INFORMATICA, 2025, Vol. 36, No. 3, 737–764

DOI: https://doi.org/10.15388/25-INFOR600

LRSC-AMRS: Leakage-Resilient and Seamlessly Compatible Anonymous Multi-Recipient Signcryption in Heterogeneous Public-Key Cryptographies

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

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

Received: April 2025; accepted: September 2025

Abstract. Anonymous multi-recipient signcryption (AMRS) is an important scheme of public-key cryptography (PKC) and applied for many modern digital applications. In an AMRS scheme, a broadcast management centre (BMC) may sign and encrypt a plaintext data (or file) to a set of multiple recipients. Meanwhile, only these recipients in the set can decrypt the plaintext data and authenticate the BMC while offering anonymity of their identities. In the past, some AMRS schemes based on various PKCs have been proposed. Recently, due to side-channel attacks, the existing cryptographic mechanisms could be broken so that leakage-resilient PKC resisting such attacks has attracted the attention of cryptographic researches. However, the work on the design of leakage-resilient AMRS (LR-AMRS) schemes is little and only suitable for multiple recipients under a single PKC. In this paper, the *first* leakage-resilient and seamlessly compatible AMRS (LRSC-AMRS) in heterogeneous PKCs is proposed. In the proposed scheme, multiple recipients can be members of two heterogeneous PKCs, namely, the public-key infrastructure PKC (PKI-PKC) or the certificateless PKC (CL-PKC). Also, we present a seamlessly compatible upgradation procedure from the PKI-PKC to the CL-PKC. The proposed scheme achieves three security properties under side-channel attacks, namely, encryption confidentiality, recipient anonymity and sender (i.e. BMC) authentication, which are formally shown by the associated security theorems. Finally, by comparing with related schemes, it is shown that the proposed LRSC-AMRS scheme is suitable for heterogeneous recipients and the computational cost of each recipient's unsigncryption algorithm is constant $O(1)$.

Key words: leakage resilience, multiple recipients, anonymity, encryption, authentication, heterogeneous public-key cryptographies.

1. Introduction

With the popularity of the Internet and wireless networks, a large number of cryptographic mechanisms have been proposed to ensure information/communication security for various applications based on public-key cryptography (PKC). Indeed, the most popular PKC

*Corresponding author.

today is still the public-key-infrastructure (PKI) PKC (PKI-PKC) (Rivest *et al.*, 1978; El-Gamal, 1985; Miller, 1985). In the PKI-PKC, each member has a secret/public key pair and there is a certificate authority (CA) who is responsible to generate and manage the associated certificates of all members' public keys. However, the PKI-PKC requires a complex PKI architecture to maintain the validity of these certificates. In 2001, Boneh and Franklin (2001) realized Shamir's identity (ID)-based concept (Shamir, 1984) to propose the ID-PKC. However, in the ID-PKC, there is a secret key generator (SKG) who is responsible to generate all members' secret keys, so that the SKG knows these secret keys which incurs a key escrow problem. Now, the usage of the certificateless PKC (CL-PKC) has attracted the attention of researchers because it has neither certificate management nor key escrow problems.

For the mentioned PKCs (i.e. the PKI-PKC, the ID-PKC and the CL-PKC) above, all secret keys must be completely protected from leakage of any partial information. However, by side-channel attacks (Brumley and Boneh, 2005; Biham *et al.*, 2008), an adversary could acquire partial information of secret keys participated in the computations of cryptographic mechanisms. Eventually, via continuous leakage, the existing cryptographic mechanisms based on these PKCs mentioned above could be broken. The designs of leakage-resilient (LR) cryptographic mechanisms resisting side-channel attacks have attracted the attention of cryptographic researches who have proposed many LR cryptographic mechanisms (Kiltz and Pietrzak, 2010; Galindo and Vivek, 2013; Wu *et al.*, 2018, 2019; Peng *et al.*, 2021; Tseng *et al.*, 2022; Xie *et al.*, 2023).

Anonymous multi-recipient signcryption (AMRS) is an important scheme of PKCs and applied for many modern digital applications, e.g. over-the-air (OTA) applications (Li *et al.*, 2023), unmanned chain stores (Park and Zhang, 2022) and digital signages (Kim *et al.*, 2024). In an AMRS scheme, there are a trusted authority, a broadcast management centre (BMC) and many recipients. The BMC and recipients first obtain their secret/public key pairs by interacting with the trusted authority. Also, the BMC may sign and encrypt a plaintext data (or file) to generate a broadcast ciphertext set (*BCS*), and convey the *BCS* to a set of multiple recipients via Internet. Meanwhile, only these recipients in the set can decrypt the plaintext data and authenticate the BMC while offering anonymity for their identities. The system architecture of an AMRS scheme in a PKC is depicted in Fig. 1.

1.1. *Motivation*

To our best knowledge, all the existing AMRS schemes are only suitable for multiple recipients under a single PKC and will be reviewed later. Here, let us consider the situation of the PKC upgradation as follows. When the original PKC (e.g. the PKI-PKC) is converted and upgraded to another new PKC (e.g. the CL-PKC), some recipients in the PKI-PKC are possibly not upgraded to the CL-PKC successfully. This situation, in the AMRS scheme, would result in three kinds of recipients, namely, initial (i.e. non-upgraded) recipients in the PKI-PKC, upgraded and new recipients in the CL-PKC, as illustrated in Fig. 2. Indeed, for the PKC upgradation, it is important to ensure that non-upgraded recipients can still use the original cryptographic functionalities (Ho *et al.*, 2024; Tseng *et al.*, 2024). However,

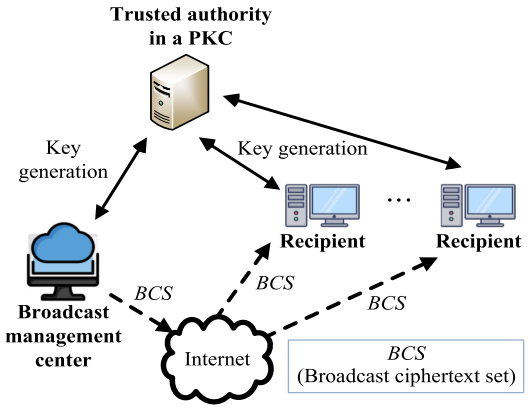

Fig. 1. The system architecture of an AMRS scheme in a PKC.

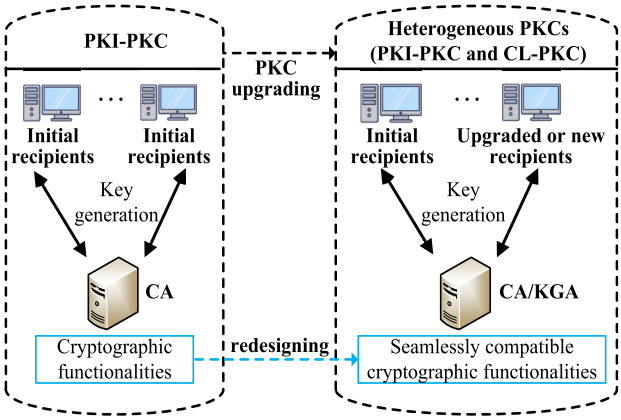

Fig. 2. An illustration of the PKC upgradation from the PKI-PKC to the CL-PKC.

the existing AMRS schemes are unsuitable for such heterogeneous PKCs. Additionally, the work on the design of leakage-resilient AMRS (LR-AMRS) schemes is little and only suitable for multiple recipients under a single PKC. Our aim of this paper is to propose the *first* leakage-resilient and seamlessly compatible AMRS (LRSC-AMRS) suitable for multiple recipients under two heterogeneous PKCs (i.e. the PKI-PKC and the CL-PKC).

1.2. *Related Work*

Here, the revolutions of AMRS and leakage-resilient AMRS (LR-AMRS) schemes based on different PKCs (i.e. the PKI-PKC, the ID-PKC and the CL-PKC) are reviewed.

Based on the PKI-PKC, Wang *et al.* (2016) employed the ring signcryption technique to propose the first PKI-AMRS scheme. Also, Tsai *et al.* (2022) used the concepts of both the LR encryption scheme in Kiltz and Pietrzak (2010) and the LR signature scheme in

Galindo and Vivek (2013) to integrate multi-recipient scenario to propose the first PKI-LR-AMRS scheme based on the PKI-PKC. The PKI-LR-AMRS scheme not only possesses the functionalities and security properties of an AMRS scheme, but also permits an adversary to continuously acquire partial information of secret keys participated in the computations. Nevertheless, these AMRS schemes based on the PKI-PKC require a complex PKI architecture to maintain the validity of recipients' certificates.

Based on the ID-PKC, Lal and Kushwah (2009) proposed the first ID-AMRS scheme, which employed the Lagrange interpolation polynomial technique to embed multiple recipients' identities to achieve anonymity between these recipients. To achieve robust security, Zhang and Xu (2010) proposed an improved ID-AMRS scheme in the standard model. However, Wang *et al.* (2012) demonstrated that these two schemes (Lal and Kushwah, 2009; Zhang and Xu, 2010) cannot achieve anonymity because an authorized recipient can check whether other recipients are authorized. Furthermore, Pang *et al.* (2015) proposed a new ID-AMRS scheme based on the ID-PKC, which achieves the anonymity of both the sender and recipients. Nevertheless, these AMRS schemes based on the ID-PKC still inherit the key escrow problem.

Based on the CL-PKC, Pang *et al.* (2018) proposed the first CL-AMRS scheme. Also, Pang *et al.* (2019) proposed a new CL-AMRS scheme, which achieves the anonymity of both the sender and recipients. Moreover, to achieve a sender's anonymity and traceability, Li *et al.* (2022) presented a new CL-AMRS scheme. For wireless body area networks, Shen *et al.* (2022) proposed a lightweight CL-AMRS scheme. Unfortunately, Dong and Zhang (2024) pointed out that Shen *et al.*'s scheme suffers from forgery attacks.

1.3. *Techniques and Contributions*

By the related work mentioned above, all the existing AMRS schemes are only suitable for multiple recipients under a single PKC (i.e. the PKI-PKC, the ID-PKC or the CL-PKC) and are unsuited for multiple recipients under heterogeneous PKCs. Also, among them, only Tsai *et al.*'s PKI-LR-AMRS scheme (2022) can withstand side-channel attacks and possess unbounded leakage resilience. We will adopt the same multiplicative blinding technique (Kiltz and Pietrzak, 2010; Galindo and Vivek, 2013; Tsai *et al.*, 2022) to refresh secret keys in our proposed scheme. In this multiplicative blinding technique, each secret key is initially separated into two fragments. Before a secret key is participated in computations, the associated two fragments must be refreshed while remaining the corresponding public key unchanged. For achieving unbounded leakage resilience, a leakage-resilient cryptographic scheme must possess two pre-conditions, namely, bounded leakage of single computation and computation leakage. Indeed, due to the multiplicative blinding technique, any two leaked partial information of a secret key are mutually independent so that it achieves independent leakage property. Also, by the independent leakage property, the total leakage information of secret keys is unbounded.

By extending the syntaxes and security games of the PKI-LR-AMRS scheme (Tsai *et al.*, 2022) in the PKI-PK and the leakage-resilient anonymous multi-recipient encryption (CL-LR-AMRE) scheme (Xie *et al.*, 2023) in the CL-PKC, a new syntax and three

security games of our proposed LRSC-AMRS scheme are demonstrated, respectively, to define the associated framework and three security properties, namely, encryption confidentiality, recipient anonymity and sender (i.e. BMC) authentication. In the generic bilinear pairing group (GBPG) model (Boneh *et al.*, 2005), based on two security assumptions of the discrete logarithm (DL) and secure hash function (SHF), we will use three security theorems, respectively, to prove the three security properties. Finally, by comparing with the related schemes, our LRSC-AMRS scheme has four merits as follows. (1) It is the *first* LRSC-AMRS scheme suitable for heterogeneous PKCs. (2) Multiple recipients in the LRSC-AMRS scheme can be initial recipients in the PKI-PKC, or new and upgraded recipients in the CL-PKC. (3) Adversaries are allowed to continuously acquire partial information of secret keys for multiple rounds, the LRSC-AMRS scheme possesses unbounded leakage resilience. (4) The computational cost of the unsigncryption algorithm is constant.

1.4. *Paper Structure*

The remaining sections are organized as follows. Section 2 introduces four preliminaries. In Section 3, the new syntax and three security games of the LRSC-AMRS scheme are demonstrated to define the associated framework and three security properties, respectively. The LRSC-AMRS scheme is proposed in Section 4. Section 5, based on three security games, three theorems are formally shown. The comparisons and performance analysis between the LRSC-AMRS scheme and some related schemes are demonstrated in Section 6. In Section 7, conclusions are given.

