unsigncryption* (*HUSE*) algorithm's k-th running and receiving CT, the decrypter ID_{CL} first updates the old secret key ($CLSK_{ID,k-1,0}$, $CLSK_{ID,k-1,1}$) to the new identity secret key ($CLISK_{ID,k,0}$, $CLISK_{ID,k,1}$), and gets the message $M = HUSE(CT, ID_{PKI}, PKIPK_{ID}, CRT_{ID}, (CLSK_{ID,k,0}, CLSK_{ID,k,1}), (CLISK_{ID,k,0}, CLISK_{ID,k,1}))$. Figure 2 depicts the inputs/outputs of the *HSE* and the

HUSE algorithms in the LR-HSC-HPKS scheme. A new framework of the LR-HSC-HPKS scheme from the PKI-PKS to the CL-PKS is presented in Definition 1.

Definition 1. The LR-HSC-HPKS scheme includes the following four parts.

- System setup: Firstly, the system parameters (SP) are initially set. The heterogeneous public-key systems consist of the PKI-PKS and the CL-PKS. The CA in the PKI-PKS and the KGC in the CL-PKS, respectively, set their secret keys and the associated public keys as follows.
  - ♦ PKI-PKS: The CA sets a secret/public key pair (SK_{CA}, PK_{CA}). Initially, the CA partitions SK_{CA} into ($SK_{CA,0,0}$, $SK_{CA,0,1}$).
  - ♦ CL-PKS: The KGC sets a secret/public key pair (SK_{KGC}, PK_{KGC}). Initially, the KGC partitions SK_{KGC} into ($SK_{KGC,0,0}$, $SK_{KGC,0,1}$).

  Also, SP, PK_{CA} and PK_{KGC} are publicly published.
- User key generation: For signers in the PKI-PKS and decrypters in the CL-PKS, two key generating procedures are presented as follows.
  - ♦ PKI-PKS: A signer with identity ID_{PKI} and the CA cooperatively run the following two algorithms.
    - Signer secret key generation: The signer ID_{PKI} sets a secret/public key pair ($PKISK_{ID}$, $PKIPK_{ID}$). Initially, the signer ID_{PKI} partitions $PKISK_{ID}$ into ($PKISK_{ID,0,0}$, $PKISK_{ID,0,1}$). Also, the signer ID_{PKI} sends (ID_{PKI}, $PKIPK_{ID}$) to the CA.
    - Signer certificate generation: For this algorithm's i-th running and giving (ID_{PKI}, $PKIPK_{ID}$), the CA first updates the old secret key ($SK_{CA,i-1,0}$, $SK_{CA,i-,1,1}$) to the new secret key ($SK_{CA,i,0}$, $SK_{CA,i,1}$), such that $SK_{CA} = SK_{CA,0,0}+SK_{CA,0,1} = SK_{CA,1,0} + SK_{CA,1,1} = \cdots = SK_{CA,i,0} + SK_{CA,i,1}$. Subsequently, the CA uses ($SK_{CA,i,0}$, $SK_{CA,i,1}$) to compute and return the certificate CRT_{ID} to the signer ID_{PKI}.
  - ♦ CL-PKS: A decrypter with identity ID_{CL} and the KGC cooperatively run the following four algorithms.
    - Decrypter secret key generation: The decrypter ID_{CL} sets a secret/public key pair ($CLSK_{ID}$, $CLPK_{ID}$). Also, the decrypter ID_{CL} sends ID_{CL} to the KGC.
    - Decrypter identity secret key generation: For this algorithm's i-th running and giving ID_{CL}, the KGC first updates the old secret key ($SK_{KGC,i-1,0}$, $SK_{KGC,i-1,1}$) to the new secret key ($SK_{KGC,i,0}$, $SK_{KGC,i,1}$) such that $SK_{KGC} = SK_{KGC,0,0} + SK_{KGC,0,1} = SK_{KGC,1,0} + SK_{KGC,1,1} = \cdots = SK_{KGC,i,0} + SK_{KGC,i,1}$. Subsequently, the KGC uses ($SK_{KGC,i,0}$, $SK_{KGC,i,1}$) to compute and return the identity secret/public key pair ($CLISK_{ID}$, $CLIPK_{ID}$) to the decrypter ID_{CL}.
    - Decrypter secret key combination: ($CLSK_{ID}$, $CLISK_{ID}$) is the decrypter ID_{CL}'s secret key pair. Initially, the decrypter ID_{CL} partitions $CLSK_{ID}$ and $CLISK_{ID}$ into ($CLSK_{ID,0,0}$, $CLSK_{ID,0,1}$) and ($CLISK_{ID,0,0}$, $CLISK_{ID,0,1}$), respectively.
    - Decrypter public key combination: ($CLPK_{ID}$, $CLIPK_{ID}$) is the decrypter ID_{CL}'s public key pair.

Hybrid signcryption (*HSE*): For the *HSE* algorithm's j-th running and giving $(M, ID_{CL}, CLPK_{ID}, CLIPK_{ID})$, the signer ID_{PKI} first updates the old secret key $(PKISK_{ID,j-1,0}, PKISK_{ID,j-1,1})$ to the new secret key $(PKISK_{ID,j,0}, PKISK_{ID,j,1})$. Then, the signer ID_{PKI} generates a ciphertext $CT = HSE(M, ID_{CL}, CLPK_{ID}, CLIPK_{ID}, (PKISK_{ID,j,0}, PKISK_{ID,j,1}))$ and returns CT to the decrypter ID_{CL}.

– *Hybrid unsigncryption* (*HUSE*): For the *Hybrid unsigncryption* (*HUSE*) algorithm's k-th running and giving CT, the decrypter ID_{CL}, respectively, updates the old secret key $(CLSK_{ID,k-1,0}, CLSK_{ID,k-1,1})$ and the identity secret key $(CLISK_{ID,k-1,0}, CLISK_{ID,k-1,1})$ to the new secret key $(CLSK_{ID,k,0}, CLSK_{ID,k,1})$ and the new identity secret key $(CLISK_{ID,k,0}, CLISK_{ID,k,1})$, and gets the message $M = HUSE(CT, ID_{PKI}, PKIPK_{ID}, CRT_{ID}, (CLSK_{ID,k,0}, CLSK_{ID,k,1}), (CLISK_{ID,k,0}, CLISK_{ID,k,1}))$.

3.2. *Adversary Games*

Based on Tseng *et al.*'s scheme (Tseng *et al.*, 2022a) and Tsai *et al.*'s scheme (Tsai *et al.*, 2023), we define two adversary games of the LR-HSC-HPKS scheme in the heterogeneous public-key systems (including the PKI-PKS and the CL-PKS).

For the *Signer certificate generation* i-th running, a pair of leak functions $(f_{SCG,i}, h_{SCG,i})$ on $(SK_{CA,i,0}, SK_{CA,i,1})$ is employed to model the leak ability of adversaries. Also, the pair $(f_{ISKG,i}, h_{ISKG,i})$ on $(SK_{KGC,i,0}, SK_{KGC,i,1})$ is employed for *Decrypter identity secret key generation*'s i-th running, the pair $(f_{HS,j}, h_{HS,j})$ on $(PKISK_{ID,j,0}, PKISK_{ID,j,1})$ is employed for *Hybrid signcryption*'s j-th running and the pair $(f_{HUS,k}, h_{HUS,k})$ on $((CLSK_{ID,k,0}, CLISK_{ID,k,0}), (CLSK_{ID,k,1}, CLISK_{ID,k,1}))$ is employed for *Hybrid unsigncryption*'s k-th running. Moreover, let $\Delta f_{SCG,i}$, $\Delta h_{SCG,i}$, $\Delta f_{ISKG,i}$, $\Delta h_{ISKG,i}$, $\Delta f_{HS,j}$, $\Delta h_{HS,j}$, $\Delta f_{HUS,k}$ and $\Delta h_{HUS,k}$ denote these functions' outputs while each output bit length is limited to τ as defined in Lemma 1. The inputs and outputs of eight leak functions are given as follows:

– $\Delta f_{SCG,i} = f_{SCG,i}(SK_{CA,i,0})$.
– $\Delta h_{SCG,i} = h_{SCG,i}(SK_{CA,i,1})$.
– $\Delta f_{ISKG,i} = f_{ISKG,i}(SK_{KGC,i,0})$.
– $\Delta h_{ISKG,i} = h_{ISKG,i}(SK_{KGC,i,1})$.
– $\Delta f_{HS,j} = f_{HS,j}(PKISK_{ID,j,0})$.
– $\Delta h_{HS,j} = h_{HS,j}(PKISK_{ID,j,1})$.
– $\Delta f_{HUS,k} = f_{HUS,k}(CLSK_{ID,k,0}, CLISK_{ID,k,0})$.
– $\Delta h_{HUS,k} = h_{HUS,k}(CLSK_{ID,k,1}, CLISK_{ID,k,1})$.

