IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 22, 2019, accepted February 8, 2019, date of publication February 19, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900072

Efficient and Anonymous Certificateless Multi-Message and Multi-Receiver Signcryption Scheme Based on ECC

LIAOJUN PANG[1,2], (Member, IEEE), MENGMENG WEI[1], AND HUIXIAN LI[3]

[1]State Key Laboratory of Integrated Services Networks, School of Life Science and Technology, Xidian University, Xi'an 710071, China
[2]Department of Computer Science, Wayne State University, Detroit, MI 48202, USA
[3]School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding authors: Liaojun Pang (liaojun.pang@wayne.edu) and Huixian Li (lihuixian@nwpu.edu.cn)

This work was supported in part by the National Key Technologies Research and Development Program of China under Grant 2018YFB1105303, in part by the Natural Science Foundation of China under Grant 61473214 and Grant 61103178, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM6064, and in part by the National Cryptography Development Fund under Grant MMJJ20170208.

ABSTRACT As the further extension of the multi-receiver signcryption, the multi-message and multi-receiver signcryption allows a sender to simultaneously signcrypt different messages for different receivers in only one logic operation, which makes it more flexible than the traditional multi-receiver signcryption in which only the same and unique message can be sent to all authorized receivers. The existing multi-message and multi-receiver signcryption schemes are constructed based on either the identity-based cryptography or the public key infrastructure-based cryptography, and thus, they have to suffer from the key escrow problem inherent in the identity-based cryptography or the public key certificate management burden related with the public key infrastructure-based cryptography. Certificateless public key cryptography provides an idea to solve the key escrow problem and eliminate the public key certificate management burden and has been applied to many cryptographic algorithms. In this paper, to avoid the above problems in the existing multi-message and multi-receiver signcryption schemes, the concept of the certificateless public key cryptography was introduced into the designing of the multi-message and multi-receiver signcryption, and a certificateless multi-message and multi-receiver signcryption scheme was proposed. The proposed scheme is free from the key escrow problem and the public key certificate management burden because it is constructed based on the certificateless public key cryptography. Moreover, compared with the existing schemes, it is improved in efficiency because it does not use the bilinear pairing operations but utilizes the limited number of scalar point multiplication on elliptic curve cryptography operations. At the same time, the proposed scheme achieves receiver anonymity.

INDEX TERMS Certificateless public key cryptography, elliptic curve cryptography, multi-message and multi-receiver signcryption, receiver anonymity.

I. INTRODUCTION

With the development of the Internet and communication technologies, multicast services have become more and more popular in our daily life, such as mobile crowdsensing [1] and cloud computing [2], [3]. Multi-receiver encryption/signcryption [4], widely considered as one of most promising and efficient solutions to one-to-many secure communication, enables the sender to securely send the same message to multiple receivers synchronously by only one logic operation, and only authorized receivers can decrypt the message independently while others cannot. Multi-receiver encryption/signcryption has been applied to paid-TV system [5], IOT [6] and smart grid [7].

However, in recent years, the multicast communication environment in reality has become more and more complex [8]. In a practical application environment, it often happens that different services need to be provided to different customers, and in this case, the multi-message and multi-receiver signcryption [9], as the further extension of

The associate editor coordinating the review of this manuscript and approving it for publication was Zhitao Guan.

the multi-receiver signcryption [10], comes into being. In a multi-receiver signcryption scheme, the sender is enabled to send the unique and same message to all authorized receivers, while in a multi-message and multi-receiver signcryption scheme, the sender is enabled to send multiple and different messages to different receivers in one logic operation, and each authorized receiver can designcrypt out his/her own plaintext message without exposing the plaintext messages of others. Nowadays, the research on the multi-message and multi-receiver signcryption has become a new hotspot in the field of information security. In this paper, our attention is mainly paid to the multi-message and multi-receiver signcryption scheme, and thus in the following, we shall mainly describe its motivation, development and shortcomings.

The idea of the multi-message and multi-receiver signcryption was firstly presented by Seo and Kim [9]. Seo and Kim proposed a domain-verifiable multi-message and multi-receiver signcryption scheme which is applied to the electronic funds transfer protocol, and in their scheme, each predetermined participant could decrypt out his/her own corresponding plaintext message and verify the whole transaction within the domain. Later, Dalia [11] proposed a chaotic public key multi-message and multi-receiver signcryption scheme. In Dalia's scheme, the strength and security of the algorithm is increased by designing the chaotic multi-key generator to generate chaotic keys for both the block cipher and keyed hash algorithms. Unfortunately, in Dalia's scheme, each authorized receiver not only could obtain his/her own plaintext message by decrypting the ciphertext, but also could obtain other authorized receivers' plaintext messages, which is not what the multi-message and multi-receiver signcryption wants. In 2008, Hassan and Esam [12] proposed a new multi-message and multi-receiver signcryption scheme. Hassan and Esam's scheme [12] is designed based on the elliptic curve cryptography (ECC) [13] and is reduced in the computation cost, but it has the same problem as Dalia's scheme [11], that is, it enables the authorized receiver to obtain other authorized receivers' plaintext messages.

To ensure the security of the group communication, Han and Gui [14] proposed a novel multi-message and multi-receiver signcryption scheme. Different from Dalia's scheme and Hassan and Esam's scheme, Han and Gui's scheme restricts the ability of the authorized receiver, which enables each authorized receiver to obtain only his/her own plaintext message by decrypting the ciphertext without exposing others' plaintext messages. Soon afterwards, based on the designing idea similar to their previous scheme, Han *et al.* proposed two other multi-message and multi-receiver signcryption schemes [15], [16]. Han *et al.*'s first scheme [15] provides an adaptive secure multicast framework based on the multi-message and multi-receiver signcryption in wireless networks, and their another scheme [16] provides a method to reduce multicast networks computational overheads by the parallel algorithm. Although Han *et al.*'s schemes [14]–[16] meet the requirements of the multi-message and multi-receiver signcryption, they all use the time-consuming operations such as the bilinear pairing operations or the modular exponentiation operations [17], and thus they are low in efficiency.