2. Preliminaries

2.1. *Bilinear Pairing Group Set*

Let $G = \langle Q \rangle$ and $G_e = \langle Q_e \rangle$, respectively, denote an addition group and a multiplication group with the same prime order p, where Q and Q_e are the associated group generators. Also, there exists a bilinear pairing mapping $\hat{e}$: $G \times G \rightarrow G_e$. These parameters above form a bilinear pairing group set $BPGS = \{G, G_e, \hat{e}, p, Q, Q_e\}$ (Boneh and Franklin, 2001). The set BPGS possesses three properties as presented below.

- Non-degenerating: $\hat{e}(Q, Q) = Q_e \neq 1$.
- Efficient computing: $\hat{e}(a \cdot Q, b \cdot Q)$ can be efficiently computed, for all $a, b \in {Z_p}^*$.
- Bilinear pairing: $\hat{e}(a \cdot Q, b \cdot Q) = \hat{e}(Q, Q)^{ab} = {Q_e}^{ab}$, for all $a, b \in {Z_p}^*$.

2.2. *GBPG Model*

Here, let us introduce a technique in proving security theorems for cryptographic schemes by using the set $BPGS = \{G, G_e, \hat{e}, p, Q, Q_e\}$, namely, the generic bilinear pairing

group (GBPG) model (Boneh *et al.*, 2005). The GBPG model is embedded in the adversary games of security theorems that are played by adversaries (querier) and a challenger (responder). When adversaries would like to perform all operations in the set $BPGS = \{G, G_e, \hat{e}, p, Q, Q_e\}$, namely, the addition in G, the multiplication in G_e and the bilinear pairing mapping $\hat{e}$, they must issue the corresponding queries (oracles) to the challenger. Upon receiving these queries, the challenger then responds the associated computation results. It is worth mentioning that all elements of G and G_e are encoded into distinct bit strings by two random mappings. Also, adversaries can issue the other queries of adversary games for security properties possessed in a cryptographic scheme. At the end of an adversary game in the GBPG model, if adversaries can find collisions on G or G_e, the *discrete logarithm* (DL) *security assumption* defined below on G or G_e will be broken.

In adversary games, the set $BPGS = \{G, G_e, \hat{e}, p, Q, Q_e\}$ has three operations OP_0, OP_e and OP_{bp} that respectively denote the addition $a \cdot Q$, the multiplication ${Q_e}^a$ and the bilinear pairing mapping $\hat{e}(a \cdot Q, b \cdot Q) = {Q_e}^{ab}$, for all $a, b \in {Z_p}^*$. Also, there are two mapping functions $\zeta : {Z_p}^* \to \Omega G$ and $\zeta_e : {Z_p}^* \to \Omega G_e$ that are employed to encode all elements of G and G_e to distinct bit-strings. Additionally, two output sets ΩG and ΩG_e satisfy $|\Omega G| = |\Omega G_e| = p$ and $\Omega G \cap \Omega G_e = \phi$, where $|\cdot|$ denotes the number of OP_0, OP_e and OP_{bp} possess the following properties.

- $OP_0(\zeta(a), \zeta(b)) \to \zeta(a + b \bmod p)$.
- $OP_e(\zeta_e(a), \zeta_e(b)) \to \zeta_e(a + b \bmod p)$.
- $OP_{bp}(\zeta(a), \zeta(b)) \to \zeta_e(a \cdot b \bmod p)$.

It is worth mentioning that $\zeta(1)$ and $\zeta_e(1)$ denote the group generators Q and Q_e, respectively.

2.3. *Security Assumptions*

In the set $BPGS = \{G, G_e, \hat{e}, p, Q, Q_e\}$, two security assumptions of our LRSC-AMRS scheme are defined below.

- **Discrete logarithm (DL) security assumption**: For an unknown $a \in Z_p^*$, and given $a \cdot Q \in G$ or $\hat{e}(Q, Q)^a \in G_e$, to compute a is hard.
- **Secure hash function (SHF) security assumption**: For a secure hash function $SH : \{0, 1\}^* \to \{0, 1\}^l$ with a fixed length l, it must satisfy one-way, weak-collision resistance and strong-collision resistance.

2.4. *Leakage Security of Secret Keys*

To measure the leakage security of secret keys due to side-channel attacks, we employ the entropy values of these secret keys to evaluate their security impact. In the entropy, a fixed-length secret key is viewed as a finite random variable. Let RV and $\mathrm{Pb}[RV = rv]$ denote a finite random variable (fixed-length secret key) and the probability of the event $RV = rv$, respectively. Also, two types of minimal entropies for single finite random variable are defined as follows.

– Minimal entropy of RV:

$$H_{\infty}(RV) = -\log_2\Big(\max_{rv} \mathrm{Pb}[RV = rv]\Big).$$

– Conditionally minimal entropy of RV under the event E:

$$\widetilde{H}_{\infty}(RV|E) = -\log_2\Big(E\Big[\max_{rv}[RV = rv|E]\Big]\Big).$$

To consider the entropy of single secret key (i.e. finite random variable RV) under a leakage function LF, Dodis *et al.* (2008) derived an inequality between two types of minimal entropies (Lemma 1). For multiple fixed-length secret keys (e.g. finite random variables $RV_0, RV_1, \ldots, RV_{n-1}$), Galindo and Vivek (2013) obtained Lemma 2 below.

Lemma 1. *Let $LF : RV \rightarrow \{0,1\}^{\gamma}$ be a leakage function of RV with a fixed-bit-length γ. Thus, we have $\widetilde{H}_{\infty}(RV|LF(RV)) \geqq H_{\infty}(RV) - \gamma$.*

Lemma 2. *Let $MP \in Z_p^*[RV_0, RV_1, \ldots, RV_{n-1}]$ be a multiple-variable polynomial with maximal degree d. For $i = 0, 1, \ldots, n-1$, let PD_i be mutually independent probability distributions of $RV_i = rv_i \leftarrow Z_p^*$ such that both $H_{\infty}(PD_i) \geqq \log p - \gamma$ and $0 \leqq \gamma \leqq \log p$. If $\gamma < \log p(1-\epsilon)$ and ϵ is a positive decimal, the probability $Pb[MP(RV_0 = rv_0, RV_1 = rv_1, \ldots, RV_{n-1} = rv_{n-1}) = 0] \leqq 2^{\gamma}(d/p)$ is negligible.*

3. Syntax (Framework) and Adversary Model of the LRSC-AMRS Scheme

In this section, we present the syntax (framework) and adversary model of the leakage-resilient and seamlessly compatible anonymous multi-recipient signcryption (LRSC-AMRS) scheme in heterogeneous public-key cryptographies. For convenience, we first present the denotations of several symbols in Table 1.

3.1. *Syntax (Framework)*

The LRSC-AMRS scheme consists of two PKCs, namely, the PKI-PKC and the CL-PKC. The CA is responsible for managing the BMC and initial recipients in the PKI-PKC. Also, the KGA is responsible for managing new and upgraded recipients in the CL-PKC. The CA and the KGA first decide their system secret/public key pairs (SSK_{CA}, SPK_{CA}) and (SSK_{KGA}, SPK_{KGA}), respectively. The key generating procedures for the BMC and three types of recipients are described in the following and also depicted in Fig. 3.

- The BMC: The BMC with identity $PKID_{BMC}$ decides the secret/public key pair (SK_{BMC}, PK_{BMC}) while conveying $(PKID_{BMC}, PK_{BMC})$ to the CA. The CA creates and sends back the certificate $CRTF_{BMC}$.

Table 1
Denotations of symbols.

Symbols	Denotations
PKC	Public-key cryptography
PKI-PKC	Public-key infrastructure PKC
CL-PKC	Certificateless PKC
CA	The certificate authority (CA) in the PKI-PKC
(SSK_{CA}, SPK_{CA})	The system secret/public key pair of the CA
KGA	The key generating authority (KGA) in the CL-PKC
(SSK_{KGA}, SPK_{KGA})	The system secret/public key pair of the KGA
BMC	The broadcast management centre (BMC) in the PKI-PKC
$PKID_{BMC}$	The identity of the BMC
(SK_{BMC}, PK_{BMC})	The secret/public key pair of the BMC
$CRTF_{BMC}$	The certificate of the BMC
$PKID_r$	The identity of a recipient in the PKI-PKC
(SK_r, PK_r)	The secret/public key pair of the recipient $PKID_r$
$CRTF_r$	The certificate of $PKID_r$
$CLID_r$	The identity of a recipient in the CL-PKC
(ISK_r, IPK_r)	The individual secret/public key pair of the recipient $CLID_r$
(MSK_r, MPK_r)	The member secret/public key pair of the recipient $CLID_r$
PD	A plaintext data
ED	An encrypted data
SEF/SDF	The symmetric encrypting/decrypting functions
edk	$SEF_{edk}()/SDF_{edk}()$, where edk is an encrypting/decrypting key
$SDHR$	A set of designated heterogeneous recipients, $SDHR = \{[(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n\}$
BCS	A broadcast ciphertext set generated by the BMC

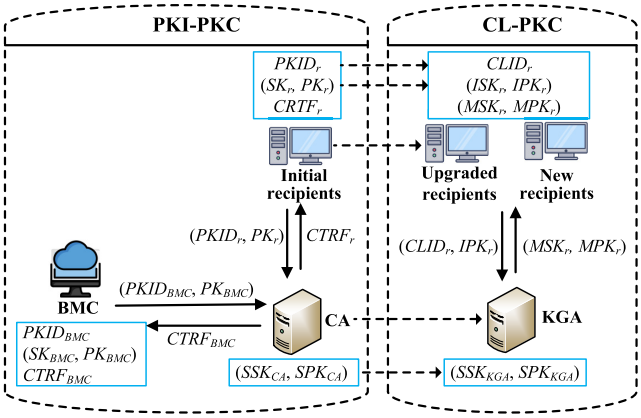

Fig. 3. Key generating procedures of the LRSC-AMRS scheme.

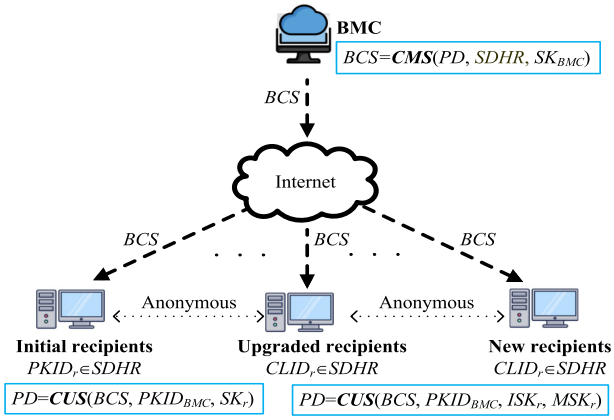

Fig. 4. The usages of the CMS and CUS algorithms in the LRSC-AMRS scheme.

- Initial recipients: An initial recipient with identity $PKID_r$ decides the secret/public key pair (SK_r, PK_r) while conveying $(PKID_r, PK_r)$ to the CA. The CA creates and sends back the certificate $CRTF_r$.
- New recipients: A new recipient with identity $CLID_r$ decides the individual secret/public key pair (ISK_r, IPK_r) while conveying $(CLID_r, IPK_r)$ to the KGA. The KGA creates and sends back the member secret/public key pair (MSK_r, MPK_r).
- Upgraded recipients: If an initial recipient with identity $PKID_r$ would like to upgrade to the CL-PKC, she/he renames $PKID_r$ to $CLID_r$ and $(PKID_r, PK_r)$ to (ISK_r, IPK_r). Also, the upgraded recipient conveys $(CLID_r, IPK_r)$ to the KGA. The KGA creates and sends back the member secret/public key pair (MSK_r, MPK_r).

In the LRSC-AMRS scheme, when the BMC would like to convey a plaintext data (PD) to a set of designated heterogeneous recipients, namely, $SDHR = \{[(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n\}$, the BMC carries out the *Compatible multi-signcryption* (*CMS*) algorithm to generate a broadcast ciphertext set $BCS = CMS(PD, SDHR, SK_{BMC})$. If some $PKID_r$ or $CLID_r$ lies in the set $SDHR$, the recipient $PKID_r$ or $CLID_r$ performs the *Compatible unsigncryption* (*CUS*) algorithm to get and validate $PD = CUS(BCS, PKID_{BMC}, SK_r)$ or $PD = CUS(BCS, PKID_{BMC}, ISK_r, MSK_r)$, respectively. Figure 4 depicts the usages of the *CMS* and *CUS* algorithms in the LRSC-AMRS scheme. For resisting side-channel attacks, the key refreshing procedure (Kiltz and Pietrzak, 2010; Galindo and Vivek, 2013; Tsai *et al.*, 2022) is used to realize leakage resilient property. For each secret/public key pair in the scheme, the secret key is separated into two fragments which must be refreshed before they are used to some computations in the proposed scheme. Also, the associated public key remains unchanged. Hence, SSK_{CA} and SSK_{KGA} are initially separated into $(SSK_{CA,0,a}, SSK_{CA,0,b})$ and $(SSK_{KGA,0,a}, SSKK_{GA,0,b})$. Also, SK_{BMC}, SK_r, ISK_r and MSK_r are separated into $(SK_{BMC,0,a}, SK_{BMC,0,b})$, $(SK_{r,0,a}, SK_{r,0,b})$, $(ISK_{r,0,a}, ISK_{r,0,b})$ and $(MSK_{r,0,a}, MSK_{r,0,b})$. Finally, the syntax (framework) of the LRSC-AMRS scheme is presented in Definition 1.