In the heterogeneous public-key systems (including the PKI-PKS and the CL-PKS), there are two types of adversaries, namely, illegitimate member (A_I) and malicious CA/KGC (A_{II}).

– Illegitimate member (A_I): A_I is used to model the attacking abilities of an illegitimate member as follows.

- A_I may obtain any signer ID_{PKI}'s secret key $PKISK_{ID}$, except for the target signer ${ID^*}_{PKI}$. Also A_I may obtain any decrypter ID_{CL}'s secret key $CLSK_{ID}$ and identity secret key $CLISK_{ID}$, except for the identity secret key $CLISK_{ID^*}$ of the target decrypter ${ID^*}_{CL}$.
- A_I may obtain a portion about $PKISK_{ID^*} = (PKISK_{ID^*,j,0}, PKISK_{ID^*,j,1})$ and $CLISK_{ID^*} = (CLISK_{ID^*,k,0}, CLISK_{ID^*,k,1})$ by two pairs of leak functions $(f_{HS,j}, h_{HS,j})$ and $(f_{HUS,k}, h_{HUS,k})$, respectively.
- A_I may obtain a portion of $SK_{CA} = (SK_{CA,i,0}, SK_{CA,i,1})$ and $SK_{KGC} = (SK_{KGC,i,0}, SK_{KGC,i,1})$ by two pairs of leak functions $(f_{SCG,i}, h_{SCG,i})$ and $(f_{ISKG,i}, h_{ISKG,i})$, respectively.

– Malicious CA/KGC (A_{II}): A_{II} is used to model the attacking abilities of a malicious CA/KGC who has both SK_{CA} and SK_{KGC}.
  - A_{II} may obtain any signer ID_{PKI}'s secret key $PKISK_{ID}$ and any decrypter ID_{CL}'s secret key $CLSK_{ID}$, except for the target signer ${ID^*}_{PKI}$ and decrypter ${ID^*}_{CL}$.
  - A_{II} may obtain a portion of $PKISK_{ID^*} = (PKISK_{ID^*,j,0}, PKISK_{ID^*,j,1})$ by the pair of leak functions $(f_{HS,j}, h_{HS,j})$.
  - A_{II} may obtain a portion of $CLSK_{ID^*} = (CLSK_{ID^*,k,0}, CLSK_{ID^*,k,1})$ by the pair of leak functions $(f_{HUS,k}, h_{HUS,k})$.

In Definitions 2 and 3, we define two adversary games $Game_1$ and $Game_2$ to model the content unforgeability (authentication) and the message confidentiality, respectively.

Definition 2 (**Game$_1$**). The adversary game $Game_1$ is played by an adversary A (A_I or A_{II}) and a challenger B. If no probabilistic polynomial-time (PPT) adversary A with a non-negligible advantage wins $Game_1$, the LR-HSC-HPKS scheme possesses the existential unforgeability (authentication) under adaptive chosen-message and side-channel attacks (EUF-ACMSCA).

– *Initialization phase*: The challenger B runs the *System setup* in Definition 1 to generate the CA's secret/public key pair (SK_{CA}, PK_{CA}) and the KGC's secret/public key pair (SK_{KGC}, PK_{KGC}). Also, B sets the system parameters (SP). In the meantime, B partitions SK_{CA} and SK_{KGC} into $(SK_{CA,0,0}, SK_{CA,0,1})$ and $(SK_{KGC,0,0}, SK_{KGC,0,1})$, respectively. Additionally, if A is an A_{II}, both SK_{CA} and SK_{KGC} are sent to A_{II}.
– *Query phase*: A (A_I or A_{II}) may adaptively request various kinds of queries (oracles) to B as follows.
  - *Signer secret key query* (ID_{PKI}): The signer ID_{PKI}'s secret key $PKISK_{ID}$ is returned.
  - *Signer certificate query* $(ID_{PKI}, PKIPK_{ID})$. For the i-th request of this query, B first updates the old secret key $(SK_{CA,i-1,0}, SK_{CA,i-1,1})$ to the new secret key $(SK_{CA,i,0}, SK_{CA,i,1})$. By $(ID_{PKI}, PKIPK_{ID})$, B uses $(SK_{CA,i,0}, SK_{CA,i,1})$ to generate and return the signer ID_{PKI}'s certificate CRT_{ID}.
  - *Signer certificate leak query* $(i, f_{SCG,i}, h_{SCG,i})$. For the i-th request of the *Signer certificate query*, the leak query can only be requested once. B returns $\Delta f_{SCG,i} = f_{SCG,i}(SK_{CA,i,0})$ and $\Delta h_{SCG,i} = h_{SCG,i}(SK_{CA,i,1})$.
  - *Decrypter identity secret key query* (ID_{CL}). For the i-th request of this query, B first updates the old secret key $(SK_{KGC,i-1,0}, SK_{KGC,i-1,1})$ to the new secret key

$(SK_{KGC,i,0}, SK_{KGC,i,1})$. By ID_{CL}, B uses $(SK_{KGC,i,0}, SK_{KGC,i,1})$ to generate and return the identity secret/public key pair $(CLISK_{ID}, CLIPK_{ID})$.

- *Decrypter identity secret key leak query* $(i, f_{ISKG,i}, h_{ISKG,i})$. For the i-th request of the *Decrypter identity secret key query*, the leak query can only be requested once. B returns $\Delta f_{ISKG,i} = f_{ISKG,i}(SK_{KGC,i,0})$ and $\Delta h_{ISKG,i} = h_{ISKG,i}(SK_{KGC,i,1})$.
- *Decrypter public key replace query* $(ID_{CL}, (CLPK'_{ID}, CLIPK'_{ID}))$. The decrypter ID_{CL}'s public key is replaced with $(CLPK'_{ID}, CLIPK'_{ID})$.
- *Decrypter secret key query* (ID_{CL}). If the *Decrypter public key replace query* $(ID_{CL}, (CLPK'_{ID}, CLIPK'_{ID}))$ is never requested, the decrypter ID_{CL}'s secret key $CLSK_{ID}$ is returned.
- *Hybrid signcryption query* (M, ID_{PKI}, ID_{CL}): B first updates the signer ID_{PKI}'s old secret key $(PKISK_{ID,j-1,0}, PKISK_{ID,j-1,1})$ to the new secret key $(PKISK_{ID,j,0}, PKISK_{ID,j,1})$, and runs the *Hybrid signcryption* to return CT.
- *Hybrid signcryption leak query* $(ID_{PKI}, j, f_{HS,j}, h_{HS,j})$: For the signer ID_{PKI}'s j-th request of the *Decrypter identity secret key query*, the leak query can only be requested once. B returns $\Delta f_{HS,j} = f_{HS,j}(PKISK_{ID,j,0})$ and $\Delta h_{HS,j} = h_{HS,j}(PKISK_{ID,j,1})$.
- *Hybrid unsigncryption query* (CT, ID_{PKI}, ID_{CL}): B first updates the decrypter ID_{CL}'s old secret key $(CLSK_{ID,k-1,0}, CLSK_{ID,k-1,1})$ and identity secret key $(CLISK_{ID,k-1,0}$, $CLISK_{ID,k-1,1})$ to the new secret key $(CLSK_{ID,k,0}, CLSK_{ID,k,1})$ and identity secret key $(CLISK_{ID,k,0}, CLISK_{ID,k,1})$, respectively. B runs the *Hybrid unsigncryption* to return M.
- *Hybrid unsigncryption leak query* $(ID_{CL}, k, f_{HUS,k}, h_{HUS,k}$: For the decrypter ID_{CL}'s k-th request of the *Decrypter identity secret key query*, the leak query can only be requested once. B returns $\Delta f_{HUS,k} = f_{HUS,k}(CLSK_{ID,k,0}, CLSK_{ID,k,1})$ and $\Delta h_{HUS,k} = h_{HUS,k}(CLISK_{ID,k,0}, CLISK_{ID,k,1})$.