In 2013, Kumar and Ansari [18] proposed a multi-message and multi-receiver signcryption based on the chaos with public verifiability. In Kumar and Ansari's scheme, the public verifiability [19] is achieved, which means that any third-party can verify whether the ciphertext is valid or not without the knowledge of the sender's or receiver's private key, and at the same time, the use of the chaos makes the security of their algorithm enhanced. Unfortunately, Kumar *et al.*'s scheme is as inefficient as Han *et al.*'s schemes due to the fact that it also utilizes modular exponentiation operations. In 2015, Nizamud Din *et al.* [20] proposed an efficient multi-message and multi-receiver signcryption scheme. Nizamud Din *et al.*'s scheme is constructed based on scalar point multiplication on ECC operations instead of the complex bilinear pairing operations, and it is improved in efficiency largely. Also for the improvement on the computation efficiency, Rahman *et al.* [21] proposed a lightweight multi-message and multi-receiver signcryption scheme in 2018. Rahman *et al.* improve the efficiency of their scheme by using divisor multiplication on hyper elliptic curve operations, and analyses show that their scheme is more efficient than Nizamud *et al.*'s scheme.

Nevertheless, it worth noting that the existing multi-message and multi-receiver signcryption schemes [9], [11], [12], [14]–[16], [18], [20], [21] mentioned above are mainly constructed based on the public key infrastructure (PKI)-based cryptography [22], which means that they should suffer from the public key certificate management burden related with the PKI-based cryptography and need an expensive cost to maintain PKI for application systems, which is not practical for small-scale and temporary applications. To avoid the public key certificate management burden, Qiu *et al.* [23] proposed an identity-based multi-message and multi-receiver signcryption scheme in 2016. In Qiu *et al.*'s scheme, identity-based cryptography (IBC) [24], in which the user's public key is related to his/her own identity and maintaining PKI is not required, is introduced into the designing of the multi-message and multi-receiver signcryption scheme so as to avoid its public key certificate management burden. Subsequently, Wang *et al.* [25] proposed another multi-message and multi-receiver signcryption scheme for ad-hoc networks. Their scheme makes use of the heterogeneous system to shift between the PKI-based cryptography and IBC, and thus it achieves two-way signcryption.

Regretfully, in Wang *et al.*'s scheme and Qiu *et al.*'s scheme, the utilization of IBC causes the inevitable key escrow problem [26] related with IBC, that is to say, the user's complete private key can be obtained by key generation center (KGC) and malicious KGC attacks cannot be prevented. In 2017, Niu *et al.* proposed two heterogeneous multi-message and multi-receiver signcryption schemes [27], [28], successively. Niu *et al.*'s first scheme [27] can shift from certificateless public key cryptography (CLC-PKC) [29]

to IBC, and their another scheme [28] can shift from IBC to CLC-PKC. Niu *et al.*'s two schemes all use CLC-PKC [29], in which the key escrow problem in IBC is solved because the user' private key is generated by the user and KGC and it cannot be obtained by KGC, and at the same time the public key certificate management burden in the PKI-based cryptography is non-existent as a result of the inheritance of IBC's advantage. Unfortunately, both of their two schemes still suffer from the key escrow problem, which results from the fact that their schemes applied IBC. Moreover, it is worth noting that multi-message and multi-receiver signcryption schemes [23], [25], [27], [28] employ the time-consuming bilinear pairing operations, which makes them low in efficiency.

As we know, the receiver anonymity [30] is a very important security attribute in multi-receiver encryption/signcryption, and thus, when we design the multi-message and multi-receiver signcryption scheme, known as a branch of the multi-receiver signcryption scheme, the receiver anonymity should also be taken into consideration. The receiver anonymity [31] means that each user can judge that whether he/she is an authorized receiver or not but cannot judge whether other users are authorized or not. However, the existing multi-message and multi-receiver signcryption schemes [9], [11], [12], [14]–[16], [18], [20], [21], [23], [25] mentioned above do not achieve the receiver anonymity and leak the receivers' privacy [32] more or less. Niu *et al.* [27], [28] have taken the receivers' privacy protection into account during the course of designing their schemes. Nevertheless, analyses show that Niu *et al.*'s scheme [27] cannot truly achieve the receiver anonymity as they expected due to the inherent structure of Lagrange interpolation polynomial [33]. Although Niu *et al.*'s another scheme [28] achieves the receiver anonymity, it is still subjected to the key escrow problem resulted from the used IBC cryptography.

Through the above analyses, it can be seen that the research on the multi-message and multi-receiver signcryption scheme has become a new hotspot in the field of information security. However, the existing multi-message and multi-receiver signcryption schemes suffer from either the public key certificate management burden related with the PKI-based cryptography or the key escrow problem inherent in IBC, because they are constructed based on either the PKI-based cryptography or IBC. Moreover, some schemes are not ideal in efficiency, and some do not even protect the receivers' privacy. Motivated by these concerns, we introduce the concept of CLC-PKC into the designing of the multi-message and multi-receiver signcryption and propose an efficient and anonymous certificateless multi-message and multi-receiver signcryption scheme based on ECC. Our scheme is constructed based on CLC-PKC, and hence it is free from the public key certificate management burden and the key escrow problem. Besides, it is designed by utilizing the limited number of scalar point multiplication on ECC operations, which makes it efficient in computation. At the same time, it achieves the receiver anonymity and protects the receivers' privacy. Compared with the existing multi-message and multi-receiver signcryption schemes, our scheme has better performance in regardless of functions or efficiency.

TABLE 1. Notations.

Name	Meaning
CLC-PKC	Certificateless public key cryptography
ECC	Elliptic curve cryptography
IBC	Identity (ID)-based cryptography
KGC	Key generation center
G_p	The addition cycle group of points on ECC
PKI	Public key infrastructure
p	Large prime number
PK_i	Public key of the user i, i represents the user's identity
P	Generator of G_p
Pr	The probability of an event
SK_i	Private key of the user i, i represents the user's identity
Z_p^*	Non-zero multiplicative group with large prime p

The rest of this paper is organized as follows: the preliminaries are introduced in Section 2 and the proposed scheme is elaborated in Section 3. In Section 4, we prove the correctness and security of the proposed scheme. A comparison is made between the proposed scheme and the existing ones in terms of functions and efficiency in Section 5. Section 6 makes a summary of the full paper.

In order to facilitate understanding, notations used in this paper are listed in TABLE 1.

II. PRELIMINARIES

In this section, we will present computational problems, algorithm models and security models used in the proposed scheme.

A. COMPUTATIONAL PROBLEMS

Define that p is a large prime number, G_p with its generator P is an addition cycle group of points on ECC, and Z_p^* is a non-zero multiplicative group. The Elliptic Curve Discrete Logarithm problem (ECDLP) and Computational Diffie-Hellman problem (CDHP) are shown as follows:

1) ECDLP

With a set of given elements $\langle P, aP\rangle \in G_p$, calculating a is called the ECDLP, where $a \in Z_p^*$.