Definition 1 (*Syntax*). An LRSC-AMRS scheme in two heterogeneous PKCs (including the PKI-PKC and the CL-PKC) has four phases.

– *Initialization phase*: Firstly, the heterogeneously public system parameters (*HPSP*) of both the PKI-PKC and the CL-PKC are decided. Also, the CA in the PKI-PKC and the KGA in the CL-PKC decide their own secret/public pairs as follows.
  • PKI-PKC: By *HPSP*, the CA decides the system secret/public key pair (SSK_{CA}, SPK_{CA}). Also, the CA separates SSK_{CA} into two fragments ($SSK_{CA,0,a}$, $SSK_{CA,0,b}$) by the key refreshing procedure.
  • CL-PKC: By *HPSP*, the KGA decides the system secret/public key pair (SSK_{KGA}, SPK_{KGA}). Also, the KGA separates SSK_{KGA} into two fragments ($SSK_{KGA,0,a}$, $SSK_{KGA,0,b}$) by the key refreshing procedure.
– At the end of this phase, *HPSP*, SPK_{CA} and SPK_{KGA} are publicly announced.
– *Key generating phase*: In the PKI-PKC, the CA is responsible for managing the BMC and initial recipients. Also, in the CL-PKC, the KGA is responsible to managing the upgraded and new recipients. The associated key generating procedures are presented as follows.
  • PKI-PKC:
    (1) In the PKI-PKC, the BMC or an initial recipient with identity $PKID_r$ first decides their own secret/public key pair (SK_{BMC}, PK_{BMC}) or (SK_r, PK_r) while separating SK_{BMC} or SK_r into ($SK_{BMC,0,a}$, $SK_{BMC,0,b}$) or ($SK_{r,0,a}$, $SK_{r,0,b}$), respectively. Also, they convey ($PKID_{BMC}$, PK_{BMC}) or ($PKID_r$, PK_r) to the CA, respectively.
    (2) Upon receiving ($PKID_{BMC}$, PK_{BMC}) or ($PKID_r$, PK_r), in the i-th round of this procedure, the CA first refreshes ($SSK_{CA,i-1,a}$, $SSK_{CA,i-1,b}$) to get ($SSK_{CA,i,a}$, $SSK_{CA,i,b}$) such that $SSK_{CA} = SSK_{CA,0,a} \cdot SSK_{CA,0,b} = SSK_{CA,1,a} \cdot SSK_{CA,1,b} = \cdots = SSK_{CA,i-1,a} \cdot SSK_{CA,i-1,b} = SSK_{CA,i,a} \cdot SSK_{CA,i,b}$. The CA then uses ($SSK_{CA,i,a}$, $SSK_{CA,i,b}$) to create and send back the certificate $CRTF_{BMC}$ or $CRTF_r$ to the BMC or the recipient $PKID_r$, respectively.
  • CL-PKC:
    (1) In the CL-PKC, a new recipient with identity $CLID_r$ first decides the individual secret/public key pair (ISK_r, IPK_r) while separating ISK_r into ($ISK_{r,0,a}$, $ISK_{r,0,b}$). Also, the recipient $CLID_r$ conveys ($CLID_r$, IPK_r) to the KGA.
    (2) Upon receiving ($CLID_r$, IPK_r), in the i-th round of this procedure, the KGA refreshes ($SSK_{KGA,i-1,a}$, $SSK_{KGA,i-1,b}$) to get ($SSK_{KGA,i,a}$, $SSK_{KGA,i,b}$) such that $SSK_{KGA} = SSK_{KGA,0,a} \cdot SSK_{KGA,0,b} = SSK_{KGA,1,a} \cdot SSK_{KGA,1,b} = \cdots = SSK_{KGA,i-1,a} \cdot SSK_{KGA,i-1,b} = SSK_{KGA,i,a} \cdot SSK_{KGA,i,b}$. The KGA then uses ($SSK_{KGA,i,a}$, $SSK_{KGA,i,b}$) to create and send back the member secret/public key pair (MSK_r, MPK_r) to the recipient $CLID_r$.
    (3) Upon receiving (MSK_r, MPK_r), the recipient $CLID_r$ separates MSK_r into ($MSK_{r,0,a}$, $MSK_{r,0,b}$). Finally, the recipient $CLID_r$ has the private key pair (ISK_r, MSK_r) and pubic key pair (IPK_r, MPK_r).

  When an initial recipient with identity $PKID_r$ in the PKI-PKC would like to upgrade to the CL-PKC, she/he renames $PKID_r$ to $CLID_r$ and (SK_r, PK_r) to (ISK_r, IPK_r),

respectively. The upgraded recipient $CLID_r$ then runs the steps (2) and (3) above to get the private key pair (ISK_r, MSK_r) and pubic key pair (IPK_r, MPK_r).

– *Compatible multi-signcryption (CMS) phase*: When the BMC would like to convey a plaintext data (PD) to a set of designated heterogeneous recipients, namely, $SDHR = \{[(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)],\ r = 1, 2, \ldots, n\}$ in the j-th round of this procedure, the BMC carries out the CMS algorithm to generate a broadcast ciphertext set BCS by running the following steps.
  (1) The BMC refreshes $(SK_{BMC,j-1,a}, SK_{BMC,j-1,b})$ to get $(SK_{BMC,j,a}, SK_{BMC,j,b})$ such that $SK_{BMC} = SK_{BMC,0,a} \cdot SK_{BMC,0,b} = SK_{BMC,1,a} \cdot SK_{BMC,1,b} = \cdots = SK_{BMC,j-1,a} \cdot SK_{BMC,j-1,b} = SK_{BMC,j,a} \cdot SK_{BMC,j,b}$.
  (2) By taking PD and $SDHR$ as input , the BMC generates $BCS = CMS(PD, SDHR, (SK_{BMC,j,a}, SK_{BMC,j,b}))$.

– *Compatible unsigncryption (CUS) phase*: Upon receiving BCS, if the recipients $PKID_r$ in the PKI-PKC or the recipients $CLID_r$ in the CL-PKC lie in the set $SDHR$, they carry out the following procedures, respectively.
  - PKI-PKC: For the k-th round of this procedure, the recipient $PKID_r$ first refreshes $(SK_{r,k-1,a}, SK_{r,k-1,b})$ to get $(SK_{r,k,a}, SK_{r,k,b})$ such that $SK_r = SK_{r,0,a} \cdot SK_{r,0,b} = SK_{r,1,a} \cdot SK_{r,1,b} = \cdots = SK_{r,k-1,a} \cdot SK_{r,k-1,b} = SK_{r,k,a} \cdot SK_{r,k,b}$. The recipient $PKID_r$ carries out the CUS algorithm to get and validate $PD = CUS(BCS, PKID_{BMC}, (SK_{r,k,a}, SK_{r,k,b}))$.
  - CL-PKC: For the k-th round of this procedure, the recipient $CLID_r$, respectively, refreshes $(ISK_{r,k-1,a}, ISK_{r,k-1,b})$ and $(MSK_{r,k-1,a}, MSK_{r,k-1,b})$ to get $(ISK_{r,k,a}, ISK_{r,k,b})$ and $(MSK_{r,k,a}, MSK_{r,k,b})$ such that $ISK_r = ISK_{r,0,a} \cdot ISK_{r,0,b} = ISK_{r,1,a} \cdot ISK_{r,1,b} = \cdots = ISK_{r,k-1,a} \cdot ISK_{r,k-1,b} = ISK_{r,k,a} \cdot ISK_{r,k,b}$ and $MSK_r = MSK_{r,0,a} \cdot MSK_{r,0,b} = MSK_{r,1,a} \cdot MSK_{r,1,b} = \cdots = MSK_{r,k-1,a} \cdot MSK_{r,k-1,b} = MSK_{r,k,a} \cdot MSK_{r,k,b}$. The recipient $CLID_r$ carries out the CUS algorithm to get and validate $PD = CUS(BCS, PKID_{BMC}, (ISK_{r,k,a}, ISK_{r,k,b}), (MSK_{r,k,a}, MSK_{r,k,b}))$.

3.2. *Adversary Model*

By extending the adversary models of the PKI-LR-AMRS scheme (Tsai *et al.*, 2022) in the PKI-PKC and the CL-LR-AMRE scheme (Xie *et al.*, 2023) in the CL-PKC, we define the adversary model of the LRSC-AMRS scheme in two heterogeneous PKCs (including the PKI-PKC and the CL-PKC) in this section.

As mentioned earlier, to measure the leakage security of secret keys due to side-channel attacks, we employ the entropy values of these secret keys to evaluate their security impact. Also, we have cited two results (i.e. Lemmas 1 and 2) about the entropies of secret keys under leakage functions. Here, we define two leakage functions f and h for the two fragments of each secret key in the proposed scheme, where Δf and Δh denote the associated outputs of f and h, respectively. Hence, we have five pairs of leakage functions as defined below.

– $\Delta f_{CA,i} = f_{CA,i}(SSK_{CA,i,a})$ and $\Delta h_{CA,i} = h_{CA,i}(SSK_{CA,i,b})$ for the CA's system secret key.

- $\Delta f_{KGA,i} = f_{KGA,i}(SSK_{KGA,i,a})$ and $\Delta h_{KGA,i} = h_{KGA,i}(SSK_{KGA,i,b})$ for the KGA's system secret key.
- $\Delta f_{BMC,j} = f_{BMC,j}(SK_{BMC,j,a})$ and $\Delta h_{BMC,j} = h_{BMC,j}(SK_{BMC,j,b})$ for the BMC's secret key.
- $\Delta f_{PKIDr,k} = f_{PKIDr,k}(SK_{r,k,a})$ and $\Delta h_{PKIDr,k} = h_{PKIDr,k}(SK_{r,k,b})$ for the recipient $PKID_r$'s secret key.
- $\Delta f_{CLIDr,k} = f_{CLIDr,k}(ISK_{r,k,a}, MSK_{r,0,a})$ and $\Delta h_{CLIDr,k} = h_{CLIDr,k}(ISK_{r,k,b}, MSK_{r,0,b})$ for the recipient $CLID_r$'s individual and member secret keys.

In the adversary model of the proposed LRSC-AMRS scheme, there are two types of adversaries whose abilities and restrictions are presented as follows.

- Illegal recipient (A_I):
  - A_I may acquire SK_r of any recipient $PKID_r$, as well as both ISK_r and MSK_r of any recipient $CLID_r$.
  - A_I is disallowed to acquire SK_r^* of the target recipient $PKID_r^*$, but it may acquire partial information of SK_r^* by two leakage functions $f_{PKIDr,k}$ and $h_{PKIDr,k}$.
  - A_I is disallowed to acquire MSK_r^* of the target recipient $CLID_r^*$, but it may acquire partial information of MSK_r^* by two leakage functions $f_{CLIDr,k}$ and $h_{CLIDr,k}$ defined above.
  - A_I may acquire partial information of SSK_{CA} by two leakage functions $f_{CA,i}$ and $h_{CA,i}$.
  - A_I may acquire partial information of SSK_{KGA} by two leakage functions $f_{KGA,i}$ and $h_{KGA,i}$.
- Malicious KGA (A_{II}): It is assumed that A_{II} possesses the system secret key SSK_{KGA} of the KGA.
  - A_{II} may acquire SK_r of any recipient $PKID_r$, and both ISK_r and MSK_r of any recipient $CLID_r$.
  - A_{II} is disallowed to acquire SK_r^* of the target recipient $PKID_r^*$, but it may acquire partial information of SK_r^* by $f_{PKIDr,k}$ and $h_{PKIDr,k}$.
  - A_{II} is disallowed to acquire ISK_r^* of the target recipient $CLID_r^*$, but it may acquire partial information of ISK_r^* by $f_{CLIDr,k}$ and $h_{CLIDr,k}$.

In the LRSC-AMRS scheme, we employ three games, respectively, to model three security properties, namely, encryption confidentiality, recipient anonymity and sender (i.e. BMC) authentication. The encryption confidentiality is modelled by the encryption indistinguishability game under chosen-ciphertext attacks (LRSC-EIND-CCA game) while the recipient anonymity is modelled by the recipient indistinguishability game under chosen-ciphertext attacks (LRSC-RIND-CCA game). Also, the sender (i.e. the BMC) authentication is modelled by the existential unforgeability game under adaptive chosen-message attacks (LRSC-EU-ACMA game). Three games are, respectively, presented in Definitions 2, 3 and 4 below, which are played by a probabilistic polynomial-time (PPT) adversary A (A_I or A_{II}) and a challenger C. It is worth mentioning that adversaries (including illegal recipient and malicious KGA) are allowed to continuously acquire partial information of secret keys for multiple rounds. By the key refreshing procedure (Kiltz and Pietrzak, 2010;

Galindo and Vivek, 2013; Tsai *et al.*, 2022), any two leaked partial information of a secret key are mutually independent.