– *Forgery phase*: Assume that A forges a ciphertext $CT^* = (T^*_0, T^*_1, T^*_2, ID^*_{PKI}, ID^*_{CL})$ for the message M^*. We say that A wins $Game_1$ if the following three provisions are true.
  - M^* can be generated by the *Hybrid unsigncryption* algorithm.
  - The *Hybrid signcryption query* $(M^*, ID^*_{PKI}, ID_{CL})$ is never issued.
  - The *Signer secret key query* (ID^*_{PKI}) is never issued.

Definition 3 (**Game$_2$**). The adversary game $Game_2$ is played by an adversary A (A_I or A_{II}) and a challenger B. If no PPT adversary A with a non-negligible advantage wins $Game_2$, the LR-HSC-HPKS scheme possesses the encryption indistinguishability (message confidentiality) under chosen-ciphertext and side-channel attacks (EIND-CCSCA).

– *Initialization phase*. The phase is the same with the *Initialization phase* in Definition 2.
– *Query phase*. The phase is the same with the *Query phase* in Definition 2.
– *Challenge phase*. A selects a target decrypter ID^*_{CL} and a message pair (M_0, M_1) as a challenge objective. B randomly selects $c \in \{0, 1\}$ and generates a challenge ciphertext CT^* by running the Hybrid signcryption with $(M_c, ID_{PKI}, ID^*_{CL})$. Also, B sends CT^* to A. Note that the following two provisions are true.

1. If A is an A_I, the *Decrypter identity secret key query* (ID^*_{CL}) is never issued.
2. If A is an A_{II}, neither the *Decrypter public key replace query* (ID^*_{CL}, ($CLPK'_{ID^*}$, $CLIPK'_{ID^*}$)) nor the *Decrypter secret key query* (ID^*_{CL}) is issued.

– *Guessing phase*. A outputs $c' \in \{0, 1\}$ and wins $Game_2$ if $c' = c$. Meanwhile, A's advantage is defined as $Adv(A) = |\text{Pb}[c' = c] - 1/2|$.

4. Our LR-HSC-HPKS Scheme

According to the framework shown in Definition 1, our LR-HSC-HPKS scheme consists of four parts as presented below.

– *System setup*: The system sets a bilinear group set $\{G, G_1, \hat{e}, Q, Q_1, q\}$ defined in Section 2.1. Moreover, the system publishes $SP = \{G, G_1, \hat{e}, Q, Q_1, q, W, T, SE/SD, SH_0, SH_1\}$, where W and T are random elements in G, SE and SD are respectively symmetric encryption and decryption functions, and $SH_0 : \{0, 1\}^* \times G \to \{0, 1\}^t$ and $SH_1 : G \times \{0, 1\}^* \to \{0, 1\}^t$ are two secure hash functions. The heterogeneous public-key systems consist of the PKI-PKS and the CL-PKS. The CA in the PKI-PKS and the KGC in the CL-PKS, respectively, set their secret/public key pairs as follows.
  - PKI-PKS: The CA randomly selects $r \in {Z_q}^*$ and then sets a secret/public key pair (SK_{CA}, PK_{CA}), where $SK_{CA} = r \cdot Q$ and $PK_{CA} = \hat{e}(Q, r \cdot Q)$. Also, the CA randomly selects $w \in {Z_q}^*$ and partitions SK_{CA} into $SK_{CA} = (SK_{CA,0,0}, SK_{CA,0,1}) = (w \cdot Q, SK_{CA} - w \cdot Q)$.
  - CL-PKS: The KGC randomly selects $t \in {Z_q}^*$ and then sets a secret/public key pair (SK_{KGC}, PK_{KGC}), where $SK_{KGC} = t \cdot Q$ and $PK_{KGC} = \hat{e}(Q, t \cdot Q)$. Also, the KGC randomly selects $s \in {Z_q}^*$ and partitions SK_{KGC} into $SK_{KGC} = (SK_{KGC,0,0}, SK_{KGC,0,1}) = (s \cdot Q, SK_{KGC} - s \cdot Q)$.

– *User key generation*: For signers in the PKI-PKS and decrypters in the CL-PKS, two key generating procedures are presented as follows.
  - PKI-PKS: A signer with identity ID_{PKI} and the CA cooperatively run the following two algorithms.
    - *Signer secret key generation*: The signer ID_{PKI} randomly selects $x \in {Z_q}^*$ and then sets a secret/public key pair $(PKISK_{ID}, PKIPK_{ID})$, where $PKISK_{ID} = x \cdot Q$ and $PKIPK_{ID} = \hat{e}(Q, x \cdot Q)$. Also, the signer ID_{PKI} randomly selects $w_i \in {Z_q}^*$ and partitions $PKISK_{ID}$ into $PKISK_{ID} = (PKISK_{ID,0,0}, PKISK_{ID,0,1}) = (w_i \cdot Q, PKISK_{ID} - w_i \cdot Q)$.
    - *Signer certificate generation*: For this algorithm's i-th running and giving $(ID_{PKI}, PKIPK_{ID})$, the CA randomly selects $w \in {Z_q}^*$ and updates the old secret key $(SK_{CA,i-1,0}, SK_{CA,i-,1,1})$ to the new secret key $(SK_{CA,i,0}, SK_{CA,i,1}) = (SK_{CA,i-1,0} + w \cdot Q, SK_{CA,i-1,1} - w \cdot Q)$, such that $SK_{CA} = SK_{CA,0,0} + SK_{CA,0,1} = SK_{CA,1,0} + SK_{CA,1,1} = \cdots = SK_{CA,i,0} + SK_{CA,i,1}$. Also, the CA uses $(SK_{CA,i,0}, SK_{CA,i,1})$ to compute and return the certificate CRT_{ID} to the signer ID_{PKI}.

♦ CL-PKS: A decrypter with identity ID_{CL} and the KGC cooperatively run the following four algorithms.

- Decrypter secret key generation: The decrypter ID_{CL} randomly selects $l \in Z_q^*$ and then sets a secret/public key pair ($CLSK_{ID}$, $CLPK_{ID}$), where $CLSK_{ID} = l \cdot Q$ and $CLPK_{ID} = \hat{e}(Q, l \cdot Q)$. Also, the decrypter ID_{CL} sends ID_{CL} to the KGC.
- *Decrypter identity secret key generation*: For this algorithm's i-th running and giving ID_{CL}, the KGC randomly selects $t_i \in Z_q^*$ and updates the old secret key ($SK_{KGC,i-1,0}$, $SK_{KGC,i-1,1}$) to the new secret key ($SK_{KGC,i,0}$, $SK_{KGC,i,1}$) = ($SK_{KGC,i-1,0} + t_i \cdot Q$, $SK_{KGC,i-1,1} - t_i \cdot Q$), such that $SK_{KGC} = SK_{KGC,0,0} + SK_{KGC,0,1} = SK_{KGC,1,0} + SK_{KGC,1,1} = \cdots = SK_{KGC,i,0} + SK_{KGC,i,1}$. Also, the KGC randomly selects $f \in Z_q^*$ and uses ($SK_{KGC,i,0}$, $SK_{KGC,i,1}$) to compute and return the identity secret/public key pair ($CLISK_{ID}$, $CLIPK_{ID}$) of the decrypter ID_{CL} as follows:

  (1) $CLIPK_{ID} = f \cdot Q$.
  (2) $\rho = SH_0(ID_{CL}, CLIPK_{ID})$.
  (3) $TK_i = SK_{KGC,i,1} + f \cdot (W + \rho \cdot T)$.
  (4) $CLISK_{ID} = SK_{KGC,i,0} + TK_i$.

- *Decrypter secret key combination*: The decrypter ID_{CL}'s secret key pair is ($CLSK_{ID}$, $CLISK_{ID}$). The ID_{CL} randomly selects δ,$\xi \in Z_q^*$, and partitions $CLSK_{ID}$ and $CLISK_{ID}$ into ($CLSK_{ID,0,0}$, $CLSK_{ID,0,1}$) = ($\delta \cdot Q$, $CLSK_{ID} - \delta \cdot Q$) and ($CLISK_{ID,0,0}$, $CLISK_{ID,0,1}$) = ($\xi \cdot Q$, $CLISK_{ID} - \xi \cdot Q$), respectively.
- *Decrypter public key combination*: The decrypter ID_{CL}'s public key pair is ($CLPK_{ID}$, $CLIPK_{ID}$).

– *Hybrid signcryption* (*HSE*): Assume that the signer ID_{PKI} wants to send a message M to the decrypter ID_{CL}. For the *HSE* algorithm's j-th running, the signer ID_{PKI} runs the following steps to generate a ciphertext CT.