Definition 1: The probability advantage that the ECDLP is solved in a probabilistic polynomial time (PPT) algorithm Π is defined as

$$Adv^{ECDLP} = Pr[a \in Z_p^* | \Pi(P, aP) = a].$$

ECDLP Assumption: It is hard to solve the ECDLP in any PPT algorithm, and thus we assume that Adv^{ECDLP} is negligible.

2) CDHP

With a set of given elements $\langle P, aP, bP\rangle \in G_p$, calculating abP is called the CDHP, where $a, b \in Z_p^*$.

Definition 2: The probability advantage that the CDHP is solved in a PPT algorithm Π is defined as

$$Adv^{CDHP} = Pr[a, b \in Z_p^* | \Pi(P, aP, bP) = abP].$$

CDHP Assumption: It is hard to solve the CDHP in any PPT algorithm, and thus we assume that Adv^{CDHP} is negligible.

B. ALGORITHM MODELS

Definition 3: The algorithm models of the proposed scheme consist of *Setup algorithm*, *Set secret value algorithm*, *Extract partial private key algorithm*, *Set public key algorithm*, *Set private key algorithm*, *Signcryption algorithm*, and *Designcryption algorithm*, shown as follows:

Setup Algorithm: With the security parameter λ as input, KGC executes the algorithm to generate the system master key s and the public parameters *params*. Then, KGC keeps s secret and makes *params* public.

Set Secret Value Algorithm: With the public parameters *params* and the user's identity ID_U as input, the user executes the algorithm to generate his/her own secret value d_U and the corresponding secret value parameter D_U.

Extract Partial Private Key Algorithm: With the user's identity ID_U, the user's secret value parameter D_U, the system master key s and the public parameters *params* as input, KGC executes the algorithm to generate the user's partial private key v_U and the user's partial public key T_U.

Set Public Key Algorithm: With the user's identity ID_U, the user's secret value parameter D_U, the user's partial public key T_U and the public parameters *params* as input, the user executes the algorithm to generate his/her own public key PK_U.

Set Private Key Algorithm: With the user's identity ID_U, the user's secret value d_U, the user's partial private key v_U, the user's public key PK_U and the public parameters *params* as input, the user executes the algorithm to generate his/her own private key SK_U.

Signcryption Algorithm: With the sender's identity ID_S, receivers' identities ID_i $(1 \leq i \leq n)$, the plaintext message set $M = \{m_1, m_2, \ldots, m_n\}$, and the public parameters *params* as input, the sender executes the algorithm to generate the signcryption ciphertext σ.

Designcryption Algorithm: With the receiver' public key PK_i, the signcryption ciphertext σ, and the public parameters *params* as input, the receiver executes the algorithm to generate his/her own plaintext message m_i.

C. SECURITY MODELS

The security models of the proposed scheme are made up of message confidentiality, receiver anonymity and unforgeability. There are two types of attackers in every security model [29]. In the first two security models, two types of attackers are called the adversary $\mathcal{A}_I$ and the adversary $\mathcal{A}_{II}$ [30], [34] respectively. In the third security model, two types of attackers are called the forger $\mathcal{F}_I$ and the forger $\mathcal{F}_{II}$ [34], respectively. $\mathcal{A}_I/\mathcal{F}_I$ can replace the user's public key arbitrarily but cannot get the system master key, while $\mathcal{A}_{II}/\mathcal{F}_{II}$ knows the system master key but cannot replace the user's public key.

1) MESSAGE CONFIDENTIALITY

Message confidentiality refers to the fact that attackers within their own attacks have no ability to successfully decrypt out the plaintext message. Referring to the model of Selvi *et al.*'s scheme [34], the message confidentiality model of the proposed scheme can be defined as the indistinguishability of certificateless multi-message and multi-receiver signcryption under selective multi-ID, chosen ciphertext attack (IND-CLMMRS-CCA). The following ***Game 1*** and ***Game 2*** are defined to meet IND-CLMMRS-CCA against $\mathcal{A}_I$ and $\mathcal{A}_{II}$, respectively.

Game 1: The game is an interaction between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}_I$ under IND-CLMMRS-CCA. Define Π as a certificateless multi-message and multi-receiver anonymous signcryption algorithm, and the specific game interaction is shown as follows:

Setup: $\mathcal{C}$ executes the algorithm to generate the system master key s and the public parameters *params*. Then, $\mathcal{C}$ sends *params* to $\mathcal{A}_I$ and keeps s secret.

Phase 1: Receiving *params* from $\mathcal{C}$, $\mathcal{A}_I$ outputs n target identities $L = \{ID_1, ID_2, \ldots, ID_n\}$, and sends them to $\mathcal{C}$.

Phase 2: $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of the following queries, and $\mathcal{C}$ makes according responds:

Set Secret Value Query: $\mathcal{A}_I$ queries $\mathcal{C}$ for the secret value of the identity ID_j. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ runs the *Set secret value algorithm* to obtain the secret value d_j, and returns it to $\mathcal{A}_I$.

Extract Partial Private Key Query: $\mathcal{A}_I$ queries $\mathcal{C}$ for the partial private key of the identity ID_j. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ runs the *Extract partial private key algorithm* to obtain the partial private key v_j, and returns it to $\mathcal{A}_I$.

Set Public Key Query: $\mathcal{A}_I$ queries $\mathcal{C}$ for the public key of the identity ID_j. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ runs the *Set public key algorithm* to obtain the public key PK_j, and returns it to $\mathcal{A}_I$.

Set Private Key Query: $\mathcal{A}_I$ queries $\mathcal{C}$ for the private key of the identity ID_j. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ runs the *Set private key algorithm* to obtain the private key SK_j, and returns it to $\mathcal{A}_I$.

Public key replacement query: With the public key PK'_j, $\mathcal{A}_I$ requests $\mathcal{C}$ for the public key replacement of the identity ID_j. Receiving the request from $\mathcal{A}_I$, $\mathcal{C}$ replaces the public key PK_j with PK'_j.

Signcryption query: With receivers' identities $L^* = \{ID_1, ID_2, \ldots, ID_n\}$ and the sender's identity ID_S, $\mathcal{A}_I$ queries $\mathcal{C}$ for the signcryption of the plaintext message set $M = \{m_1, m_2, \ldots, m_n\}$. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ runs the *Signcryption algorithm* to obtain the signcryption ciphertext $\sigma \leftarrow Signcryption(params, M, L^*, ID_S)$, and returns it to $\mathcal{A}_I$.

Designcryption query: With receivers' identities $L^* = \{ID_1, ID_2, \ldots, ID_n\}$, $\mathcal{A}_I$ queries $\mathcal{C}$ for the designcryption of the signcryption ciphertext σ. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ runs the *Designcryption algorithm*to obtain the plaintext message $m_i \leftarrow Designcryption(params, \sigma, L^*)$, and returns it to $\mathcal{A}_I$.