Definition 2 (*LRSC-EIND-CCA game*). The LRSC-AMRS scheme achieves the encryption confidentiality if no PPT adversary A (A_I or A_{II}) with a non-negligible advantage wins the LRSC-EIND-CCA game as shown below.

- *Setup*: By running the *Initialization phase* of the LRSC-AMRS scheme, the challenger C sets $HPSP$, (SSK_{CA}, SPK_{CA}), and (SSK_{KGA}, SPK_{KGA}). If A is type of A_{II}, SSK_{KGA} is conveyed to A. Finally, $HPSP$, SPK_{CA} and SPK_{KGA} are publicly announced.
- *Query*: A may adaptively request to C the following queries.
  - *Secret key query* ($PKID_{BMC}/PKID_r$): By $PKID_{BMC}$ or $PKID_r$, C returns (SK_{BMC}, PK_{BMC}) or (SK_r, PK_r).
  - *Certificate query* (($PKID_{BMC}$, PK_{BMC})/($PKID_r$, PK_r)): For the i-th request of this query, C refreshes ($SS_{KCA,i-1,a}$, $SS_{KCA,i-1,b}$) to get ($SS_{KCA,i,a}$, $SS_{KCA,i,b}$). By ($PKID_{BMC}$, PK_{BMC}) or ($PKID_r$, PK_r), C uses ($SS_{KCA,i,a}$, $SSK_{CA,i,b}$) to create and send back $CRTF_{BMC}$ or $CRTF_r$ to the BMC or the recipient $PKID_r$, respectively.
  - *Certificate leakage query* (i, $f_{CA,i}$, $h_{CA,i}$): For the i-th *Certificate query*, A may request this *leakage* query only once. C returns $\Delta f_{CA,i} = f_{CA,i}(SSK_{CA,i,a})$ and $\Delta h_{CA,i} = h_{CA,i}(SSK_{CA,i,b})$.
  - *Individual secret key query* ($CLID_r$): By $CLID_r$, C returns (ISK_r, IPK_r) if the *Public key replacement query* ($CLID_r$, (IPK'_r, MPK'_r)) is never requested. Otherwise, C returns "failure".
  - *Member secret key query* ($CLID_r$, IPK_r): For the i-th request of this query, C refreshes ($SSK_{KGA,i-1,a}$, $SSK_{KGA,i-1,b}$) to get ($SSK_{KGA,i,a}$, $SSK_{KGA,i,b}$). By ($CLID_r$, IPK_r), C uses ($SSK_{KGA,i,a}$, $SSK_{KGA,i,b}$) to create and send back (MSK_r, MPK_r).
  - *Member secret key leakage query* (i, $f_{KGA,i}$, $h_{KGA,i}$): For the i-th *Member secret key query*, A may request this *leakage* query only once. C returns $\Delta f_{KGA,i} = f_{KGA,i}(SSK_{KGA,i,a})$ and $\Delta h_{KGA,i} = h_{KGA,i}(SSK_{KGA,i,b})$.
  - *Public key replacement query* ($CLID_r$, (IPK'_r, MPK'_r)): C records this replacement.
  - *Compatible multi-signcryption* (*CMS*) *query* (PD, $SDHR$, $PKID_{BMC}$): For the j-th request of this query, C refreshes ($SK_{BMC,j-1,a}$, $SK_{BMC,j-1,b}$) to get ($SK_{BMC,j,a}$, $SK_{BMC,j,b}$). By (PD, $SDHR$), C uses ($SK_{BMC,j,a}$, $SK_{BMC,j,b}$) to create and send back $BCS = CMS(PD, SDHR, (SK_{BMC,j,a}, SK_{BMC,j,b}))$.
  - *Compatible multi-signcryption* (*CMS*) *leakage query* (j, $f_{BMC,j}$, $h_{BMC,j}$): For the j-th *Compatible multi-signcryption (CMS) query*, A may request this *leakage* query only once. C returns $\Delta f_{BMC,j} = f_{BMC,j}(SK_{BMC,j,a})$ and $\Delta h_{BMC,j} = h_{BMC,j}(SK_{BMC,j,b})$.
  - *Compatible unsigncryption* (*CUS*) *query* ($PKID_r/CLID_r$, BCS): For the k-th request of this query with $PKID_r$ or $CLID_r$, C runs the following associated procedures.
    (1) For $PKID_r$, C refreshes ($SK_{r,k-1,a}$, $SK_{r,k-1,b}$) to get ($SK_{r,k,a}$, $SK_{r,k,b}$). C returns $PD = CUS(BCS, PKID_{BMC}, (SK_{r,k,a}, SK_{r,k,b}))$.

(2) For $CLID_r$, C refreshes $(ISK_{r,k-1,a}, ISK_{r,k-1,b})$ and $(MSK_{r,k-1,a}, MSK_{r,k-1,b})$, respectively, to get $(ISK_{r,k,a}, ISK_{r,k,b})$ and $(MSK_{r,k,a}, MSK_{r,k,b})$. C returns $PD = CUS(BCS, PKID_{BMC}, (ISK_{r,k,a}, ISK_{r,k,b}), (MSK_{r,k,a}, MSK_{r,k,b}))$.

- *Compatible unsigncryption* (*CUS*) *leakage query* $(k, (f_{PKIDr,k}, h_{PKIDr,k})/(f_{CLIDr,k}, h_{CLIDr,k}))$: For the k-th *Compatible unsigncryption (CUS) query* with $PKID_r$ or $CLID_r$. For $PKID_r$, C sends back $\Delta f_{PKIDr,k} = f_{PKIDr,k}(SK_{r,k,a})$ and $\Delta h_{PKIDr,k} = h_{PKIDr,k}(SK_{r,k,b})$. For $CLID_r$, C sends back $\Delta f_{CLIDr,k} = f_{CLIDr,k}(ISK_{r,k,a}, MSK_{r,0,a})$ and $\Delta h_{CLIDr,k} = h_{CLIDr,k}(ISK_{r,k,b}, MSK_{r,0,b})$.

– *Challenge*: A conveys a plaintext data pair (PD_1, PD_2) and $SDHR = \{[(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n\}$ to C. C selects a random value $\lambda \in \{1, 2\}$ and refreshes $(SK_{BMC,j-1,a}, SK_{BMC,j-1,b})$ to get $(SK_{BMC,j,a}, SK_{BMC,j,b})$. Finally, C generates and sends back $BCS = CMS(PD_\lambda, SDHR, (SK_{BMC,j,a}, SK_{BMC,j,b}))$. In addition, the following two conditions must be satisfied.
  1. For A_I, it cannot request the *Secret key query* $(PKID_r)$ or *Member secret key query* $(CLID_r, IPK_r)$, for $r = 1, 2, \ldots, n$.
  2. For A_{II}, it cannot request the *Secret key query* $(PKID_r)$, *Individual secret key query* $(CLID_r)$ or *Public key replacement query* $(CLID_r, (IPK'_r, MPK'_r))$, for $r = 1, 2, \ldots, n$.
– *Guess*: If A outputs $\lambda' \in \{1, 2\}$ and $\lambda' = \lambda$, it means that A wins the LRSC-EIND-CCA game and the associated advantage is $Adv(A) = |\text{Pb}[\lambda' = \lambda] - 1/2|$.

Definition 3 (*LRSC-RIND-CCA game*). The LRSC-AMRS scheme achieves the recipient anonymity if no PPT adversary A (A_I or A_{II}) with a non-negligible advantage wins the LRSC-RIND-CCA game as shown below.

– *Setup* and *Query* are the same as those of Definition 2.
– *Challenge*: A conveys a plaintext data PD and $SDHR = \{[(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n+1\}$ to C. C selects a random value $\lambda \in \{1, 2\}$ and sets $SDHR' = \{[(PKID_\lambda, PK_\lambda) \| (CLID_\lambda, IPK_\lambda, MPK_\lambda)], [(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)], r = 3, \ldots, n+1\}$. Finally, C refreshes $(SK_{BMC,j-1,a}, SK_{BMC,j-1,b})$ to get $(SK_{BMC,j,a}, SK_{BMC,j,b})$, and generates and sends back $BCS = CMS(PD, SDHR', (SK_{BMC,j,a}, SK_{BMC,j,b}))$. In addition, the following two conditions must be satisfied.
  1. For A_I, it cannot request the *Secret key query* $(PKID_\lambda)$ or *Member secret key query* $(CLID_\lambda, IPK_\lambda)$, for $\lambda = 1$ and 2.
  2. For A_{II}, it cannot request the *Secret key query* $(PKID_\lambda)$, *Individual secret key query* $(CLID_\lambda)$ or *Public key replacement query* $(CLID_\lambda, (IPK'_\lambda, MPK'_\lambda))$, for $\lambda = 1$ and 2.
– *Guess*: If A outputs $\lambda' \in \{1, 2\}$ and $\lambda' = \lambda$, it means that A wins the LRSC-RIND-CCA game and the associated advantage is $Adv(A) = |\text{Pb}[\lambda' = \lambda] - 1/2|$.

Definition 4 (*LRSC-EU-ACMA game*). The LRSC-AMRS scheme achieves the BMC authentication if no PPT adversary A (i.e. impersonating the BMC) with a non-negligible advantage wins the LRSC-EU-ACMA game as shown below.

– *Setup* and *Query* are the same as those of Definition 2.
– *Forgery*: A forges and sends C a broadcast ciphertext set BCS' for a plaintext data PD and $SDHR = \{[(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n\}$. For any recipients $PKID_r$ and $CLID_r$ in $SDHR$, if they may, respectively, carry out the CUS algorithm to get and validate $PD = CUS(BCS', PKID_{BMC}, (SK_{r,k,a}, SK_{r,k,b}))$ or $PD = CUS(BCS', PKID_{BMC}, (ISK_{r,k,a}, ISK_{r,k,b}), (MSK_{r,k,a}, MSK_{r,k,b}))$, then A wins the LRSC-EU-ACMA game. Note that A cannot request the *Secret key query* $(PKID_{BMC})$.

4. The Proposed LRSC-AMRS Scheme

According to the syntax presented in Definition 1, the proposed LRSC-AMRS scheme in two heterogeneous PKCs (including the PKI-PKC and the CL-PKC) has four phases that are presented as follows.

– *Initialization phase*: By the bilinear pairing group set $BPGS = \{G, G_e, \hat{e}, p, Q, Q_e\}$ defined in previous section, the heterogeneously public system parameters ($HPSP$) about the PKI-PKC and the CL-PKC are decided as $HPSP = \{G, G_e, \hat{e}, p, Q, Q_e, A, B, SEF/SDF, SH_0, SH_1, SH_2, SH_3, SH_4, SH_5\}$, where $A, B \in G$, SEF/SDF are symmetric encrypting/decrypting functions, and $SH_0 : \{0,1\}^* \times G \times G \to \{0,1\}^l$, $SH_1 : G \to \{0,1\}^l$, $SH_2 : G \times G \to \{0,1\}^l$, $SH_3, SH_4 : \{0,1\}^l \to \{0,1\}^l$ and $SH_5 : G \times \{0,1\}^* \to \{0,1\}^l$ are six secure hash functions. Also, the CA in the PKI-PKC and the KGA in the CL-PKC decide their own secret/public pairs as follows.
  • PKI-PKC: By $HPSP$, the CA selects two random values $s, t_0 \in Z_p^*$, and decides the system secret/public key pair (SSK_{CA}, SPK_{CA}), where $SSK_{CA} = s \cdot Q$ and $SPK_{CA} = \hat{e}(Q, s \cdot Q)$. Also, the CA uses the key refreshing procedure to separate SSK_{CA} into a pair of two fragments $(SSK_{CA,0,a}, SSK_{CA,0,b})$, where $SSK_{CA,0,a} = t_0 \cdot Q$ and $SSK_{CA,0,b} = SSK_{CA} - t_0 \cdot Q$.
  • CL-PKC: By $HPSP$, the KGA selects two random values $u, v_0 \in Z_p^*$, and decides the system secret/public key pair (SSK_{KGA}, SPK_{KGA}), where $SSK_{KGA} = u \cdot Q$ and $SPK_{KGA} = \hat{e}(Q, u \cdot Q)$. Also, the KGA uses the key refreshing procedure to separate SSK_{KGA} into a pair of two fragments $(SSK_{KGA,0,a}, SSK_{KGA,0,b})$, where $SSK_{KGA,0,a} = v_0 \cdot Q$ and $SSK_{KGA,0,b} = SSK_{KGA} - v_0 \cdot Q$.
– At the end of this phase, $HPSP$, SPK_{CA} and SPK_{KGA} are publicly announced.
– *Key generating phase*: In the PKI-PKC, the CA is responsible for managing the BMC and initial recipients. Also, in the CL-PKC, the KGA is responsible for managing upgraded and new recipients. The associated key generating procedures are presented as follows.
  • PKI-PKC:
    (1) In the PKI-PKC, the BMC selects two random values $w, x_0 \in Z_p^*$, and decides her/his own secret/public key pair (SK_{BMC}, PK_{BMC}) while separating SK_{BMC} into $(SK_{BMC,0,a}, SK_{BMC,0,b})$, where $SK_{BMC} = w \cdot Q$, $PK_{BMC} = \hat{e}(Q, w \cdot Q)$, $SK_{BMC,0,a} = x_0 \cdot Q$ and $SK_{BMC,0,b} = SK_{BMC} - x_0 \cdot Q$. Also, for an initial

recipient with identity $PKID_r$, she/he selects two random values $y, z_0 \in Z_p^*$, and decides her/his own secret/public key pair (SK_r, PK_r) while separating SK_r into $(SK_{r,0,a}, SK_{r,0,b})$, where $SK_r = y \cdot Q$, $PK_r = \hat{e}(Q, y \cdot Q)$, $SK_{r,0,a} = z_0 \cdot Q$ and $SK_{r,0,b} = SK_r - z_0 \cdot Q$. Also, they convey $(PKID_{BMC}, PK_{BMC})$ or $(PKID_r, PK_r)$ to the CA, respectively.