(1) Randomly select $h \in Z_q^*$ and update the old secret key ($PKISK_{ID,j-1,0}$, $PKISK_{ID,j-1,1}$) into the new secret key ($PKISK_{ID,j,0}$, $PKISK_{ID,j,0}$) = ($PKISK_{ID,j-1,0} + h \cdot Q$, $PKISK_{ID,j-1,1} - h \cdot Q$).
(2) Randomly select $n \in Z_q^*$, and compute $T_1 = n \cdot Q$, $EK_1 = (CLPK_{ID})^n$, $EK_2 = (PK_{KGC} \cdot \hat{e}(CLIPK_{ID}, (W + \rho \cdot T)))^n$, where $\rho = SH_0(ID_{CL}, CLIPK_{ID})$.
(3) Generate $T_2 = SE_{EK}(M)$, where $EK = EK_1 \oplus EK_2$ is an encryption key.
(4) Compute $TS = PKISK_{ID,j,0} + (n \cdot (W + \beta \cdot T))$, where $\beta = SH_1(T_1, T_2, ID_{PKI}, ID_{CL}, M)$.
(5) Generate a signature $T_0 = PKISK_{ID,j,1} + TS$.
(6) Set $CT = (T_0, T_1, T_2, ID_{PKI}, ID_{CL})$.

– *Hybrid unsigncrypion* (*HUSE*): For the *Hybrid unsigncryption* (*HUSE*) algorithm's k-th running and giving CT, the decrypter ID_{CL} runs the following steps to get the message M.

(1) Randomly select $v \in Z_q^*$, and update the old secret key ($CLSK_{ID,k-1,0}$, $CLSK_{ID,k-1,1}$) and the old identity secret key ($CLISK_{ID,k-1,0}$, $CLISK_{ID,k-1,1}$)

to the new secret key $(CLSK_{ID,k,0}, CLSK_{ID,k,1}) = (CLSK_{ID,k-1,0} + v \cdot Q, CLSK_{ID,k-1,1} - v \cdot Q)$ and the new identity secret key $(CLISK_{ID,k,0}, CLISK_{ID,k,1}) = (CLISK_{ID,k-1,0} + v \cdot Q, CLISK_{ID,k-1,1} - v \cdot Q)$, respectively.

(2) Generate $TEK_1 = \hat{e}(T_1, CLSK_{ID,k,0})$ and $TEK_2 = \hat{e}(T_1, CLISK_{ID,k,0})$.
(3) Compute $EK'_1 = TEK_1 \cdot \hat{e}(T_1, CLSK_{ID,k,1})$ and $EK'_2 = TEK_2 \cdot \hat{e}(T_1, CLISK_{ID,k,1})$.
(4) Recover $M = SD_{EK'}(T_2)$, where $EK' = EK'_1 \oplus EK'_2$.
(5) Set $\beta' = SH_1(T_1, T_2, ID_{PKI}, ID_{CL}, M)$.
(6) Output M if $\hat{e}(Q, T_0) = PKIPK_{ID} \cdot \hat{e}(T_1, (W + \beta' \cdot T))$ is true.

The correctness of two equations $EK' = EK'_1 \oplus EK'_2 = EK_1 \oplus EK_2 = EK$ and $\hat{e}(Q, T_0) = PKIPK_{ID} \cdot \hat{e}(T_1, (W + \beta' \cdot T))$ are shown as follows.

$$
\begin{aligned}
\surd\ EK' &= EK'_1 \oplus EK'_2 \\
&= TEK_1 \cdot \hat{e}(T_1, CLSK_{ID,k,1}) \oplus TEK_2 \cdot \hat{e}(T_1, CLISK_{ID,k,1}) \\
&= \hat{e}(T_1, CLSK_{ID,k,0}) \cdot \hat{e}(T_1, CLSK_{ID,k,1}) \oplus \hat{e}(T_1, CLISK_{ID,k,0}) \\
&\quad \cdot \hat{e}(T_1, CLISK_{ID,k,1}) \\
&= \hat{e}(T_1, CLSK_{ID}) \oplus \hat{e}(T_1, CLISK_{ID}) \\
&= \hat{e}(n \cdot Q, CLSK_{ID}) \oplus \hat{e}(n \cdot Q, CLISK_{ID}) \\
&= \hat{e}(Q, CLSK_{ID})^n \oplus \hat{e}\big(n \cdot Q, SK_{KGC} + \big(f \cdot (W + \rho \cdot T)\big)\big) \\
&= \hat{e}(Q, CLSK_{ID})^n \oplus \hat{e}(n \cdot Q, SK_{KGC}) \cdot \hat{e}\big(n \cdot Q, \big(f \cdot (W + \rho \cdot T)\big)\big) \\
&= \hat{e}(Q, CLSK_{ID})^n \oplus \hat{e}(Q, SK_{KGC})^n \cdot \hat{e}\big(f \cdot Q, \big(n \cdot (W + \rho \cdot T)\big)\big) \\
&= (CLPK_{ID})^n \oplus \big(PK_{KGC} \cdot \hat{e}\big(CLIPK_{ID}, (W + \rho \cdot T)\big)\big)^n \\
&= EK_1 \oplus EK_2.
\end{aligned}
$$

$$
\begin{aligned}
\surd\ \hat{e}(Q, T_0) &= \hat{e}(Q, PKISK_{ID,j,1} + TS) \\
&= \hat{e}(Q, PKISK_{ID,j,1} + \big(PKISK_{ID,j,0} + \big(n \cdot (W + \beta \cdot T)\big)\big) \\
&= \hat{e}\big(Q, PKISK_{ID} + \big(n \cdot (W + \beta \cdot T)\big)\big) \\
&= \hat{e}(Q, PKISK_{ID}) \cdot \hat{e}\big(Q, \big(n \cdot (W + \beta \cdot T)\big)\big) \\
&= PKIPK_{ID} \cdot \hat{e}\big(n \cdot Q, (W + \beta \cdot T)\big) \\
&= PKIPK_{ID} \cdot \hat{e}\big(T_1, \big(W + \beta' \cdot T\big)\big).
\end{aligned}
$$

5. Security Analysis

In Definitions 2 and 3, we define two adversary games $Game_1$ and $Game_2$, respectively, to model the content unforgeability (authentication) and the message confidentiality in the LR-HSC-HPKS scheme. Under $Game_1$ and $Game_2$, Theorems 1 and 2 show that the LR-HSC-HPKS scheme is EUF-ACMSCA-secure and EIND-CCSCA-secure against both A_I and A_{II}, respectively.

Theorem 1. *Based on the SH assumption and the DL assumption in the GBG model, the LR-HSC-HPKS scheme is EUF-ACMSCA-secure against adversaries A (A_I and A_{II}).*

Proof. An adversary A and a challenger B cooperatively play $Game_1$ as follows.