Challenge: $\mathcal{A}_I$ chooses the sender's identity ID_S, selects two plaintext message sets $M_0 = \{m_1^0, m_2^0, \ldots, m_n^0\}$ and $M_1 = \{m_1^1, m_2^1, \ldots, m_n^1\}$, where $|m_i^0| = |m_i^1|$ $(1 \le i \le n)$, and sends two plaintext message sets and the sender's identity ID_S to $\mathcal{C}$. Receiving $\{M_0, M_1, ID_S\}$ from $\mathcal{A}_I$, $\mathcal{C}$ randomly chooses a bit $\mu \in \{0, 1\}$, calculates the signcryption ciphertext $\sigma^* \leftarrow$ *Signcryption*$(params, M_\mu, L, ID_S)$, and returns σ^* to $\mathcal{A}_I$.

Phase 3: $\mathcal{A}_I$ asks $\mathcal{C}$ for the same queries as Phase 2. However, there are the following restrictions:

1) $\mathcal{A}_I$ cannot query for the partial private key of any target identity in L.

2) A_I cannot query for the private key of the target identity whose public key has been replaced.

3) $\mathcal{A}_I$ cannot query for the designcryption of the signcryption ciphertext σ^*.

Guess: According to the phases performed by $\mathcal{A}_I$ and $\mathcal{C}$ above, $\mathcal{A}_I$ outputs a bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, $\mathcal{A}_I$ wins the game. Otherwise, $\mathcal{A}_I$ fails. The probability advantage that $\mathcal{A}_I$ wins the game is

$$Adv_{\Pi}^{IND-CLMMRS-CCA}(\mathcal{A}_I) = \left|\Pr\left[\mu' = \mu\right] - \frac{1}{2}\right|.$$

Definition 4: If for $\mathcal{A}_I$ under IND-CLMMRS-CCA, its probability advantage of winning ***Game 1*** meets $Adv_{\Pi}^{IND-CLMMRS-CCA}(\mathcal{A}_I) \le \varepsilon$ within PPT t, the algorithm Π is said to meet (t, ε)-IND-CLMMRS-CCA-$\mathcal{A}_I$ security, where ε is the non-negligible probability advantage.

Game 2: The game is an interaction between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}_{II}$ under IND-CLMMRS-CCA. Define Π as a certificateless multi-message and multi-receiver anonymous signcryption algorithm, and the specific game interaction is shown as follows:

Setup: $\mathcal{C}$ executes the algorithm to generate the system master key s and the public parameters *params*. Then, $\mathcal{C}$ sends s and *params* to $\mathcal{A}_{II}$.

Phase 1: Receiving s and *params* from $\mathcal{C}$, $\mathcal{A}_{II}$ outputs n target identities $L = \{ID_1, ID_2, \ldots, ID_n\}$ and sends them to $\mathcal{C}$.

Phase 2: $\mathcal{A}_{II}$ asks $\mathcal{C}$ for the same queries as Phase 2 in ***Game 1***, and $\mathcal{C}$ makes according responds.

Challenge: $\mathcal{A}_{II}$ chooses the sender's identity ID_S, selects two plaintext message sets $M_0 = \{m_1^0, m_2^0, \ldots, m_n^0\}$ and $M_1 = \{m_1^1, m_2^1, \ldots, m_n^1\}$, where $|m_i^0| = |m_i^1|$ $(1 \le i \le n)$, and sends two plaintext message sets and the sender's identity ID_S to $\mathcal{C}$. Receiving $\{M_0, M_1, ID_S\}$ from $\mathcal{A}_{II}$, $\mathcal{C}$ randomly chooses a bit $\mu \in \{0, 1\}$, calculates the signcryption ciphertext $\sigma^* \leftarrow$ *Signcryption*$(params, M_\mu, L, ID_S)$, and returns σ^* to $\mathcal{A}_{II}$.

Phase 3: $\mathcal{A}_{II}$ asks $\mathcal{C}$ for the same queries as Phase 2. However, there are the following restrictions:

1) $\mathcal{A}_{II}$ cannot query for the secret value of any target identity in L.

2) A_{II} cannot query for the private key of the target identity whose public key has been replaced.

3) $\mathcal{A}_{II}$ cannot query for the designcryption of the signcryption ciphertext σ^*.

Guess: According to the phases performed by $\mathcal{A}_{II}$ and $\mathcal{C}$ above, $\mathcal{A}_{II}$ outputs a bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, $\mathcal{A}_{II}$ wins the game. Otherwise, $\mathcal{A}_{II}$ fails. The probability advantage that $\mathcal{A}_{II}$ wins the game is

$$Adv_{\Pi}^{IND-CLMMRS-CCA}(\mathcal{A}_{II}) = \left|\Pr\left[\mu' = \mu\right] - \frac{1}{2}\right|$$

.

Definition 5: If for $\mathcal{A}_{II}$ under IND-CLMMRS-CCA, its probability advantage of winning ***Game 2*** meets $Adv_{\Pi}^{IND-CLMMRS-CCA}(\mathcal{A}_{II}) \le \varepsilon$ within PPT t, the algorithm Π is said to meet (t, ε)-IND-CLMMRS-CCA-$\mathcal{A}_{II}$ security, where ε is the non-negligible probability advantage.

2) RECEIVER ANONYMITY

Receiver anonymity refers to the fact that attackers within their own attacks have no ability to successfully obtain authorized receivers' identities. Referring to the model of Islam *et al.*'s scheme [30], the receiver anonymity model of the proposed scheme can be defined as the anonymous indistinguishability of certificateless multi-message and multi-receiver signcryption under selective multi-ID, chosen ciphertext attack (ANON-IND-CLMMRS-CCA). The following ***Game 3*** and ***Game 4*** are defined to meet ANON-IND-CLMMRS-CCA against $\mathcal{A}_I$ and $\mathcal{A}_{II}$, respectively.

Game 3: The game is an interaction between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}_I$ under ANON-IND-CLMMRS-CCA. Define Π as a certificateless multi-message and multi-receiver anonymous signcryption algorithm, and the specific game interaction is shown as follows:

Setup: The step is the same as Setup in ***Game 1***.

Phase 1: Receiving *params*from $\mathcal{C}$, $\mathcal{A}_I$ outputs two target identities $L = \{ID_0, ID_1\}$ and sends them to $\mathcal{C}$.

Phase 2: $\mathcal{A}_I$ asks $\mathcal{C}$ for the same queries as Phase 2 in ***Game 1***, and $\mathcal{C}$ makes according responds.