(2) Upon receiving $(PKID_{BMC}, PK_{BMC})$ or $(PKID_r, PK_r)$, the CA selects a random value $t_i \in Z_p^*$, and refreshes $(SSK_{CA,i-1,a}, SSK_{CA,i-1,b})$ to get $(SSK_{CA,i,a}, SSK_{CA,i,b})$, where $SSK_{CA,i,a} = SSK_{CA,i-1,a} + t_i \cdot Q$ and $SSK_{CA,i,b} = SSK_{CA,i-1,b} - t_i \cdot Q$. By a leakage-resilient signature scheme (Tsai *et al.*, 2022), the CA then uses $(SSK_{CA,i,a}, SSK_{CA,i,b})$ to create and send back the certificate $CRTF_{BMC}$ or $CRTF_r$ to the BMC or the recipient $PKID_r$, respectively.

- CL-PKC:

(1) In the CL-PKC, a new recipient with identity $CLID_r$ selects two random values $\alpha, \beta_0 \in Z_p^*$, and decides the individual secret/public key pair (ISK_r, IPK_r) while separating ISK_r into $(ISK_{r,0,a}, ISK_{r,0,b})$, where $ISK_r = \alpha \cdot Q$, $IPK_r = \hat{e}(Q, \alpha \cdot Q)$, $ISK_{r,0,a} = \beta_0 \cdot Q$ and $ISK_{r,0,b} = ISK_r - \beta_0 \cdot Q$. Also, the recipient $CLID_r$ conveys $(CLID_r, IPK_r)$ to the KGA.

(2) Upon receiving $(CLID_r, IPK_r)$, the KGA selects a random value $v_i \in Z_p^*$, and refreshes $(SSK_{KGA,i-1,a}, SSK_{KGA,i-1,b})$ to get $(SSK_{KGA,i,a}, SSK_{KGA,i,b})$, where $SSK_{KGA,i,a} = SSK_{KGA,i-1,a} + v_i \cdot Q$ and $SSK_{KGA,i,b} = SSK_{KGA,i-1,b} - v_i \cdot Q$. The KGA uses $(SSK_{KGA,i,a}, SSK_{KGA,i,b})$ to create and send back the member secret/public key pair (MSK_r, MPK_r) to the recipient $CLID_r$ by running the following steps.

(a) Select a random value $\delta \in Z_p^*$.

(b) Compute $MPK_r = \delta \cdot Q$ and $CLTemp_i = SSK_{KGA,i,a} + \delta \cdot (A + \theta \cdot B)$, where $\theta = SH_0(CLID_r, IPK_r, MPK_r)$.

(c) Compute $MSK_r = CLTemp_i + SSK_{KGA,i,b}$.

(3) Upon receiving (MSK_r, MPK_r), the recipient $CLID_r$ selects a random value $\gamma_0 \in Z_p^*$, and separates MSK_r into $(MSK_{r,0,a}, MSK_{r,0,b})$, where $MSK_{r,0,a} = \gamma_0 \cdot Q$ and $MSK_{r,0,b} = MSK_r - \gamma_0 \cdot Q$. Finally, the recipient $CLID_r$ has the private key pair (ISK_r, MSK_r) and pubic key pair (IPK_r, MPK_r).

When an initial recipient with identity $PKID_r$ in the PKI-PKC would like to upgrade to the CL-PKC, she/he renames $PKID_r$ to $CLID_r$ and (SK_r, PK_r) to (ISK_r, IPK_r), respectively. The upgraded recipient $CLID_r$ then runs the steps (2) and (3) above to get the private key pair (ISK_r, MSK_r) and pubic key pair (IPK_r, MPK_r).

– *Compatible multi-signcryption (CMS) phase*: When the BMC would like to convey a plaintext data PD to a set of designated heterogeneous recipients, namely, $SDHR = \{[(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n\}$, the BMC carries out the *CMS* algorithm to generate a broadcast ciphertext set *BCS* by running the following steps.

(1) The BMC selects a random value $x_j \in Z_p^*$, and refreshes $(SK_{BMC,j-1,a}, SK_{BMC,j-1,b})$ to get $(SK_{BMC,j,a}, SK_{BMC,j,b})$, where $SK_{BMC,j,a} = SK_{BMC,j-1,a} + x_j \cdot Q$ and $SK_{BMC,j,b} = SK_{BMC,j-1,b} - x_j \cdot Q$.

(2) By taking PD and $SDHR$ as input, the BMC generates $BCS = CMS(PD, SDHR, (SK_{BMC,j,a}, SK_{BMC,j,b}))$ by running the following steps.
   (a) Select an encrypting/decrypting key $edk \in \{0,1\}^l$ and generate an encrypted data $ED = SEF_{edk}(PD)$.
   (b) Select a random value $m \in Z_p^*$ and compute $M = m \cdot Q$.
   (c) For $(PKID_r, PK_r)$ in $SDHR$, compute $PCK_r = (PK_r)^m$ and a common key $CK_r = SH_1(PCK_r)$.
   (d) For $(CLID_r, IPK_r, MPK_r)$ in $SDHR$, compute $CCK_{r,0} = (IPK_r)^m$, $CCK_{r,1} = (SPK_{KGA} \cdot \hat{e}(MPK_r, A + \theta \cdot B))^m$ and $CK_r = SH_2(CCK_{r,0}, CCK_{r,1})$, where $\theta = SH_0(CLID_r, IPK_r, MPK_r)$.
   (e) According to Steps (c) and (d) above, generate $C_r = SH_3(CK_r) \| (SH_4(CK_r) \oplus edk)$, for $r = 1, 2, \ldots, n$.
   (f) Compute $STemp = SK_{BMC,j,a} + m \cdot (A + \rho \cdot B)$, where $\rho = SH_5(M, C_1, C_2, \ldots, C_n, PD, ED)$.
   (g) Generate a signature $\sigma = STemp + SK_{BMC,j,b}$.
   (h) Set the broadcast ciphertext set $BCS = \langle (C_1, C_2, \ldots, C_n), M, ED, \sigma \rangle$.

– *Compatible unsigncryption* (*CUS*) *phase*: Upon receiving $BCS = \langle (C_1, C_2, \ldots, C_n), M, ED, \sigma \rangle$, if the recipient $PKID_r$ in the PKI-PKC or the recipient $CLID_r$ in the CL-PKC lie in the set $SDHR$, they carry out the following procedures.
   (1) The recipients $PKID_r$ and $CLID_r$ run the associated computations, respectively.
      • PKI-PKC: The recipient $PKID_r$ selects a random value $z_k \in Z_p^*$, and refreshes $(SK_{r,k-1,a}, SK_{r,k-1,b})$ to get $(SK_{r,k,a}, SK_{r,k,b})$, where $SK_{r,k,a} = SK_{r,k-1,a} + z_k \cdot Q$ and $SK_{r,k,b} = SK_{r,k-1,b} - z_k \cdot Q$. The recipient $PKID_r$ computes $PT_1 = \hat{e}(M, SK_{r,k,a})$, $PCK_r = PT_1 \cdot \hat{e}(M, SK_{r,k,b})$ and $CK_r = SH_1(PCK_r)$.
      • CL-PKC: The recipient $CLID_r$ selects two random values $\beta_k, \gamma_k \in Z_p^*$, and respectively refreshes $(ISK_{r,k-1,a}, ISK_{r,k-1,b})$ and $(MSK_{r,k-1,a}, MSK_{r,k-1,b})$ to get $(ISK_{r,k,a}, ISK_{r,k,b})$ and $(MSK_{r,k,a}, MSK_{r,k,b})$, where $ISK_{r,k,a} = ISK_{r,k-1,a} + \beta_k \cdot Q$, $ISK_{r,k,b} = ISK_{r,k-1,b} - \beta_k \cdot Q$, $MSK_{r,k,a} = MSK_{r,k-1,a} + \gamma_k \cdot Q$ and $MSK_{r,k,b} = MSK_{r,k-1,b} - \gamma_k \cdot Q$. Also, the recipient $CLID_r$ computes $CT_0 = \hat{e}(M, ISK_{r,k,a})$, $CCK_{r,0} = CT_0 \cdot \hat{e}(M, ISK_{r,k,b})$, $CT_1 = \hat{e}(M, MSK_{r,k,a})$, $CCK_{r,1} = CT_1 \cdot \hat{e}(M, MSK_{r,k,b})$ and $CK_r = SH_2(CCK_{r,0}, CCK_{r,1})$.
   (2) According to Step (1) above, the recipient $PKID_r$ or $CLID_r$ obtains CK_r, and computes $SH_3(CK_r)$ and $SH_4(CK_r)$.
   (3) Use $SH_3(CK_r)$ in (2) to find C_r while truncating $SH_3(CK_r)$ from C_r.
   (4) Get edk by computing $SH_4(CK_r) \oplus (SH_4(CK_r) \oplus edk)$.
   (5) Get the plaintext data $PD = SDF_{edk}(ED)$ and compute $\rho = SH_5(M, C_1, C_2, \ldots, C_n, PD, ED)$.
   (6) If $\hat{e}(Q, \sigma) = PK_{BMC} \cdot \hat{e}(M, A + \rho \cdot B)$ holds, output PD and "True"; otherwise, output "invalid".

Correctness:

The correctness about $\hat{e}(Q, \sigma) = PK_{BMC} \cdot \hat{e}(M, A + \rho \cdot B)$, $CK_r = SH_1(PCK_r)$ and $CK_r = SH_2(CCK_{r,0}, CCK_{r,1})$ in the LRSC-AMRS scheme is verified by the following

equalities.

$$\begin{aligned}
\hat{e}(Q,\sigma) &= \hat{e}(Q, STemp + SK_{BMC,j,b}) \\
&= \hat{e}\big(Q, SK_{BMC,j,a} + m\cdot(A+\rho\cdot B) + SK_{BMC,j,b}\big) \\
&= \hat{e}\big(Q, SK_{BMC} + m\cdot(A+\rho\cdot B)\big) \\
&= \hat{e}(Q, SK_{BMC})\cdot\hat{e}(M, A+\rho\cdot B) \\
&= PK_{BMC}\cdot\hat{e}(M, A+\rho\cdot B).
\end{aligned}$$

$$\begin{aligned}
PCK_r &= PT_1\cdot\hat{e}(M, SK_{r,k,b}) \\
&= \hat{e}(M, SK_{r,k,a})\cdot\hat{e}(M, SK_{r,k,b}) \\
&= \hat{e}(m\cdot Q, SK_{r,k,a} + SK_{r,k,b}) \\
&= \hat{e}(Q, SK_r)^m \\
&= (PK_r)^m.
\end{aligned}$$

$$\begin{aligned}
CCK_{r,0} &= CT_0\cdot\hat{e}(M, ISK_{r,k,b}) \\
&= \hat{e}(M, ISK_{r,k,a})\cdot\hat{e}(M, ISK_{r,k,b}) \\
&= \hat{e}(m\cdot Q, ISK_{r,k,a} + ISK_{r,k,b}) \\
&= \hat{e}(Q, ISK_r)^m \\
&= (IPK_r)^m.
\end{aligned}$$

$$\begin{aligned}
CCK_{r,1} &= CT_1\cdot\hat{e}(M, MSK_{r,k,b}) \\
&= \hat{e}(M, MSK_{r,k,a})\cdot\hat{e}(M, MSK_{r,k,b}) \\
&= \hat{e}(m\cdot Q, MSK_{r,k,a} + MSK_{r,k,b}) \\
&= \hat{e}(Q, MSK_r)^m \\
&= \hat{e}\big(Q, SSK_{KGA} + \delta\cdot(A+\theta\cdot B)\big)^m \\
&= \big(\hat{e}(Q, SSK_{KGA})\cdot\hat{e}\big(Q, \delta\cdot(A+\theta\cdot B)\big)\big)^m \\
&= \big(SPK_{KGA}\cdot\hat{e}\big(\delta\cdot Q, (A+\theta\cdot B)\big)\big)^m \\
&= \big(SPK_{KGA}\cdot\hat{e}\big(MPK_r, (A+\theta\cdot B)\big)\big)^m.
\end{aligned}$$

5. Security Analysis

In Definitions 2, 3 and 4, we have employed three games (i.e. LRSC-EIND-CCA, LRSC-RIND-CCA and LRSC-EU-ACMA) to model three security properties of the LRSC-AMRS scheme, namely, encryption confidentiality, recipient anonymity and sender (i.e. BMC) authentication. Under the GBPG model presented in Section 2.2, we will use three theorems, respectively, to prove the three security properties based on two security assumptions of the discrete logarithm (DL) and secure hash function (SHF).