– *Initialization phase*. B runs the *System setup* in Definition 1 to generate $SP = \{G, G_1, \hat{e}, Q, Q_1, q, W, T, SE/SD, SH_0, SH_1\}$, the CA's secret/public key pair (SK_{CA}, PK_{CA}) and the KGC's secret/public key pair (SK_{KGC}, PK_{KGC}). Additionally, if A is an A_{II}, both SK_{CA} and SK_{KGC} are sent to A_{II}. Also, six initially empty lists LT_a, LT_b, LT_{SK}, LT_{ISK}, LT_{HSE} and LT_{SH} are constructed as follows.
  - LT_a: Each element of G is recorded as a pair of (multi-variate polynomial, bit-string) in LT_a, represented as $(\Psi G_{x,y,z}, \Omega G_{x,y,z})$, where the three x, y and z, denote type-x query, y-th query and z-th item, respectively. Also, B records $(\Psi Q, \Omega G_{S,0,1})$, $(\Psi W, \Omega G_{S,0,2})$, $(\Psi T, \Omega G_{S,0,3})$, $(\Psi SK_{CA}, \Omega G_{S,0,4})$ and $(\Psi SK_{KGC}, \Omega G_{S,0,5})$ in LT_a. In the subsequent *Query phase*, there is an auto-transformation process that can transform $\Psi G_{x,y,z}$ (or $\Omega G_{x,y,z}$) to $\Omega G_{x,y,z}$ (or $\Psi G_{x,y,z}$).
  - LT_b: Each element of G_1 is recorded as a pair of (multi-variate polynomial, bit-string) in LT_b, represented as $(\Psi G_{1,x,y,z}, \Omega G_{1,x,y,z})$, where x, y and z are identical with those in LT_a. Additionally, B records $(\Psi PK_{CA}, \Omega G_{1,S,0,1})$ and $(\Psi PK_{KGC}, \Omega G_{1,S,0,1})$ in LT_b. Also, there is an auto-transformation process that can transform $\Psi G_{1,x,y,z}$ (or $\Omega G_{1,x,y,z}$) to $\Omega G_{1,x,y,z}$ (or $\Psi G_{1,x,y,z}$).
  - LT_{SK}: A secret/public key pair of ID_{PKI}/ID_{CL} is recorded as a tuple (ID_{PKI}/ID_{CL}, $\Psi PKISK_{ID}/\Psi CLSK_{ID}$, $\Psi PKIPK_{ID}/\Psi CLPK_{ID}$) in LT_{SK}.
  - LT_{ISK}: An identity secret/public key pair of ID_{CL} is recorded as a tuple (ID_{CL}, $\Psi CLISK_{ID}$, $\Psi CLIPK_{ID}$) in LT_{ISK}.
  - L_{HSE}: The related contents of requesting the *Hybrid signcryption query* (M, ID_{PKI}, ID_{CL}) are recorded as a tuple $(M, \Psi T_0, \Psi T_1, T_2, \Psi EK_1, \Psi EK_2, \Psi\beta, ID_{PKI}, ID_{CL})$ in L_{HSE}.
  - LT_{SH}: The related contents of requesting $SH_1()$ are recorded as a pair $(\Omega T_1||T_2||ID_{PKI}||ID_{CL}||M, \Omega\beta)$.
– *Query phase*: A (A_I or A_{II}) may adaptively request various kinds of queries (oracles) to B at most p times as follows.
  - O_a *query* $(\Omega G_{O,r,i}, \Omega G_{O,r,j}, OP)$: B first transforms $(\Omega G_{O,r,i}, \Omega G_{O,r,j})$ to $(\Psi G_{O,r,i}, \Psi G_{O,r,j})$. B computes $\Psi G_{O,r,k} = \Psi G_{O,r,i} + \Psi G_{O,r,j}$ if OP is "addition". Otherwise, B computes $\Psi G_{O,l,k} = \Psi G_{O,r,i} - \Psi G_{O,r,j}$. Also, B records $(\Psi G_{O,r,k}, \Omega G_{O,r,k})$ in LT_a.
  - O_m *query* $(\Omega G_{1,O,r,i}, \Omega G_{1,O,r,j}, OP)$: B first transforms $(\Omega G_{1,O,r,i}, \Omega G_{1,O,r,j})$ to $(\Psi G_{1,O,r,i}, \Psi G_{1,O,r,j})$. B computes $\Psi G_{1,O,r,k} = \Psi G_{1,O,r,i} + \Psi G_{1,O,r,j}$ if OP is "multiplication". Otherwise, B computes $\Psi G_{1,O,r,k} = \Psi G_{1,O,r,i} - \Psi G_{1,O,r,j}$. Also, B records $(\Psi G_{1,O,r,k}, \Omega G_{1,O,r,k})$ in LT_b.
  - $O_{\hat{e}}$ *query* $(\Omega G_{O,l,i}, \Omega G_{O,l,j})$: B first transforms $(\Omega G_{O,r,i}, \Omega G_{O,l,j})$ to $(\Psi G_{O,r,i}, \Psi G_{O,r,j})$. B computes $\Psi G_{1,O,r,k} = \Psi G_{O,r,i} \cdot \Psi G_{O,r,j}$ and records $(\Psi G_{1,O,r,k}, \Omega G_{1,O,r,k})$ in LT_b.
  - *Signer secret key query* (ID_{PKI}): B uses ID_{PKI} to find $(ID_{PKI}, \Psi PKISK_{ID}, \Psi PKIPK_{ID})$ in LT_{SK}. If found, B transforms $\Psi PKISK_{ID}$ to return $\Omega PKISK_{ID}$. Otherwise, B chooses ΨGR in G and computes $\Psi PKR = \Psi Q \cdot \Psi GR$. B records $(PKI_{ID}, \Psi PKISK_{ID} = \Psi GR, \Psi PKIPK_{ID} = \Psi PKR)$ in LT_{SK}. Also, B respectively records $(\Psi GR, \Omega GR)$ and $(\Psi PKR, \Omega PKR)$ in LT_a and LT_b, and returns ΩGR and ΩPKR.

- *Signer certificate query* (ID_{PKI}, $\Omega PKIPK_{ID}$): For the i-th request of this query, B first updates the old secret key $\Psi SK_{CA} = (\Psi SK_{CA,i-1,0}, \Psi SK_{CA,i-1,1})$ to the new secret key $\Psi SK_{CA} = (\Psi SK_{CA,i,0}, \Psi SK_{CA,i,1})$, and uses $(\Psi SK_{CA,i,0}, \Psi SK_{CA,i,1})$ to generate and return the signer ID_{PKI}'s certificate CRT_{ID}.
- *Signer certificate leak query* $(i, f_{SCG,i}, h_{SCG,i})$: For the i-th request of the *Signer certificate query*, the leak query can only be requested once. B returns $\Delta f_{SCG,i} = f_{SCG,i}(SK_{CA,i,0})$ and $\Delta h_{SCG,i} = h_{SCG,i}(SK_{CA,i,1})$.
- *Decrypter identity secret key query* (ID_{CL}). For the i-th request of this query, B first updates the old secret key $\Psi SK_{KGC} = (\Psi SK_{KGC,i-1,0}, \Psi SK_{KGC,i-1,1})$ to the new secret key $\Psi SK_{KGC} = (\Psi SK_{KGC,i,0}, \Psi SK_{KGC,i,1})$. B chooses ΨGT and $\Psi\rho$ in G, and generates the *decrypter* ID_{CL}'s identity secret/public key pair $(\Psi CLISK_{ID} = \Psi SK_{KGC} + \Psi GT \cdot (\Psi W + \Psi\rho \cdot \Psi T), \Psi CLIPK_{ID} = \Psi GT)$. B records $(\Psi CLISK_{ID}, \Omega CLISK_{ID})$, $(\Psi CLIPK_{ID}, \Omega CLIPK_{ID})$ and $(\Psi\rho, \Omega\rho = ID_{CL}||\Omega CLIPK_{ID})$ in LT_a. Also, B records $(ID_{CL}, \Psi CLISK_{ID}, \Psi CLIPK_{ID})$ in LT_{ISK}, and returns both $\Omega CLISK_{ID}$ and $\Omega CLIPK_{ID}$.
- *Decrypter identity secret key leak query* $(i, f_{ISKG,i}, h_{ISKG,i})$. For the i-th request of the *Decrypter identity secret key query*, the leak query can only be requested once. B returns $\Delta f_{ISKG,i} = f_{ISKG,i}(SK_{KGC,i,0})$ and $\Delta h_{ISKG,i} = h_{ISKG,i}(SK_{KGC,i,1})$.
- *Decrypter public key replace query* $(ID_{CL}, (\Omega CLPK'_{ID}, \Omega CLIPK'_{ID}))$. B transforms $(\Omega CLPK'_{ID}, \Omega CLIPK'_{ID})$ to $(\Psi CLPK'_{ID}, \Psi CLIPK'_{ID})$. B modifies $(CL_{ID}, -, \Psi CLPK'_{ID})$ in LT_{SK} and $(CL_{ID}, -, \Psi CLIPK'_{ID})$ in LT_{ISK}.
- *Decrypter secret key query* (ID_{CL}). B uses ID_{CL} to find $(ID_{CL}, \Psi CLSK_{ID}, \Psi CLPK_{ID})$ in LT_{SK}. If found, B transforms $\Psi CLSK_{ID}$ to return $\Omega CLSK_{ID}$. Otherwise, B chooses ΨGR in G and computes $\Psi PKR = \Psi Q \cdot \Psi GR$. B records $(ID_{CL}, \Psi CLSK_{ID} = \Psi GR, \Psi CLPK_{ID} = \Psi PKR)$ in LT_{SK}. Also, B respectively records $(\Psi GR, \Omega GR)$ and $(\Psi PKR, \Omega PKR)$ in LT_a and LT_b, and returns both ΩGR and ΩPKR.
- *Hybrid signcryption query* (M, ID_{PKI}, ID_{CL}): B first updates the signer ID_{PKI}'s old secret key $\Psi PKISK_{ID} = (\Psi PKISK_{ID,j-1,0}, \Psi PKISK_{ID,j-1,1})$ to the new secret key $\Psi PKISK_{ID} = (\Psi PKISK_{ID,j,0}, \Psi PKISK_{ID,j,1})$. B performs the following detailed processes to return CT.