Challenge: $\mathcal{A}_I$ chooses a plaintext message set $M = \{m_1, m_2, \ldots, m_n\}$, a group of receivers' identities $L^* = \{ID_2, ID_3, \ldots, ID_n\}$, and the sender's identity ID_S. Then, $\mathcal{A}_I$ sends the

plaintext message set M, receivers' identities L^* and the sender's identity ID_S to $\mathcal{C}$. Receiving $\{M, ID_S, L^*\}$ from $\mathcal{A}_I$, $\mathcal{C}$ randomly chooses a bit $\mu \in \{0, 1\}$, calculates the signcryption ciphertext $\sigma^* \leftarrow$*Signcryption*$(params, M, ID_\mu, L^*, ID_S)$, and returns σ^* to $\mathcal{A}_I$.

Phase 3: The step is the same as Phase 3 in ***Game 1***.

Guess: According to the phases performed by $\mathcal{A}_I$ and $\mathcal{C}$ above, $\mathcal{A}_I$ outputs a bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, $\mathcal{A}_I$ wins the game. Otherwise, $\mathcal{A}_I$ fails. The probability advantage that $\mathcal{A}_I$ wins the game is

$$Adv_{\Pi}^{ANON-IND-CLMMRS-CCA}(\mathcal{A}_I) = \left|\Pr\left[\mu' = \mu\right] - \frac{1}{2}\right|$$

Definition 6: If for $\mathcal{A}_I$ under ANON-IND-CLMMRS-CCA, its probability advantage of winning ***Game 3*** meets $Adv_{\Pi}^{ANON-IND-CLMMRS-CCA}(\mathcal{A}_I) \le \varepsilon$ within PPT t, the

algorithm Π is said to meet (t, ε)-ANON-IND-CLMMRS-CCA-$\mathcal{A}_I$ security, where ε is the non-negligible probability advantage.

Game 4: The game is an interaction between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}_{II}$ under ANON-IND-CLMMRS-CCA. Define Π as a certificateless multi-message and multi-receiver anonymous signcryption algorithm, and the specific game interaction is shown as follows:

Setup: The step is the same as Setup in ***Game 2***.

Phase 1: Receiving s and *params* from $\mathcal{C}$, $\mathcal{A}_{II}$ outputs two target identities $L = \{ID_0, ID_1\}$ and sends them to $\mathcal{C}$.

Phase 2: $\mathcal{A}_{II}$ asks $\mathcal{C}$ for the same queries as Phase 2 in ***Game 2***, and $\mathcal{C}$ makes according responds.

Challenge: $\mathcal{A}_{II}$ chooses a plaintext message set $M = \{m_1, m_2, \ldots, m_n\}$, a group of receivers' identities $L^* = \{ID_2, ID_3, \ldots, ID_n\}$, and the sender's identity ID_S. Then, $\mathcal{A}_{II}$ sends the plaintext message set M, receivers' identities L^* and the sender's identity ID_S to $\mathcal{C}$. Receiving $\{M, ID_S, L^*\}$ from $\mathcal{A}_{II}$, $\mathcal{C}$ randomly chooses a bit $\mu \in \{0, 1\}$, calculates the signcryption ciphertext $\sigma^* \leftarrow Signcryption(params, M, ID_\mu, L^*, ID_S)$, and returns σ^* to $\mathcal{A}_{II}$.

Phase 3: The step is the same as Phase 3 in ***Game 2***.

Guess: According to the phases performed by $\mathcal{A}_{II}$ and $\mathcal{C}$ above, $\mathcal{A}_{II}$ outputs a bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, $\mathcal{A}_{II}$ wins the game. Otherwise, $\mathcal{A}_{II}$ fails. The probability advantage that $\mathcal{A}_{II}$ wins the game is

$$Adv_{\Pi}^{ANON-IND-CLMMRS-CCA}(\mathrm{A}_{II}) = \left| \Pr\left[\mu' = \mu\right] - \frac{1}{2} \right|.$$

Definition 7: If for $\mathcal{A}_{II}$ under ANON-IND-CLMMRS-CCA, its probability advantage of winning ***Game 4*** meets $Adv_{\prod}^{ANON-IND-CLMMRS-CCA}(\mathrm{A}_{II}) \leq \varepsilon$ within PPT t, the algorithm Π is said to meet (t, ε)-ANON-IND-CLMMRS-CCA-$\mathcal{A}_{II}$ security, where ε is the non-negligible probability advantage.

3) UNFORGEABILITY

Unforgeability refers to the fact that attackers within their own attacks have no ability to successfully forge the sender's signature. Referring to the model of Selvi *et al.*'s scheme [34], the unforgeability model of the proposed scheme can be defined as the strong existential unforgeability of certificateless multi-message and multi-receiver signcryption under selective multi-ID, chosen plaintext attack (SUF-CLMMRS-CPA). The following ***Game 5*** and ***Game 6*** are defined to meet SUF-CLMMRS-CPA against $\mathcal{F}_I$ and $\mathcal{F}_{II}$, respectively.

Game 5: The game is an interaction between the challenger $\mathcal{C}$ and the forger $\mathcal{F}_I$ under SUF-CLMMRS-CPA. Define Π as a certificateless multi-message and multi-receiver anonymous signcryption algorithm, and the specific game interaction is shown as follows:

Setup: The step is the same as Setup in ***Game 1***.

Phase 1: Receiving *params* from $\mathcal{C}$, $\mathcal{F}_I$ outputs n target identities $L = \{ID_1, ID_2, \ldots, ID_n\}$ and sends them to $\mathcal{C}$.

Attack: $\mathcal{F}_I$ asks C for the same queries as Phase 2 in ***Game 1***, and $\mathcal{C}$ makes according responds.

Forgery: $\mathcal{F}_I$ outputs the forged signcryption ciphertext σ^* and a group of receivers' identities $L^* = \{ID_1, ID_2, \ldots, ID_n\}$. If the signcryption ciphertext σ^* can be decrypted and verified correctly by any receiver in L^*, $\mathcal{F}_I$ wins the game. Otherwise, $\mathcal{F}_I$ fails. However, it is worth noting that σ^* is not generated by *Signcryption query*, and other restrictions are the same as Phase 3 in ***Game 1***.

Definition 8: If for $\mathcal{F}_I$ under SUF-CLMMRS-CPA, its probability advantage of winning ***Game 5*** meets $Adv_{\prod}^{SUF-CLMMRS-CPA}(\mathrm{F}_I) \leq \varepsilon$ within PPT t, the algorithm Π is said to meet (t, ε)-SUF-CLMMRS-CPA-$\mathcal{F}_I$ security, where ε is the non-negligible probability advantage.