Theorem 1. *In the GBPG model, based on the DL and SHF security assumptions, the LRSC-AMRS scheme achieves the encryption confidentiality in the LRSC-EIND-CCA game.*

Proof. The LRSC-EIND-CCA game is played by a PPT adversary A (A_I or A_{II}) and a challenger C, and consists of *Setup*, *Query*, *Challenge* and *Guess* as shown below.

– *Setup*: By running the *Initialization phase* of the LRSC-AMRS scheme, the challenger C sets $HPSP = \{G, G_e, \hat{e}, p, Q, Q_e, A, B, SEF/SDF, SH_0, SH_1, SH_2, SH_3, SH_4, SH_5\}$. Also, C decides (SSK_{CA}, SPK_{CA}) of the CA and (SSK_{KGA}, SPK_{KGA}) of the KGA. If A is type of A_{II}, SSK_{KGA} is conveyed to A. Finally, $HPSP$, SPK_{CA} and SPK_{KGA} are publicly announced. Meanwhile, C sets five lists LT_0, LT_e, LT_{PKI}, LT_{CL} and LT_{CMS} as follows.
  - LT_0 is set to log the i-th element of G by the pair (ΨG_i, ΩG_i), where ΨG_i and ΩG_i denote a multi-variate polynomial and the corresponding bit string. Initially, C adds (ΨQ, ΩG_1), (ΨA, ΩG_2), (ΨB, ΩG_3), (ΨSSK_{CA}, ΩG_4) and (ΨSSK_{KGA}, ΩG_5) to LT_0. Here, there is an auto-converting procedure between ΨG_i and ΩG_i for any queries in the *Query*.
  - LT_e is set to log the i-th element of G_e by the pair (ΨGE_i, ΩGE_i), where ΨGE_i and ΩGE_i denote a multi-variate polynomial and the corresponding bit string. Initially, C adds (ΨQ_e, ΩGE_1), (ΨSPK_{CA}, ΩGE_2) and (ΨSPK_{KGA}, ΩGE_3) to LT_e. Here, there is an auto-converting procedure between ΨGE_i and ΩGE_i for any queries in the *Query*.
  - LT_{PKI} is set to log the secret/public key pairs of the BMC and a recipient with identity $PKID_r$ in the PKI-PKC by the tuples ($PKID_{BMC}$, ΨSK_{BMC}, ΨPK_{BMC}) and ($PKID_r$, ΨSK_r, ΨPK_r), respectively.
  - LT_{CL} is set to log the individual secret/public and member secret/public key pairs of a recipient with identity $CLID_r$ in the CL-PKC by the tuple ($CLID_r$, ΨISK_r, ΨIPK_r, ΨMSK_r, ΨMPK_r).
  - LT_{CMS} is set to log the details of *Compatible multi-signcryption* (*CMS*) algorithm by the tuple (ΨM, ΨPCK_r/($\Psi CCK_{r,0}$, $\Psi CCK_{r,1}$), edk) for each recipient $PKID_r/CLID_r$ in the set $SDHR$.

– *Query*: A may adaptively request to C the following queries at most q times.
  - OP_0 *query* (ΩG_i, ΩG_j, *Operator*): By converting ΩG_i and ΩG_j in LT_0, C first gets the corresponding ΨG_i and ΨG_j. If *Operator* = " + ", C computes $\Psi G_k = \Psi G_i + \Psi G_j$. If *Operator* = " − ", C computes $\Psi G_k = \Psi G_i - \Psi G_j$. Finally, C adds (ΨG_k, ΩG_k) in LT_0.
  - OP_e *query* (ΩGE_i, ΩGE_j, *Operator*): By converting ΩGE_i and ΩGE_j in LT_e, C first gets the corresponding ΨGE_i and ΨGE_j. If *Operator* = " × ", C computes $\Psi GE_k = \Psi GE_i + \Psi GE_j$. If *Operator* = "/", C computes $\Psi GE_k = \Psi GE_i - \Psi GE_j$. Finally, C adds (ΨGE_k, ΩGE_k) in LT_e.
  - OP_{bp} *query* (ΩG_i, ΩG_j): By converting ΩG_i and ΩG_j in LT_0, C first gets the corresponding ΨG_i and ΨG_j. Also, C computes $\Psi GE_k = \Psi G_i \cdot \Psi G_j$ and adds (ΨG_k, ΩG_k) in LT_e.
  - *Secret key query* ($PKID_{BMC}/PKID_r$): By $PKID_{BMC}/PKID_r$, C searches ($PKID_{BMC}/PKID_r$, $\Psi SK_{BMC}/\Psi SK_r$, $\Psi PK_{BMC}/\Psi PK_r$) in LT_{PKI}. If found, C returns ($\Omega SK_{BMC}/\Omega SK_r$, $\Omega PK_{BMC}/\Omega PK_r$) by converting $\Psi SK_{BMC}/\Psi SK_r$ in LT_0 and $\Psi PK_{BMC}/\Psi PK_r$ in LT_e. If not found, C picks a new variate ΨSK_{BMC} or

ΨSK_r in G, and computes $\Psi PK_{BMC} = \Psi Q \cdot \Psi SK_{BMC}$ or $\Psi PK_r = \Psi Q \cdot \Psi SK_r$. Also, C adds $(PKID_{BMC}/PKID_r, \Psi SK_{BMC}/\Psi SK_r, \Psi PK_{BMC}/\Psi PK_r)$ in LT_{PKI}. Meanwhile, C adds $(\Psi SK_{BMC}/\Psi SK_r, \Omega SK_{BMC}/\Omega SK_r)$ in LT_0 and $(\Psi PK_{BMC}/\Psi PK_r, \Omega PK_{BMC}/\Omega PK_r)$ in LT_e, and returns $(\Omega SK_{BMC}/\Omega SK_r, \Omega PK_{BMC}/\Omega PK_r)$.

- *Certificate query* $((PKID_{BMC}, \Omega PK_{BMC})/(PKID_r, \Omega PK_r))$: For the i-th request of this query, C refreshes $(\Psi SSK_{CA,i-1,a}, \Psi SSK_{CA,i-1,b})$ to get $(\Psi SSK_{CA,i,a}, \Psi SSK_{CA,i,b})$. By $(PKID_{BMC}, \Omega PK_{BMC})$ or $(PKID_r, \Omega PK_r)$, C uses $(\Psi SSK_{CA,i,a}, \Psi SSK_{CA,i,b})$ to create and send back $CRTF_{BMC}$ or $CRTF_r$ to the BMC or the recipient $PKID_r$, respectively.
- *Certificate leakage query* $(i, f_{CA,i}, h_{CA,i})$: For the i-th *Certificate query*, A may request this *leakage* query only once. C returns $\Delta f_{CA,i} = f_{CA,i}(\Psi SSK_{CA,i,a})$ and $\Delta h_{CA,i} = h_{CA,i}(\Psi SSK_{CA,i,b})$.
- *Individual secret key query* $(CLID_r)$: By $CLID_r$, C searches $(CLID_r, \Psi ISK_r, \Psi IPK_r, \Psi MSK_r, \Psi MPK_r)$ in LT_{CL}. If found, C returns $(\Omega ISK_r, \Omega IPK_r)$ by converting ΨISK_r in LT_0 and ΨIPK_r in LT_e. If not found, C picks a new variate ΨISK_r in G, and computes $\Psi IPK_r = \Psi Q \cdot \Psi ISK_r$. Also, C adds $(CLID_r, \Psi ISK_r, \Psi IPK_r, -, -)$ in LT_{CL}. Meanwhile, C adds $(\Psi ISK_r, \Omega ISK_r)$ in LT_0 and $(\Psi IPK_r, \Omega IPK_r)$ in LT_e, and returns $(\Omega ISK_r, \Omega IPK_r)$.
- *Member secret key query* $(CLID_r, IPK_r)$: By $CLID_r$, C searches $(CLID_r, \Psi ISK_r, \Psi IPK_r, \Psi MSK_r, \Psi MPK_r)$ in LT_{CL}. If found, C returns $(\Omega MSK_r, \Omega MPK_r)$ by converting ΨMSK_r in LT_0 and ΨMPK_r in LT_e. If not found and for the i-th request of this query, C first refreshes $(\Psi SSK_{KGA,i-1,a}, \Psi SSK_{KGA,i-1,b})$ to get $(\Psi SSK_{KGA,i,a}, \Psi SSK_{KGA,i,b})$. C then picks two new variates ΨMPK_r and $\Psi\theta$ in G, and computes $\Psi MSK_r = \Psi SSK_{KGA} + \Psi MPK_r \cdot (\Psi A + \Psi\theta \cdot \Psi B)$. Also, C adds $(CLID_r, \Psi ISK_r, \Psi IPK_r, \Psi MSK_r, \Psi MPK_r)$ in LT_{CL}. Meanwhile, C adds $(\Psi MPK_r, \Omega MPK_r)$, $(\Psi\theta, \Omega\theta)$ and $(\Psi MSK_r, \Omega MSK_r)$ in LT_0, and returns $(\Omega MSK_r, \Omega MPK_r)$.
- *Member secret key leakage query* $(i, f_{KGA,i}, h_{KGA,i})$: For the i-th *Member secret key query*, A may request this *leakage* query only once. C returns $\Delta f_{KGA,i} = f_{KGA,i}(\Psi SSK_{KGA,i,a})$ and $\Delta h_{KGA,i} = h_{KGA,i}(\Psi SSK_{KGA,i,b})$.
- *Public key replacement query* $(CLID_r, (\Omega IPK'_r, \Omega MPK'_r))$: By converting $\Omega IPK'_r$ and $\Omega MPK'_r$ in LT_0, C first gets the corresponding $\Psi IPK'_r$ and $\Psi MPK'_r$. C modifies $(CLID_r, -, \Psi IPK'_r, -, \Psi MPK'_r)$ in LT_{CL}.
- *Compatible multi-signcryption* (*CMS*) *query* $(PD, SDHR, PKID_{BMC})$: For the j-th request of this query, C refreshes $(\Psi SK_{BMC,j-1,a}, \Psi SK_{BMC,j-1,b})$ to get $(\Psi SK_{BMC,j,a}, \Psi SK_{BMC,j,b})$. By $(PD, SDHR)$, C uses $(\Psi SK_{BMC,j,a}, \Psi SK_{BMC,j,b})$ to create and send back $BCS = CMS(PD, SDHR, (\Psi SK_{BMC,j,a}, \Psi SK_{BMC,j,b}))$ as follows.
  (a) Select an encrypting/decrypting key $edk \in \{0,1\}^l$ and generate an encrypted data $ED = SEF_{edk}(PD)$.
  (b) Pick two new variates ΨM and $\Psi\theta$ in G.
  (c) For $(PKID_r, PK_r)$ in *SDHR*, compute $\Psi PCK_r = \Psi M \cdot \Psi PK_r$ and $CK_r = SH_1(\Omega PCK_r)$, where ΩPCK_r is the associated bit string of ΨPCK_r.

(d) For $(CLID_r, IPK_r, MPK_r)$ in $SDHR$, compute $\Psi CCK_{r,0} = \Psi M \cdot \Psi IPK_r$, $\Psi CCK_{r,1} = \Psi M \cdot (\Psi SPK_{KGA} + \Psi MPK_r \cdot (\Psi A + \Psi\theta \cdot \Psi B))$ and $CK_r = SH_2(\Omega CCK_{r,0}, \Omega CCK_{r,1})$, where $\Omega CCK_{r,0}$ and $\Omega CCK_{r,1}$ are the associated bit strings of $\Psi CCK_{r,0}$ and $\Psi CCK_{r,1}$.

(e) According to Steps (c) and (d) above, generate $C_r = SH_3(CK_r) \| (SH_4(CK_r) \oplus edk)$, for $r = 1, 2, \ldots, n$.