  (1) By ID_{CL}, find $(ID_{CL}, \Psi CLIPK_{ID}, \Psi CLISK_{ID})$ in LT_{ISK} and $(ID_{CL}, \Psi CLPK_{ID}, \Psi CLSK_{ID})$ in LT_{SK}. Meanwhile, transform $\Psi CLIPK_{ID}$ to $\Omega CLIPK_{ID}$.
  (2) Select $\Psi\rho$ and Ψn in G and record $(\Psi\rho, ID_{CL}||\Omega CLIPK_{ID})$ in LT_a.
  (3) Compute $\Psi EK_1 = \Psi CLPK_{ID} \cdot \Psi n$ and $\Psi EK_2 = (\Psi PK_{KGC} + (\Psi CLIPK_{ID} \cdot (\Psi W + \Psi\rho \cdot \Psi T))) \cdot \Psi n$.
  (4) Transform Ψn, ΨEK_1 and ΨEK_2 to Ωn, ΩEK_1 and ΩEK_2, respectively.
  (5) Compute $\Omega EK = \Omega EK_1 \oplus \Omega EK_2$ and $T_2 = SE_{\Omega EK}(M)$.
  (6) Compute $\Omega\beta = SH_1(\Omega n, T_2, ID_{PKI}, ID_{CL}, M)$, select $\Omega\beta$ in G, and record $(\Phi\beta, \Omega\beta)$ in LT_a.
  (7) Compute $\Psi T_0 = \Psi PKISK_{ID} + (\Psi n \cdot (\Psi W + \Psi T \cdot \Psi\beta))$ and transform ΨT_0 to ΩT_0.
  (8) Record $(M, \Psi T_0, \Psi n, T_2, \Psi EK_1, \Psi EK_2, \Psi\beta, ID_{PKI}, ID_{CL})$ in L_{HSE}.
  (9) Return $CT = (\Omega T_0, \Omega n, T_2, ID_{PKI}, ID_{CL})$.

- *Hybrid signcryption leak query* $(ID_{PKI}, j, f_{HS,j}, h_{HS,j})$: For the signer ID_{PKI}'s j-th request of the *Decrypter identity secret key query*, the leak query can only be requested once. B returns $\Delta f_{HS,j} = f_{HS,j}(PKISK_{ID,j,0})$ and $\Delta h_{HS,j} = h_{HS,j}(PKISK_{ID,j,1})$.
- *Hybrid unsigncryption query* (CT, ID_{PKI}, ID_{CL}): B first updates the decrypter ID_{CL}'s old secret key $(CLSK_{ID,k-1,0}, CLSK_{ID,k-1,1})$ and identity secret key $(CLISK_{ID,k-1,0}$, $CLISK_{ID,k-1,1})$ to $\Psi CLSK_{ID} = (\Psi CLSK_{ID,k,0}, \Psi CLSK_{ID,k,1})$ and $\Psi CLISK_{ID} = (\Psi CLISK_{ID,k,0}, \Psi CLISK_{ID,k,1})$, respectively. B performs the following detailed processes to return M.

  (1) By ID_{PKI}, find $(ID_{PKI}, \Psi PKIPK_{ID})$ in LT_{SK} and transform $\Psi PKIPK_{ID}$ to $\Omega PKIPK_{ID}$.
  (2) Transform ΩT_0 and Ωn to ΨT_0 and Ψn, respectively.
  (3) Compute $\Psi EK_1 = \Psi n \cdot \Psi CLSK_{ID}$ and $\Psi EK_2 = \Psi n \cdot \Psi CLISK_{ID}$.
  (4) Set $\Omega\beta = SH_1(\Omega n, T_2, ID_{PKI}, ID_{CL}, M)$ and transform $\Omega\beta$ to $\Psi\beta$.
  (5) Use $(\Psi T_0, \Psi n, T_2, \Psi n, \Psi EK_1, \Psi n, \Psi EK_2, \Psi\beta, ID_{PKI}, ID_{CL})$ to find $(M, \Psi T_0, \Psi T_1, T_2, \Psi EK_1, \Psi EK_2, \Psi\beta, ID_{PKI}, ID_{CL})$ in L_{HSE}.
  (6) If found, return M. Otherwise, return "invalid".

- *Hybrid unsigncryption leak query* $(ID_{CL}, k, f_{HUS,k}, h_{HUS,k})$: For the decrypter ID_{CL}'s k-th request of the *Decrypter identity secret key query*, the leak query can only be requested once. B returns $\Delta f_{HUS,k} = f_{HUS,k}(CLSK_{ID,k,0}, CLSK_{ID,k,1})$ and $\Delta h_{HUS,k} = h_{HUS,k}(CLISK_{ID,k,0}, CLISK_{ID,k,1})$.

– *Forgery phase*: Assume that A forges a ciphertext $CT^* = (T^*_0, T^*_1, T^*_2, ID^*_{PKI}, ID^*_{CL})$ for the message M^*, we say that A wins $Game_1$ when three provisions mentioned in the *Forgery phase* of Definition 2 (i.e. $Game_1$) are true.

In the following, let us first evaluate the advantage of A_I without requesting any leak queries in $Game_1$, denoted as $Adv_1(A_{I-wo})$. By $Adv_1(A_{I-wo})$, we then evaluate the advantage of A_I with requesting all leak queries in $Game_1$, denoted as $Adv_1(A_I)$. By similar analysis, $Adv_1(A_{II})$ is also gained.

■ **The evaluation of $\mathbf{Adv_1(A_{I-wo})}$**: In the GBG model, if adversaries can find collisions in G and G_1, the *discrete logarithm problem* in G and G_1 will be resolved. The total number of elements in both LT_a and LT_b is first counted. In the *Query phase*, A_I may request various kinds of queries (oracles) to B at most p times while the number of the added elements in a query (i.e. the *Hybrid signcryption query*) is at most 6. Therefore, we have $|LT_a| + |LT_b| \leqq 6p$. Also, the maximal degrees of polynomials in LT_a and LT_b are 3 and 6, respectively. Moreover, $Adv_1(A_{I-wo})$ includes two cases' probabilities as evaluated below.

(1) Pb[*Forgery*]: Let Pb[*Forgery*] denote the probability that A_I forges a ciphertext $CT^* = (T^*_0, T^*_1, T^*_2, ID^*_{PKI}, ID^*_{CL})$ for a message M^* that satisfies $\hat{e}(Q, T^*_0) = PKIPK_{ID^*} \cdot \hat{e}(T^*_1, (W + \beta' \cdot T))$ in the *Hybrid unsigncryption*. That is, we have $\Psi Q \cdot \Psi T^*_0 = \Psi PKIPK_{ID*} + \Psi T^*_1 \cdot (\Psi W + \Psi\beta' \cdot \Psi T)$ and set

$\Psi f = \Psi Q \cdot \Psi T^*{}_0 - (\Psi PKIPK_{ID*} + \Psi T^*{}_1 \cdot (\Psi W + \Psi\beta' \cdot \Psi T))$ that has degree 3. By Lemma 2, we have Pb[*Forgery*] $= 3/q$ because the probability of $\Psi f = 0$ is $3/q$.