Game 6: The game is an interaction between the challenger $\mathcal{C}$ and the forger $\mathcal{F}_{II}$ under SUF-CLMMRS-CPA. Define Π as a certificateless multi-message and multi-receiver anonymous signcryption algorithm, and the specific game interaction is shown as follows:

Setup: The step is the same as Setup in ***Game 2***.

Phase 1: Receiving s and *params* from $\mathcal{C}$, $\mathcal{F}_{II}$ outputs n target identities $L = \{ID_1, ID_2, \ldots, ID_n\}$ and sends them to $\mathcal{C}$.

Attack: $\mathcal{F}_{II}$ asks C for the same queries as Phase 2 in ***Game 2***, and $\mathcal{C}$ makes according responds.

Forgery: $\mathcal{F}_{II}$ outputs the forged signcryption ciphertext σ^* and a group of receivers' identities $L^* = \{ID_1, ID_2, \ldots, ID_n\}$. If the signcryption ciphertext σ^* can be decrypted and verified correctly by any receiver in L^*, $\mathcal{F}_{II}$ wins the game. Otherwise, $\mathcal{F}_{II}$ fails. However, it is worth noting that σ^* is not generated by *Signcryption query*, and other restrictions are the same as Phase 3 in ***Game 2***.

Definition 9: If for $\mathcal{F}_{II}$ under SUF-CLMMRS-CPA, its probability advantage of winning ***Game 6*** meets $Adv_{\prod}^{SUF-CLMMRS-CPA}(\mathrm{F}_{II}) \leq \varepsilon$ within PPT t, the algorithm Π is said to meet (t, ε)-SUF-CLMMRS-CPA-$\mathcal{F}_{II}$ security, where ε is the non-negligible probability advantage.

III. THE PROPOSED SCHEME

Participants of the proposed scheme consist of KGC, the sender S and receivers $R_1, R_2, \ldots, R_n$. The specific scheme is made up of *Setup algorithm*, *Extract key algorithm*, *Signcryption algorithm* and *Designcryption algorithm*. The specific *Extract key algorithm* consists of *Set secret value algorithm*, *Extract partial private key algorithm*, *Set public key algorithm* and *Set private key algorithm*, shown as follows:

A. SETUP ALGORITHM

KGC executes the algorithm to generate the system master key s and the public parameters *params*, shown as follows:

1) With the security parameter λ as input, KGC chooses a large prime number p, an elliptic curve $E(F_p)$ on the finite field F_p, an addition cyclic group G_p on $E(F_p)$, and one generator P of G_p;

2) Randomly choose $s \in Z_p^*$ as the system master key, and compute $P_{\text{pub}} = sP$ as the system public key;

3) Choose a pair of secure symmetric encryption /decryption function E_x/D_x (for example *AES*), where x is the symmetric key;

4) Define five secure anti-collision hash functions:

$$H_0 : \{0,1\}^* \times G_p \times G_p \to Z_p^*, \quad H_1 : \{0,1\}^* \times G_p \to Z_p^*$$
$$H_2 : Z_p^* \times G_p \to \{0,1\}^*, \quad H_3 : Z_p^* \to \{0,1\}^*,$$
$$H_4 : \{0,1\}^* \times Z_p^* \times Z_p^* \times \ldots \times Z_p^* \times \{0,1\}^* \times G_p \times Z_p^* \to Z_p^*;$$

5) Publish the public parameters $params = \{p, E(F_p), F_p, G_p, P_{\text{pub}}, E_x, D_x, H_0, H_1, H_2, H_3, H_4\}$, and keep s secret.

B. EXTRACT KEY ALGORITHM

KGC and the user jointly execute the algorithm to generate the user's public key and private key, shown as follows:

1) SET SECRET VALUE ALGORITHM

The user randomly chooses an integer $d_i \in Z_p^*$ as his/her own secret value, and computes $D_i = d_iP$ as his/her own secret value parameter, then sends ID_i and D_i to KGC through the public channel.

2) EXTRACT PARTIAL PRIVATE KEY ALGORITHM

Receiving ID_i and D_i from the user, KGC randomly chooses an integer $t_i \in Z_p^*$, and computes the user's partial public key $T_i = t_iP$ and partial private key $v_i = l_it_i + s(\text{mod}p)$, where $l_i = H_0(ID_i, D_i, T_i)$. Then, KGC sends v_i to the user through the secure channel, and sends T_i to the user through the public channel, respectively.

3) SET PUBLIC KEY ALGORITHM

Receiving v_i and T_i from KGC, the user verifies whether the equation $v_iP = H_0(ID_i, D_i, T_i)T_i + P_{\text{pub}}$ holds. If yes, the user accepts v_i and T_i, computes $PK_i = D_i + H_0(ID_i, D_i, T_i)T_i$ as his/her own public key, and sends PK_i to KGC for publication. Otherwise, the user rejects v_i and T_i.

4) SET PRIVATE KEY ALGORITHM

The user computes $x_i = d_iH_1(ID_i, PK_i)$ and $y_i = v_iH_1(ID_i, PK_i)$, and sets $SK_i = (x_i, y_i)$ as his/her own private key.

C. SIGNCRYPTION ALGORITHM

With the sender's private key SK_S, the public parameters *params* and the receivers' public keys $\{PK_1, PK_2, \ldots, PK_n\}$ as input, the sender S signcrypts the plaintext message set $M = \{m_1, m_2, \ldots, m_n\}$ as follows:

1) Randomly choose an integer $r \in Z_p^*$, and compute $R = rP$, $K_i = rH_1(ID_i, PK_i)(PK_i + P_{\text{pub}})$ and $\alpha_i = H_0(ID_i, K_i, R)$, for $i = 1, 2, \ldots, n$;

2) Randomly choose an integer $\theta \in Z_p^*$, and compute

$$\varphi(x) = \prod_{i=1}^{n}(x - \alpha_i) + \theta(\text{mod } p)$$
$$= x^n + c_{n-1}x^{n-1} + \ldots + c_1x + c_0,$$

where $c_i \in Z_p^*$;

3) Compute $S = (H_2(\alpha_1, R)||H_3(\alpha_1) \oplus m_1, H_2(\alpha_2, R)||H_3(\alpha_2) \oplus m_2, \ldots, H_2(\alpha_n, R)||H_3(\alpha_n) \oplus m_n)$, $\beta = H_3(\theta)$, and $V = E_\beta(S||ID_S)$;

4) Compute $w = (x_S + y_S)r^{-1}$;

5) Compute $z = H_4(S, \theta, c_0, c_1, \ldots, c_{n-1}, V, R, w)$;

6) Set $(c_0, c_1, \ldots, c_{n-1}, R, V, w, z)$ as the signcryption ciphertext σ, and broadcast it in the communication channel.