(f) Pick a new variate $\Psi\rho$ in G, and compute $\Psi\sigma = \Psi SK_{BMC} + \Psi M \cdot (\Psi A + \Psi\rho \cdot \Psi B)$.

(g) Set $BCS = \langle (C_1, C_2, \ldots, C_n), \Omega M, ED, \Omega\sigma \rangle$, where ΩM and $\Omega\sigma$ are the associated bit strings of ΨM and $\Psi\sigma$.

- *Compatible multi-signcryption* (*CMS*) *leakage query* $(j, f_{BMC,j}, h_{BMC,j})$: For the j-th *Compatible multi-signcryption (CMS) query*, A may request this *leakage* query only once. C returns $\Delta f_{BMC,j} = f_{BMC,j}(\Psi SK_{BMC,j,a})$ and $\Delta h_{BMC,j} = h_{BMC,j}(\Psi SK_{BMC,j,b})$.
- *Compatible unsigncryption* (*CUS*) *query* $(PKID_r/CLID_r, BCS)$: For the k-th request of this query with $PKID_r$ or $CLID_r$, C runs the following associated procedures.
  (1) For $PKID_r$, C refreshes $(\Psi SK_{r,k-1,a}, \Psi SK_{r,k-1,b})$ to get $(\Psi SK_{r,k,a}, \Psi SK_{r,k,b})$. C then computes $\Psi PCK_r = \Psi M \cdot (\Psi Q \cdot \Psi SK_r)$ and uses $(\Psi M, \Psi PCK_r)$ to find $(\Psi M, \Psi PCK_r, edk)$ in LT_{CMS} to get $PD = SDF_{edk}(ED)$. If $\Psi Q \cdot \Psi\sigma = \Psi PK_{BMC} + \Psi Q \cdot (\Psi M \cdot (\Psi A + \Psi\rho \cdot \Psi B))$ holds, output PD and "True"; otherwise, output "invalid".
  (2) For $CLID_r$, C respectively refreshes $(\Psi ISK_{r,k-1,a}, \Psi ISK_{r,k-1,b})$ and $(\Psi MSK_{r,k-1,a}, \Psi MSK_{r,k-1,b})$ to get $(\Psi ISK_{r,k,a}, \Psi ISK_{r,k,b})$ and $(\Psi MSK_{r,k,a}, \Psi MSK_{r,k,b})$. C then computes $\Psi CCK_{r,0} = \Psi M \cdot (\Psi Q \cdot \Psi ISK_r)$, $\Psi CCK_{r,1} = \Psi M \cdot (\Psi Q \cdot \Psi SSK_{KGA} + \Psi Q \cdot \Psi MSK_r \cdot (\Psi A + \Psi\theta + \Psi B))$ and uses $(\Psi M, \Psi CCK_{r,0}, \Psi CCK_{r,1})$ to find $(\Psi M, (\Psi CCK_{r,0}, \Psi CCK_{r,1}), edk)$ in LT_{CMS} to get $PD = SDF_{edk}(ED)$. If $\Psi Q \cdot \Psi\sigma = \Psi PK_{BMC} + \Psi Q \cdot (\Psi M \cdot (\Psi A + \Psi\rho \cdot \Psi B))$ holds, output PD and "True"; otherwise, output "invalid".
- *Compatible unsigncryption* (*CUS*) *leakage query* $(k, (f_{PKIDr,k}, h_{PKIDr,k})/(f_{CLIDr,k}, h_{CLIDr,k}))$: For the k-th *Compatible unsigncryption (CUS) query* with $PKID_r$ or $CLID_r$, C runs the associated procedures, respectively. For $PKID_r$, C sends back $\Delta f_{PKIDr,k} = f_{PKIDr,k}(\Psi SK_{r,k,a})$ and $\Delta h_{PKIDr,k} = h_{PKIDr,k}(\Psi SK_{r,k,b})$. For $CLID_r$, C sends back $\Delta f_{CLIDr,k} = f_{CLIDr,k}(\Psi ISK_{r,k,a}, \Psi MSK_{r,0,a})$ and $\Delta h_{CLIDr,k} = h_{CLIDr,k}(\Psi ISK_{r,k,b}, \Psi MSK_{r,0,b})$.

– *Challenge*: A conveys a plaintext data pair (PD_1, PD_2) and $SDHR = \{[(PKID_r, PK_r) \| (CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n\}$ to C. C selects a random value $\lambda \in \{1, 2\}$ and refreshes $(\Psi SK_{BMC,j-1,a}, \Psi SK_{BMC,j-1,b})$ to get $(\Psi SK_{BMC,j,a}, \Psi SK_{BMC,j,b})$. Finally, C generates and sends back $BCS = CMS(PD_\lambda, SDHR, PKID_{BMC})$. In addition, the two conditions presented in Definition 2 must be satisfied.
– *Guess*: If A outputs $\lambda' \in \{1, 2\}$ and $\lambda' = \lambda$, it means that A wins the LRSC-EIND-CCA game and the associated advantage is $Adv(A) = |\mathrm{Pb}[\lambda' = \lambda] - 1/2|$.

As mentioned earlier, in the GBPG model, if adversaries can find collisions on G or G_e, the *discrete logarithm (DL) security assumption* on G or G_e will be broken. For computing

the collision probability, the total amount of elements and maximal polynomial degrees of LT_0 and LT_e are counted as follows. In each query in the Query, at most 3 elements are added in LT_0 or LT_e. Since A may adaptively request to C all kinds of queries at most q times, $|LT_0| + |LT_e| \leqq 3q$. For LT_0, in the *Compatible multi-signcryption (CMS) query*, $\Psi CCK_{r,1}$ has at most degree 4. Since the maximal degree of LT_0 is 4, the maximal degree of LT_e is 8 by $\Psi GE_k = \Psi G_i \cdot \Psi G_j$ of the Q_{pf} query.

In the following, let $Adv(A_{I-wo})$ be the advantage of A_I without requesting any leakage queries and $Adv(A_{I-wo}) = \text{Pb}[A_{I-wo}] + |\text{Pb}[\lambda' = \lambda] - 1/2|$, which are defined and computed as follows.

- Pb$[A_{I-wo}]$: It represents the probability of finding collisions on LT_0 or LT_e (i.e. G or G_e). For LT_0, assume that all elements $(\Psi G_i, \Omega G_i)$ consist of c kinds of variates. Thus, c random values $v_t \in Z_p^*$ (for $t = 1, 2, \ldots, c$) are randomly chosen. For any two polynomials ΨG_i and ΨG_j in LT_0, we compute $\Psi G_k = \Psi G_i - \Psi G_j$. If $\Psi G_k(v_1, v_2, \ldots, v_c) = 0$, we say that a collision in LT_0 is found. By Lemma 2, since the maximal degree of LT_0 is 4 and no partial information ($\gamma = 0$) is leaked to adversaries, the probability of $\Psi G_k(v_1, v_2, \ldots, v_c) = 0$ is at most $4/p$. Also, for LT_0, there are $\binom{|LT_0|}{2}$ pairs of $(\Psi G_i, \Psi G_j)$. Therefore, the collision probability on LT_0 is $\binom{|LT_0|}{2}(4/p)$. By similar computation, the collision probability on LT_e is $\binom{|LT_e|}{2}(8/p)$. Hence, we have

$$\text{Pb}[A_{I-wo}] = \binom{|LT_0|}{2}(4/p) + \binom{|LT_e|}{2}(8/p) \leqq (8/p)\big(|LT_0| + |LT_e|\big)^2 \leqq 128q^2/p.$$

- Pb$[\lambda' = \lambda]$: It represents the probability of $\lambda' = \lambda$ in the *Guess*. Since no partial information ($\gamma = 0$) is leaked to adversaries, we have Pb$[\lambda' = \lambda] \leqq 1/2$.

By the computations above, we have

$$Adv(A_{I-wo}) = \text{Pb}[A_{I-wo}] + \big|\text{Pb}\big[\lambda' = \lambda\big] - 1/2\big| \leqq 128q^2/p.$$

Let $Adv(A_I)$ be the advantage of the adversary A_I with requesting four leakage queries (including *Certificate leakage query*, *Member secret key leakage query*, *Compatible multi-signcryption leakage query* and *Compatible unsigncryption leakage query*) in the *Query*. By the key refreshing procedure, any two leaked partial information of a secret key are mutually independent. Thus, A_I gains at most 2γ bits of each secret key SSK_{CA}, SSK_{KGA}, SK_{BMC}, ISK_r or MSK_r. Based on $Adv(A_{I-wo})$, we have

$$Adv(A_I) \leqq Adv(A_{I-wo}) \cdot 2^{2\gamma} \leqq \big(128q^2/p\big) \cdot 2^{2\gamma} = O\big(\big(q^2/p\big) \cdot 2^{2\gamma}\big).$$

By Lemma 2, $Adv(A_I) = O((q^2/p) \cdot 2^{2\gamma})$ is negligible if $\gamma < \log p(1 - \epsilon)$. For the advantage $Adv(A_{II})$ of the adversary A_{II}, we have $Adv(A_{II}) = O((q^2/p) \cdot 2^{2\gamma})$ by similar evaluation. □

Theorem 2. *In the GBPG model, based on the DL and SHF security assumptions, the LRSC-AMRS scheme achieves the recipient anonymity in the LRSC-RIND-CCA game.*

Proof. The LRSC-RIND-CCA game is played by a PPT adversary A (A_I or A_{II}) and a challenger C, and consists of *Setup*, *Query*, *Challenge* and *Guess* as shown below.

- *Setup* and *Query* are the same as those in the proof of Theorem 1.
- *Challenge*: A conveys a plaintext data PD and $SDHR = \{[(PKID_r, PK_r)\|(CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n+1\}$ to C. C selects a random value $\lambda \in \{1, 2\}$ and sets $SDHR' = \{[(PKID_\gamma, PK_\gamma)\|(CLID_\gamma, IPK_\gamma, MPK_\gamma)], [(PKID_r, PK_r)\|(CLID_r, IPK_r, MPK_r)], r = 3, \ldots, n+1\}$. Finally, C refreshes $(\Psi SK_{BMC,j-1,a}, \Psi SK_{BMC,j-1,b})$ to get $(\Psi SK_{BMC,j,a}, \Psi SK_{BMC,j,b})$, and generates and sends back $BCS = CMS(PD, SDHR', PKID_{BMC})$. In addition, the two conditions presented in Definition 3 must be satisfied.
- *Guess*: If A outputs $\lambda' \in \{1, 2\}$ and $\lambda' = \lambda$, it means that A wins the LRSC-RIND-CCA game and the associated advantage is $Adv(A) = |\text{Pb}[\lambda' = \lambda] - 1/2|$.

By a similar evaluation in the proof of Theorem 1, we have $Adv(A_I) = O((q^2/p) \cdot 2^{2\gamma})$ and $Adv(A_{II}) = O((q^2/p) \cdot 2^{2\gamma})$. By Lemma 2, both $Adv(A_I)$ and $Adv(A_{II})$ are negligible if $\gamma < \log p(1-\epsilon)$. □

Theorem 3. *In the GBPG model, based on the DL and SHF security assumptions, the LRSC-AMRS scheme achieves the BMC authentication in the LRSC-EU-ACMA game.*

Proof. The LRSC-EU-ACMA game is played by a PPT adversary A (i.e. impersonating the BMC) and a challenger C, and consists of *Setup*, *Query* and *Forgery* as shown below.

- *Setup*: It is identical with the *Setup* phase in the proof of Theorem 1.
- *Query*: A may adaptively request to C all queries at most q times, except for the *Secret key query* ($PKID_{BMC}$) because A would like to impersonate the SMC to generate a valid broadcast ciphertext set BCS'.
- *Forgery*: A forges and sends C a broadcast ciphertext set BCS' for a plaintext data PD and $SDHR = \{[(PKID_r, PK_r)\|(CLID_r, IPK_r, MPK_r)], r = 1, 2, \ldots, n\}$. For any recipients $PKID_r$ and $CLID_r$ in $SDHR$, if they may, respectively, carry out the CUS algorithm to get and validate $PD = CUS(BCS', PKID_{BMC}, (SK_{r,k,a}, SK_{r,k,b}))$ and $PD = CUS(BCS', PKID_{BMC}, (ISK_{r,k,a}, ISK_{r,k,b}), (MSK_{r,k,a}, MSK_{r,k,b}))$, it means that A wins the LRSC-EU-ACMA game.

In the following, let $Adv(A_{wo})$ be the advantage of A without requesting any leakage queries and $Adv(A_{wo}) = \text{Pb}[A_{wo}] + \text{Pb}[\text{Valid-forging}]$, which are defined and computed as follows.