(2) Pb[*Collision*]: Let Pb[*Collision*] denote the probability that A_I may find collisions in LT_a or LT_b. Assume that the polynomials in LT_a have s variates, represented by using s random integers $u_i \in Z_q{}^*$, for $i = 1, 2, \ldots, s$. Let $(\Psi G_j, \Psi G_k)$ denote a pair of two different polynomials in LT_a so that there are $\binom{|LT_a|}{2}$ pairs of $(\Psi G_j, \Psi G_k)$. For each pair, we set $\Psi G_l(u_1, u_2, \ldots, u_s) = \Psi G_j - \Psi G_k$. If there exists any $\Psi G_l = 0$, a collision in LT_a has occurred. Since there are $\binom{|LT_a|}{2}$ pairs of $(\Psi G_j, \Psi G_k)$ and the maximal degree of polynomials in LT_a is 3, we have that Pb[*Collision*] in LT_a is $(3/q)\binom{|LT_a|}{2}$. By similar arguments, we have that Pb[*Collision*] in LT_b is $(6/q)\binom{|LT_b|}{2}$. Since $|LT_a| + |LT_b| \leqq 6p$, we have

$$\begin{aligned}\text{Pb}[Collision] &\leqq (3/q)\binom{|LT_a|}{2} + (6/q)\binom{|LT_b|}{2} \\ &\leqq (6/q)\big(|LT_a| + |LT_b|\big)^2 \\ &\leqq 216p^2/q = O\big(p^2/q\big).\end{aligned}$$

Due to the above discussions, we have

$$\begin{aligned}Adv_1(A_{I-wo}) &= \text{Pb}[Forgey] + \text{Pb}[Collision] \\ &\leqq 3/q + O\big(p^2/q\big) \\ &= O\big(p^2/q\big).\end{aligned}$$

■ **The evaluation of $\mathbf{Adv_1(A_I)}$**: By $Adv_1(A_{I-wo})$, we evaluate the advantage $Adv_1(A_I)$ of A_I with requesting all leak queries in $Game_1$. These leak queries include *Signer certificate leak query*, *Decrypter identity secret key leak query*, *Hybrid signcryption leak query* and *Hybrid unsigncryption leak query*. Due to the key updating process, any two leaked portions of a secret key are mutually independent. Therefore, A_I could gain at most 2τ bits of SK_{CA}, 2τ bits of SK_{KGC}, 2τ bits of $PKISK_{ID}$, and 2τ bits of both $CLSK_{ID}$ and $CLISK_{ID}$. Hence, we have

$$Adv_1(A_I) \leqq Adv_1(A_{I-wo}) \cdot 2^{2\tau} = O\big(\big(p^2/q\big) \cdot 2^{2\tau}\big).$$

It is obvious that $Adv_1(A_I) = O((p^2/q) \cdot 2^{2\tau})$ is negligible if $p = poly(\log q)$ by Lemma 2.

■ **The evaluation of $\mathbf{Adv_1(A_{II})}$**: A_{II} is used to model the attacking ability of a malicious CA/KGC who has both SK_{CA} and SK_{KGC}. Therefore, A_{II} could gain at most 2τ bits of $PKISK_{ID}$, and 2τ bits of $CLSK_{ID}$ or $CLISK_{ID}$. By similar analysis of $Adv_1(A_I)$, we also have $Adv_1(A_{II}) = O((p^2/q) \cdot 2^{2\tau})$, that is negligible if $p = poly(\log q)$ by Lemma 2. □

Theorem 2. *Based on the SH assumption and the DL assumption in the GBG model, the LR-HSC-HPKS scheme is EIND-CCSCA-secure against adversaries A (A_I and A_{II}).*

Proof. An adversary A and a challenger B cooperatively play $Game_2$ as follows.

- *Initialization phase*: It is exactly the same as the *Initialization phase* in the proof of Theorem 1.
- *Query phase*: It is exactly like the *Query phase* of Theorem 1.
- *Challenge phase*: A selects a target decrypter ID^*_{CL} and a message pair (M_0, M_1) as a challenge objective. B randomly selects $c \in \{0, 1\}$ and generates a challenge ciphertext CT^* by running the Hybrid signcryption with $(M_c, ID_{PKI}, ID^*_{CL})$. Also, B sends CT^* to A. Note that two provisions mentioned in the *Challenge phase* of Definition 3 (i.e. $Game_2$) must be satisfied.
- *Guessing phase*: A outputs $c' \in \{0, 1\}$ and wins $Game_2$ if $c' = c$. Meanwhile, A's advantage is defined as $Adv(A) = |\text{Pb}[c' = c] - 1/2|$.

By similar evaluations as in the proof of Theorem 1, we can evaluate the advantages of A_I without requesting any leak queries in $Game_2$, denoted as $Adv_2(A_{I-wo})$. By $Adv_2(A_{I-wo})$, we then evaluate the advantage of A_I with requesting all leak queries in $Game_2$, denoted as $Adv_2(A_I)$. By similar analysis, $Adv_2(A_{II})$ is also gained.

■ **The evaluation of $\mathbf{Adv_2(A_{I-wo})}$**: $Adv_2(A_{I-wo})$ includes two cases' probabilities as evaluated below.

(1) Pb[*Guessing*]: Since A_{I-wo} is not permitted to request any leak query, there is no useful information about secret keys. Therefore, the probability of guessing $c' = c$ is $1/2$, namely, Pb[*Guessing*] $= 1/2$.
(2) Pb[*Collision*]: The probability is identical to the probability Pb[*Collision*] in the proof of Theorem 1, namely, Pb[*Collision*] $= O(p^2/q)$.

Due to the above discussions, we have

$$\begin{aligned} Adv_2(A_{I-wo}) &= |\text{Pb}[c' = c] - 1/2| \\ &= |\text{Pb}[Guessing] - 1/2| + |\text{Pb}[Collision]| \\ &= O(p^2/q). \end{aligned}$$

■ **The evaluation of $\mathbf{Adv_2(A_I)}$**: By $Adv_2(A_{I-wo})$, we evaluate the advantage $Adv_2(A_I)$ of A_I with requesting all leak queries in $Game_2$. By the same evaluation as $Adv_1(A_I)$ in the proof of Theorem 1, A_I could gain at most 2τ bits of SK_{CA}, 2τ bits of SK_{KGC}, 2τ bits of $PKISK_{ID}$, and 2τ bits of both $CLSK_{ID}$ and $CLISK_{ID}$. Hence, we also have

$$Adv_2(A_I) \leqq Adv_2(A_{I-wo}) \cdot 2^{2\tau} = O\left((p^2/q) \cdot 2^{2\tau}\right).$$

It is obvious that $Adv_2(A_I) = O((p^2/q) \cdot 2^{2\tau})$ is negligible if $p = poly(\log q)$ by Lemma 2.

Table 3
Required costs (ms) of three time-consuming computations.

Devices	T_{bil}	T_{mul}	T_{exp}
PDA	≈96 ms	≈30 ms	≈30 ms
PC	≈20 ms	≈6 ms	≈6 ms

Table 4
Computational complexities and costs (ms) of our LR-HSC-HPKS scheme.

Algorithms	Computational complexities	Costs on a PDA	Costs on a PC
System setup	$T_{bil} + 2T_{mul}$	156 ms	32 ms
User key generation for the PKI-PKS	$T_{bil} + 3T_{mul}$	186 ms	38 ms
User key generation for the CL-PKS	$T_{bil} + 7T_{mul}$	306 ms	62 ms
Hybrid signcryption	$T_{bil} + 5T_{mul} + 2T_{exp}$	306 ms	62 ms
Hybrid unsigncryption	$6T_{bil} + 2T_{mul}$	636 ms	132 ms

■ **The evaluation of $\mathbf{Adv_2(A_{II})}$**: A_{II} is used to model the attacking abilities of a malicious CA/KGC who has both SK_{CA} and SK_{KGC}. Therefore, A_{II} could gain at most 2τ bits of $PKISK_{ID}$, and 2τ bits of $CLSK_{ID}$ or $CLISK_{ID}$. By similar analysis of $Adv_2(A_I)$, we also have $Adv_2(A_{II}) = O((p^2/q) \cdot 2^{2\tau})$, that is negligible if $p = poly(\log q)$ by Lemma 2. □

6. Performance Analysis

In the following, the notations of three time-consuming computations are defined.

- T_{bil}: The computational complexity of running a bilinear pairing $\hat{e} : G \times G \rightarrow G_1$.
- T_{mul}: The computational complexity of running a multiplication in G.
- T_{exp}: The computational complexity of running an exponentiation in G_1.

By the performance experiences conducted in Xiong and Qin (2015), Table 3 lists the required costs (ms) of three time-consuming computations on a mobile device (PDA) and a PC. The security parameter of a bilinear group set $\{G, G_1, \hat{e}, Q, Q_1, q\}$ is set to a 512-bit prime order q. Also, the PDA and the PC are equipped with 624 MHz and 3 GHz CPUs, respectively. Table 4 lists the computational complexities and the required running costs (ms) of our LR-HSC-HPKS scheme in terms of *System setup*, *User key generation*, *Hybrid signcryption* (*HSE*) and *Hybrid unsigncryption* (*HUSE*) algorithms. For achieving leakage resilient property, the key updating process for each secret key must be employed, so that our scheme adds some extra computations. Nevertheless, by Table 4, the proposed scheme is well suitable for running on both a PDA and a PC. The point is that our scheme is the first hybrid signcryption scheme with leakage resilience.