D. DESIGNCRYPTION ALGORITHM

Receiving the signcryption ciphertext $\sigma = (c_0, c_1, \ldots, c_{n-1}, R, V, w, z)$, every receiver R_i uses his/her own private key SK_i to designcrypt σ as follows:

1) Compute $K_i' = (x_i + y_i)R$ and $\alpha_i' = H_0(ID_i, K_i', R)$;

2) Compute $\varphi(x) = x^n + c_{n-1}x^{n-1} + \ldots + c_1x + c_0$, $\theta' = \varphi(\alpha_i')$ and $\beta' = H_3(\theta')$;

3) Compute $S'||ID_S = D_{\beta'}(V)$ and $z' = H_4(S', \theta, c_0, c_1, \ldots, c_{n-1}, V, R, w)$, and check whether the equation $z' = z$ holds. If yes, the receiver R_i continues with the following steps. Otherwise, the receiver R_i rejects S', and exits the designcryption algorithm;

4) Compute $H_2(\alpha_i', R)$ and $H_3(\alpha_i')$, find out the corresponding $H_2(\alpha_i, R)||(H_3(\alpha_i) \oplus m_i)$ in S' by $H_2(\alpha_i', R)$, and compute the plaintext message $m_i = (H_3(\alpha_i) \oplus m_i) \oplus H_3(\alpha_i)$.

5) The receiver R_i obtains the sender's public key PK_S, and checks whether the equation $wR = H_1(ID_S, PK_S)(PK_S + P_{\text{pub}})$ holds. If yes, the receiver R_i accepts the plaintext message m_i. Otherwise, the receiver R_i rejects the plaintext message m_i, and exits the designcryption algorithm.

IV. CORRECTNESS ANALYSIS AND SECURITY PROOFS

A. CORRECTNESS ANALYSIS

Theorem 1: The user's partial private key verification is correct in *Extract key algorithm*.

Proof: The correctness of the user's partial private key verification is guaranteed by the establishment of the equation $v_iP = H_0(ID_i, D_i, T_i)T_i + P_{\text{pub}}$, and the deduction that the equation holds is shown as follows:

$$\begin{aligned} v_iP &= (l_it_i + s(\text{mod}p))P \\ &= l_iT_i + P_{pub} \\ &= H_0(ID_i, D_i, T_i)T_i + P_{pub}. \end{aligned}$$

Through the above derivation, it can be seen that the equation $v_iP = H_0(ID_i, D_i, T_i)T_i + P_{\text{pub}}$ holds. As a result, the verification of the user's partial private key in *Extract key algorithm* is correct. ■

Theorem 2: The *Designcryption algorithm* is correct.

Proof: The correctness of *Designcryption algorithm* is guaranteed by establishments of equations $z' = z$ and $wR = H_1(ID_S, PK_S)(PK_S + P_{\text{pub}})$, and deductions that these two equations hold are shown in the following 1) and 2), respectively.

1) For every receiver R_i, with the signcryption ciphertext σ, he/she has $K_i' = (x_i + y_i)R$ and $\alpha_i' = H_0(ID_i, K_i', R)$. Then, with α_i', he/she can compute $\theta' = \varphi(\alpha_i')$, and then get $\beta' = H_3(\theta')$ and $S'||ID_S = D_{\beta'}(V)$. Finally, he/she has

$z' = H_4(S', \theta, c_0, c_1, \ldots, c_{n-1}, V, R, w)$. Thus, the equation $z' = z$ holds.

2) When decrypting out the sender's identity ID_S, the receiver can obtain the sender's public key and has

$$\begin{aligned} wR &= (x_S + y_S)r^{-1}R \\ &= (d_S + v_S)H_1(ID_S, PK_S)\, r^{-1}rP \\ &= H_1(ID_S, PK_S)\,(d_S + v_S)P \\ &= H_1(ID_S, PK_S)\,(D_S + (l_S t_S + s)P) \\ &= H_1(ID_S, PK_S)\,(D_S + l_S T_S + sP) \\ &= H_1(ID_S, PK_S)\,(D_S + H_0(ID_S, D_S, T_S)T_S + sP) \\ &= H_1(ID_S, PK_S)\,(PK_S + P_{\text{pub}}) \end{aligned}$$

That is to say, the equation $wR = H_1(ID_S, PK_S)(PK_S + P_{\text{pub}})$ holds.

Through the derivations of 1) and 2) above, it can be seen that equations $z' = z$ and $wR = H_1(ID_S, PK_S)(PK_S + P_{\text{pub}})$ hold. As a result, the *Designcryption algorithm* is correct. ■

B. SECURITY PROOFS

Based on security models in Section 2, we prove the security of the proposed scheme as follows: the message confidentiality is dependent on the establishment of the following ***Theorem 3*** and ***Theorem 4***, the receiver anonymity relies on the establishment of the following ***Theorem 5*** and ***Theorem 6***, and the unforgeability depends on the establishment of the following ***Theorem 7*** and ***Theorem 8***.

Theorem 3: IND-CLMMRS-CCA against the adversary $\mathcal{A}_I$. Under the random oracle model, if $\mathcal{A}_I$ under IND-CLMMRS-CCA can win ***Game 1*** with the non-negligible probability advantage ε in PPT t ($\mathcal{A}_I$ can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_{sk} times set private key queries,q_r times public key replacement queries, q_s times signcryption queries and q_{us} times designcryption queries.), the CDHP can be solved by the challenger $\mathcal{C}$ with the non-negligible probability advantage $\varepsilon' \geq \frac{\varepsilon}{}\left(2nq_s + q_{H_2} + q_{H_1}\right)\left(1 - \frac{q_s(nq_s+q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right)$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$, where t_{pm} is the time of a scalar point multiplication on ECC operation.

Proof: Assume that $\mathcal{A}_I$ attacks IND-CLMMRS-CCA security of the proposed scheme, $\mathcal{C}$ is a CDHP challenger, and H_0, H_1, H_2, H_3 and H_4 are hash functions defined under the random oracle model. With a set of given elements $\langle P, aP, bP\rangle$, $\mathcal{C}$ hopes to solve the CDHP by interacting with $\mathcal{A}_I$. The specific interactions between $\mathcal{A}_I$ and $\mathcal{C}$ are shown as follows:

Setup: $\mathcal{C}$ executes the algorithm to generate the system master key $s = a \in Z_p^*$ and the public parameters $params = \{p, E(F_p), F_p, G_p, P_{sys} = aP, E_x, D_x, H_0, H_1, H_2, H_3, H_4\}$. Then, $\mathcal{C}$ sends $params$ to $\mathcal{A}_I$ and keeps s secret.