- Pb[A_{wo}]: It represents the probability of finding collisions on LT_0 or LT_e (i.e. G or G_e). By the same arguments of Pb[A_{I-wo}] in the proof of Theorem 1, we have $\text{Pb}[A_{wo}] \leqq 128q^2/p$.
- Pb[Valid-forging]: It represents the probability of forging a valid tuple $BCS' = \langle (C_1, C_2, \ldots, C_n), \Omega M', ED, \Omega\sigma' \rangle$ in the *Forgery*. C gets $\Psi M'$ and $\Psi\sigma'$ by converting $\Omega M'$ and $\Omega\sigma'$ in LT_0. Since BCS' is valid, the equality $\Psi Q \cdot \Psi\sigma' = \Psi PK_{BMC} + \Psi Q \cdot \Psi M' \cdot (\Psi A + \Psi\rho \cdot \Psi B)$ must hold. Thus, we have a multiple-variable polynomial $\Psi MP = \Psi Q \cdot \Psi\sigma' - (\Psi PK_{BMC} + \Psi Q \cdot \Psi M' \cdot (\Psi A + \Psi\rho \cdot \Psi B)) = 0$.

By Lemma 2, since ΨMP is an element of LT_e with the maximal degree 8, we have Pb[Valid-forging] $= 8/p$.

By the computations of Pb$[A_{wo}]$ and Pb[Valid-forging] above, we have

$$Adv(A_{wo}) = \text{Pb}[A_{wo}] + \text{Pb[Valid-forging]} \leqq 128q^2/p + 8/p = O\big(q^2/p\big).$$

Let $Adv(A)$ be the advantage of A with requesting two kinds of leakage queries (including *Certificate leakage query* and *Compatible multi-signcryption leakage query*) in the *Query*. By the key refreshing procedure, any two leaked partial information of a secret key are mutually independent. Thus, A gains at most 2γ bits of each secret key SSK_{CA} or SK_{BMC}. Based on $Adv(A_{wo})$, we have

$$Adv(A) \leqq Adv(A_{wo}) \cdot 2^{2\gamma} = O\big(\big(q^2/p\big) \cdot 2^{2\gamma}\big).$$

By Lemma 2, $Adv(A) = O((q^2/p) \cdot 2^{2\gamma})$ is negligible if $\gamma < \log p(1-\epsilon)$. □

6. Comparisons and Performance Analysis

Table 2 below lists the comparisons between our LRSC-AMRS scheme and some related AMRS schemes (Wang *et al.*, 2016; Tsai *et al.*, 2022; Wang *et al.*, 2012; Pang *et al.*, 2015, 2018; Li *et al.*, 2022) in terms of PKC, group, time complexity of multi-signcryption, time complexity of unsigncryption, leakage resilience and heterogeneous recipients. Two PKI-AMRS schemes in Wang *et al.* (2016), Tsai *et al.* (2022) and two ID-AMRS schemes in Wang *et al.* (2012), Pang *et al.* (2015) are implemented under the BP group. Two CL-AMRS schemes in Pang *et al.* (2018), Li *et al.* (2022) are constructed under the elliptic curve (EC) group and enjoy better performance than the constructions under the bilinear pairing (BP) group. It is worth mentioning that Tsai *et al.*'s scheme (2022) is the first AMRS with leakage resilience property. The point is that our scheme is not only suitable for multiple recipients under two heterogeneous PKCs (i.e. the PKI-PKC and the CL-PKC), but also possesses leakage resilience property.

Additionally, five schemes (Wang *et al.*, 2016, 2012; Pang *et al.*, 2015, 2018; Li *et al.*, 2022) employ the Lagrange interpolation polynomial technique to achieve anonymity between these recipients. In these schemes, a sender constructs and broadcasts an interpolation polynomial $IP(x) = \prod_{r=1}^{n}(x - CK_r) + edk (\text{mod } p) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1} + x^n$, where n denotes the number of recipients and edk is the encrypting/decrypting key. By the interpolation polynomial $IP(x)$, each authorized recipient ID_r uses her/his secret key to get $edk = IP(CK_r)$ and decrypts the plaintext data PD. Therefore, the required time complexities of multi-signcryption and unsigncryption are $O(n^2)$ and $O(n)$, respectively. By the *Compatible multi-signcryption* (*CMS*) and *Compatible unsigncryption* (*CUS*) algorithms presented in Section 4, the required time complexities are $O(n)$ and $O(1)$, respectively.

Table 2
Comparisons between our LRSC-AMRS scheme and some previously related AMRS schemes.

Schemes	PKC	Group	Time complexity of multi-signcryption	Time complexity of unsigncryption	Leakage resilience	Heterogeneous recipients
Wang *et al.*'s scheme (2016)	PKI-PKC	BP	$O(n^2)$	$O(n)$	No	No
Tsai *et al.*'s scheme (2022)	PKI-PKC	BP	$\boldsymbol{O(n)}$	$\boldsymbol{O(1)}$	**Yes**	No
Wang *et al.*'s scheme (2012)	ID-PKC	BP	$O(n^2)$	$O(n)$	No	No
Pang *et al.*'s scheme (2015)	ID-PKC	BP	$O(n^2)$	$O(n)$	No	No
Pang *et al.*'s scheme (2018)	CL-PKC	EC	$O(n^2)$	$O(n)$	No	No
Li *et al.*'s scheme (2022)	CL-PKC	EC	$O(n^2)$	$O(n)$	No	No
Our scheme	PKI-PKC CL-PKC	BP	$\boldsymbol{O(n)}$	$\boldsymbol{O(1)}$	**Yes**	**Yes**

Table 3
Computation notations and time (ms) of two time-consuming operations on a PC and a mobile device.

Notation	Meaning	Computation time on a PC	Computation time on a mobile device
T_{bp}	Bilinear pairing mapping	≈ 20.1	≈ 96.2
T_{me}	Multiplication in G or exponentiation in G_e	≈ 6.4	≈ 30.7

Table 4
The required computation costs and time (ms) for the CMS and the CUS phases.

Phase	Computational costs	$n = 10$	$n = 50$	$n = 100$
The CMS phase performed on a PC	$nT_{bp} + (3n + 4)T_{me}$	≈ 418.6	≈ 1990.6	≈ 3955.6
The CUS phase performed on a mobile device	$6T_{bp} + 3T_{me}$	≈ 669.3	≈ 669.3	≈ 669.3

In the following, we present the required computation costs and time of the proposed LRSC-AMRS scheme on a PC and a mobile device. Based on the computational simulations in Xiong and Qin (2015), Table 3 lists the computational notations and the associated computation time of two time-consuming operations on a PC and a mobile device, where the PC is equipped with a 3 GHz Pentium CPU under the MS Windows and the mobile device is equipped with a 624 MHz PXA270 CPU under a Linux system. It is worth mentioning that the adopted bilinear pairing group set has a 1024-bit RSA security level. Indeed, the required computation cost for a recipient $CLID_r$ is greater than that for a recipient $PKID_r$. Therefore, we consider the required time for multiple recipients $CLID_r$, where $r = 1, 2, \ldots, n$. Table 4 demonstrates the required computation costs and time for the compatible multi-signcryption (CMS) and the compatible unsigncryption (CUS)

phases in the proposed LRSC-AMRS scheme. By Table 4, the performance of the proposed LRSC-AMRS scheme is suitable for a PC and a mobile device.

7. Conclusions and Future Work

In this paper, the first LRSC-AMRS scheme suitable for heterogeneous PKCs (PKI-PKC and CL-PKC) has been proposed. As mentioned earlier, the LRSC-AMRS scheme must possess three security properties, namely, encryption confidentiality, recipient anonymity and sender (i.e. BMC) authentication, which have been modelled by the LRSC-EIND-CCA, LRSC-RIND-CCA and LRSC-EU-ACMA games, respectively. In the three games, adversaries (including illegal recipient and malicious KGA) are allowed to continuously acquire partial information of secret keys for multiple rounds. In the GBPG model, based on the DL and SHF security assumptions, three theorems have been shown that the LRSC-AMRS scheme achieves three security properties against adversaries. By comparing with the related schemes, the LRSC-AMRS scheme has four merits as listed below.

(1) It is the *first* LRSC-AMRS scheme suitable for heterogeneous PKCs.
(2) Multiple recipients in the LRSC-AMRS scheme can be initial recipients in the PKI-PKC, or new and upgraded recipients in the CL-PKC.
(3) Since adversaries are allowed to continuously acquire partial information of secret keys for multiple rounds, the LRSC-AMRS scheme possesses unbounded leakage resilience.
(4) The computational cost of the *unsigncryption* algorithm is constant $O(1)$.

Finally, let us point out possible future work. Due to the practical realization of quantum computers, post-quantum cryptography has attracted the attention of researchers. As far as we know, there exists no quantum-resistant AMRS scheme. Therefore, to propose a post-quantum AMRS scheme is an interesting topic. Furthermore, it is more practical to propose a post-quantum AMRS scheme suitable for recipients in heterogeneous PKCs.

Acknowledgements

The authors would like to appreciate anonymous reviewers for their valuable comments and suggestions on this paper that have resulted in the improvement of quality, completeness and readability. This research was partially supported by National Science and Technology Council, Taiwan, under contract no. NSTC113-2221-E-018-024-MY2.

References

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

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

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

Dong, C., Zhang, J. (2024). On the security of multi-receiver certificateless generalized signcryption scheme for WBANs. *IEEE Transactions on Dependable and Secure Computing*, 21(4), 4302–4303.

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4), 469–472.

Galindo, D., Vivek, 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.

Ho, T.-C., Tseng, Y.-M., Huang, S.-S. (2024). Leakage-resilient hybrid signcryption in heterogeneous public-key systems. *Informatica*, 35(1), 131–154.

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

Kim, T., Jang, J., Jeon, G., Kim, J. (2024). Investigating driver preferences for traffic information using digital signage and road surface holograms. *KSCE Journal of Civil Engineering*, 28, 1475–1488.

Lal, S., Kushwah, P. (2009). Anonymous ID-based signcryption scheme for multiple receivers. IACR Cryptology ePrint Archive, Article-ID 345.

Li, H., Wu, C., Pang, L. (2022). Completely anonymous certificateless multi-receiver signcryption scheme with sender traceability. *Journal of Information Security and Applications*, 71, 103384.

Li, X., Gong, Y., Huang, K., Niu, Z. (2023). Over-the-air integrated sensing, communication, and computation in IoT networks. *IEEE Wireless Communications*, 30(1), 32–38.

Miller, V.S. (1985). Use of elliptic curves in cryptography. In: *Advances in Cryptology – CRYPTO'85*, *LNCS*, Vol. 218, pp. 417–426.

Park, Y., Zhang, Y. (2022). Technology readiness and technology paradox of unmanned convenience store users. *Journal of Retailing and Consumer Services*, 65, 102523.

Pang, L., Gao, L., Li, H., Wang, Y. (2015). Anonymous multi-receiver ID-based signcryption scheme. *IET information Security*, 9(3), 194–201.

Pang, L., Kou, M., Wei, M., Li, H. (2018). Efficient anonymous certificateless multi-receiver signcryption scheme without bilinear pairings. *IEEE Access*, 6, 78123–78135.

Pang, L., Kou, M., Wei, M., Li, H. (2019). Anonymous certificateless multi-receiver signcryption scheme without secure channel. *IEEE Access*, 7, 84091–84106.

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

Shamir, A. (1984). Identity-based cryptosystems and signature schemes. In: *Advances in Cryptology – CRYPTO'84*, *LNCS*, Vol. 196, pp. 47–53.

Shen, J., Gui, Z., Chen, X., Zhang, J., Xiang, Y. (2022). Lightweight and certificateless multi-receiver secure data transmission protocol for wireless body area networks. *IEEE Transactions on Dependable and Secure Computing*, 19(3), 1464–1475.

Tsai, T.-T., Tseng, Y.-M., Huang, S.-S., Xie, J.-Y., Hung, Y.-H. (2022). Leakage-resilient anonymous multi-recipient signcryption under a continual leakage model. *IEEE Access*, 10, 104636–104648.

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.

Tseng, Y.-M., Ho, T.-C., Tsai, T.-T., Huang, S.-S. (2024). AHMRE-SCST: lightweight anonymous heterogeneous multi-recipient encryption with seamlessly compatible system transformation for IoT devices. *IEEE Internet of Things Journal*, 11(17), 28508–28525.

Wang, H., Zhang, Y., Qin, B. (2012). Analysis and improvements of two identity based anonymous signcryption schemes for multiple receivers. In: *Proceedings of IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1057–1062.

Wang, Q., He, M., Zheng, X. (2016). Privacy-preserving communication for vehicular with multi-receiver conditionally anonymous ring signcryption. In: *Proceedings of 3rd International Conference on Materials Engineering, Manufacturing Technology and Control*, pp. 496–501.

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.

Zhang, B., Xu, Q. (2010). An ID-based anonymous signcryption scheme for multiple receivers secure in the standard model. In: *Proceedings of Advances in Computer Science and Information Technology*, *LNCS*, Vol. 6059, pp. 15–27.

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.-C. Ho is currently pursuing the PhD degree with the Department of Mathematics, National Changhua University of Education, Changhua, Taiwan. Her research interests include applied cryptography, information security and leakage-resilience cryptography.

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