7. Conclusions and Future Work

In recent years, many scholars have been studying several hybrid signcryption schemes in heterogeneous environments, but these schemes cannot withstand side-channel attacks, namely, these schemes do not possess the leakage-resilience property. Fortunately, the *first* leakage-resilient hybrid signcryption in heterogeneous public-key systems (LR-HSC-HPKS) has been proposed in this paper. Also, a new framework and two new adversary games of the LR-HSC-HPKS scheme were defined. Based on the SH assumption and the DL assumption in the GBG model, the proposed LR-HSC-HPKS scheme is EUF-ACMSCA-secure and EIND-CCSCA-secure against adversaries A (A_I and A_{II}), namely, illegitimate member (A_I) and malicious CA/KGC (A_{II}). Furthermore, by comparing with the previously proposed hybrid signcryption schemes, the proposed scheme has the following merits: (1) It is the first hybrid signcryption scheme resisting to side-channel attacks. (2) It possesses the unbounded leakage-resilient property, namely, allowing adversaries to repeatedly learn a portion of the secret key used in each computation. (3) All secret keys of the proposed scheme are allowed to be leaked to adversaries while maintaining the security of the proposed scheme. Finally, by the computational simulation results, performance analysis is demonstrated to show that the proposed scheme is well suitable for running on both a PDA and a PC. In the future, it is an interesting topic to propose a leakage-resilient hybrid signcryption scheme with equality test functionality in heterogeneous public-key systems.

Funding

This research was partially supported by National Science and Technology Council, Taiwan, under contract No. NSTC112-2221-E-018-011.

References

Akavia, A., Goldwasser, S., Vaikuntanathan, V. (2009). Simultaneous hardcore bits and cryptography against memory attacks. In: *Theory of Cryptography, TCC'09*, LNCS, Vol. 5444, pp. 474–495.

Ali, I., Lawrence, T., Omala, A.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: *Advances in Cryptology – ASIACRYPT 2003*, LNCS, 2894, pp. 452–473.

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

Baek, J., Steinfeld, R., Zheng, Y. (2007). Formal proofs for the security of signcryption. *Journal of Cryptology*, 20(2), 203–235.

Barbosa, M., Farshim, P. (2008). Certificateless signcryption. In: *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS'08*, pp. 369–372.

Biham, E., Carmeli, Y., Shamir, A. (2008). Bug attacks. In: *Advances in Cryptology – CRYPTO 2008*, LNCS, Vol. 5157, pp. 221–240.

Boneh, D., Franklin, M. (2001). Identity-based encryption from the Weil pairing. In: *Advances in Cryptology – CRYPTO 2001*, LNCS, 2139, pp. 213–229.

Boneh, D., Boyen, X., Goh, E. (2005). Hierarchical identity-based encryption with constant size ciphertext. In: *Advances in Cryptology–EURO–CRYPT 2005, 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.

Elkhalil, A., Zhang, J., Elhabob, R., Eltayieb, N. (2021). An efficient signcryption of heterogeneous systems for internet of vehicles. *Journal of Systems Architecture*, 113, 101885.

Galindo, D., Virek, S. (2013). A practical leakage-resilient signature scheme in the generic group model. In: *Selected Areas in Cryptography, 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.

Hou, Y., Huang, X., Chen, Y., Kumari, S., Xiong, H. (2021). Heterogeneous signcryption scheme supporting equality test from PKI to CLC toward IoT. *Transactions on Emerging Telecommunications Technologies*, 32(8), e4190.

Huang, Q., Wong, D.-S., Yang, G. (2011). Heterogeneous signcryption with key privacy. *Computer Journal*, 54(4), 525–536.

Karati, A., Islam, S.H., Biswas, G.P., Bhuiyan, M.Z., Vijayakumar, P., Karuppiah, M. (2018). Provably secure identity-based signcryption scheme for crowdsourced industrial Internet of Things environments. *IEEE Internet of Things Journal*, 5(4), 2904–2914.

Kiltz, E., Pietrzak, K. (2010). Leakage resilient Elgamal encryption. In: *Advances in Cryptology – ASIACRYPT 2010*, LNCS, Vol. 6477, pp. 595–612.

Li, C., Yang, G., Wong, D., Deng, X., Chow, S.S.M. (2010). An efficient signcryption scheme with key privacy and its extension to ring signcryption. *Journal of Computing and Security*, 18(3), 451–473.

Li, F., Xiong, P. (2013). Practical secure communication for integrating wireless sensor networks into the Internet of Things. *IEEE Sensors Journal*, 13(10), 3677–3684.

Li, F., Shirase, M., Takagi, T. (2013a). Certificateless hybrid signcryption. *Mathematical and Computer Modelling*, 57, 324–343.

Li, F., Zhang, H., Takagi, T. (2013b). Efficient signcryption for heterogeneous systems. *IEEE Systems Journal*, 7(3), 420–429.

Li, F., Han, Y., Jin, C. (2016a). Practical access control for sensor networks in the context of the internet of things. *Computer Communications*, 89–90, 154–164.

Li, F., Han, Y., Jin, C. (2016b). Practical signcryption for secure communication of wireless sensor networks. *Wireless Personal Communications*, 89, 1391–1412.

Liu, J., Zhang, L., Sun, R., Du, X., Guizani, M. (2018). Mutual heterogeneous signcryption schemes for 5G network slicings. *IEEE Access*, 6, 7854–7863.

Niu, S., Shao, H., Su, Y., Wang, C. (2023). Efficient heterogeneous signcryption scheme based on edge computing for industrial internet of things. *Journal of Systems Architecture*, 136, 102836.

Pan, X., Jin, Y., Wang, Z., Li, F. (2022). A pairing-free heterogeneous signcryption scheme for unmanned aerial vehicles. *IEEE Internet of Things Journal*, 9(19), 19426–19437.

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 ACM*, 21(2), 120–126.

Sun, Y., Li, H. (2010). Efficient signcryption between TPKC and IDPKC and its multi-receiver construction. *Science China Information Sciences*, 53, 557–566.

Tsai, T.-T., Tseng, Y.-M., Huang, S.-S. (2023). Leakage-resilient certificateless signcryption scheme under a continual leakage model. *IEEE Access*, 11, 54448–54461.

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., Huang, S.-S., Tsai, T.-T. (2022a). Practical leakage-resilient signcryption scheme suitable for mobile environments. In: *2022 IEEE 11th Global Conference on Consumer Electronics (GCCE)*, Osaka, Japan, 2022, pp. 383–384. https://doi.org/10.1109/GCCE56475.2022.10014332.

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

Tseng, Y.-M., Tsai, T.-T., Huang, S.-S. (2023). Fully continuous leakage-resilient certificate-based signcryption scheme for mobile communications. *Informatica*, 34(1), 199–222.

Wei, G., Shao, J., Xiang, Y., Zhu, P., Lu, R. (2015). Obtain confidentiality or/and authenticity in big data by ID-based generalized signcryption. *Information Sciences*, 318, 111–122.

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. (2019). An identity-based authenticated key exchange protocol resilient to continuous key leakage. *IEEE Systems Journal*, 13(4), 3968–3979.

Xie, J.-Y., Tseng, Y.-M., Huang, S.-S. (2023). Leakage-resilient anonymous multi-receiver certificateless encryption resistant to side-channel attacks. *IEEE Systems Journal*, 17(2), 2674–2685.

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.

Xiong, H., Zhao, Y., Hou, Y., Huang, X., Jin, C., Wang, L., Kumari, S. (2021). Heterogeneous signcryption with equality test for IIoT environment. *IEEE Internet of Things Journal*, 8(21), 16142–16152.

Xiong, H., Hou, Y., Huang, X., Zhao, Y., Chen, C.-M. (2022). Heterogeneous signcryption scheme from IBC to PKI with equality test for WBANs. *IEEE Systems Journal*, 16(2), 2391–2400.

Zheng, Y. (1997). Digital signcryption or how to achieve cost (signature & encryption) cost (signature) + cost (encryption). In: *Advances in Cryptology – CRYPTO '97*, LNCS, Vol. 1294, pp. 165–179.

T.-C. Ho is currently working toward her PhD degree in the Department of Mathematics, National Changhua University of Education, Changhua, Taiwan. Her research interests include applied cryptography and leakage-resilience cryptography.

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 is an editor of several international journals.

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 obtained his PhD from the University of Illinois at Urbana-Champaign in 1997 under the supervision of Professor Bruce C. Berndt.