Phase 1: Receiving $params$ from $\mathcal{C}$, $\mathcal{A}_I$ outputs n target identities $L = \{ID_1, ID_2, \ldots, ID_n\}$ and sends them to $\mathcal{C}$. Then, $\mathcal{A}_I$ asks the challenger $\mathcal{C}$ for a series of the following H_i (i = 0, 1, 2, 3, 4) *queries*, and the challenger $\mathcal{C}$ makes according responds:

1) H_0-query: With the tuple (ID_i, D_i, T_i) as input, $\mathcal{A}_I$ queries $\mathcal{C}$ for H_0 hash value. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, D_i, T_i, l_i) in list L_0-H_0. If yes, $\mathcal{C}$ returns l_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly chooses an integer $l_i \in Z_p^*$, returns l_i to $\mathcal{A}_I$, and stores the tuple (ID_i, D_i, T_i, l_i) in list L_0-H_0.

2) H_0-query: With the tuple (ID_i, K_i, R) as input, $\mathcal{A}_I$ queries $\mathcal{C}$ for H_0 hash value. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, K_i, R, α_i) in list L_1-H_0. If yes, $\mathcal{C}$ returns α_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly chooses an integer $\alpha_i \in Z_p^*$, returns α_i to $\mathcal{A}_I$, and stores the tuple (ID_i, K_i, R, α_i) in list L_1-H_0.

3) H_1-query: With the tuple (ID_i, PK_i) as input, $\mathcal{A}_I$ queries $\mathcal{C}$ for H_1 hash value. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, PK_i, ζ_i) in list L-H_1. If yes, $\mathcal{C}$ returns ζ_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly chooses an integer $\zeta_i \in Z_p^*$, returns ζ_i to $\mathcal{A}_I$, and stores the tuple (ID_i, PK_i, ζ_i) in list L-H_1.

4) H_2-query: With the tuple (α_i, R) as input, $\mathcal{A}_I$ queries $\mathcal{C}$ for H_2 hash value. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (α_i, R, γ_i) in list L-H_2. If yes, $\mathcal{C}$ returns γ_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly chooses an integer $\gamma_i \in Z_p^*$, returns γ_i to $\mathcal{A}_I$, and stores the tuple (α_i, R, γ_i) in list L-H_2.

5) H_3-query: With the tuple (α_i) as input, $\mathcal{A}_I$ queries $\mathcal{C}$ for H_3 hash value. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (α_i, δ_i) in list L_0-H_3. If yes, $\mathcal{C}$ returns δ_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly chooses an integer $\delta_i \in Z_p^*$, returns δ_i to $\mathcal{A}_I$, and stores the tuple (α_i, δ_i) in list L_0-H_3.

6) H_3-query: With the tuple (θ_i) as input, $\mathcal{A}_I$ queries $\mathcal{C}$ for H_3 hash value. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (θ_i, β_i) in list L_1-H_3. If yes, $\mathcal{C}$ returns β_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly chooses an integer $\beta_i \in Z_p^*$, returns β_i to $\mathcal{A}_I$, and stores the tuple (θ_i, β_i) in list L_1-H_3.

7) H_4-query: With the tuple (S_i, θ_i, c_i, V_i, R_i, w_i) as input, $\mathcal{A}_I$ queries $\mathcal{C}$ for H_4 hash value, where $c_i = (c_{i_0}, c_{i_1}, \ldots, c_{in_{-1}})$. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (S_i, θ_i, c_i, V_i, R_i, w_i, z_i) in list L-H_4. If yes, $\mathcal{C}$ returns z_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ randomly chooses an integer $z_i \in Z_p^*$, returns z_i to $\mathcal{A}_I$, and stores the tuple (S_i, θ_i, c_i, V_i, R_i, w_i, z_i) in list L-H_4.

Phase 2: $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of the following queries, and $\mathcal{C}$ makes according responds:

1) *Key query*: With ID_i as input, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ gets the tuple (ID_i, d_i, v_i, SK_i, PK_i) from list L-K. Otherwise, $\mathcal{C}$ randomly chooses d_i, l_i, $a_i \in Z_p^*$, sets $l_i = H_0(\mathrm{ID}_i, D_i, T_i)$ and $T_i = (a_i P\text{-}P_{sys})l_i^{-1}$, and computes $PK_i = D_i + l_i T_i$, $x_i = d_i H_1(ID_i, PK_i)$ and $y_i = v_i H_1(ID_i, PK_i)$, where $D_i = d_i P$, $v_i = a_i$ and $v_i P = l_i T_i + P_{sys}$. Then, $\mathcal{C}$ performs the following steps:

a) If $ID_i = ID_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ sets d_i as the secret key, $v_i = \perp$ as the partial private key, $SK_i = (x_i, \perp)$ as the

private key, and PK_i as the public key. Then, $\mathcal{C}$ stores the tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K.

b) If $ID_i \neq ID_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ sets d_i as the secret value, v_i as the partial private key, $SK_i = (x_i, y_i)$ as the private key, and PK_i as the public key. Then, $\mathcal{C}$ stores the tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K.

$\mathcal{C}$ updates the tuple (ID_i, D_i, T_i, l_i) in list L_0-H_0.

2) *Set secret value query*: $\mathcal{A}_I$ queries $\mathcal{C}$ for the secret value of the identity ID_i. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ returns d_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ preforms *Key query* to obtain (ID_i, d_i, v_i, SK_i, PK_i), and returns d_i to $\mathcal{A}_I$.

3) *Extract partial private key query*: $\mathcal{A}_I$ queries $\mathcal{C}$ for the partial private key of the identity ID_i. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ responds as follows:

a) If $ID_i = \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ returns "failure" to $\mathcal{A}_I$.

b) If $ID_i \neq \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ returns v_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ preforms *Key query* to obtain (ID_i, d_i, v_i, SK_i, PK_i), and returns v_i to $\mathcal{A}_I$.

4) *Set public key query*: $\mathcal{A}_I$ queries $\mathcal{C}$ for the public key of the identity ID_i. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ returns PK_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ preforms *Key query* to obtain (ID_i, d_i, v_i, SK_i, PK_i), and returns PK_i to $\mathcal{A}_I$.

5) *Set private key query*: $\mathcal{A}_I$ queries $\mathcal{C}$ for the private key of the identity ID_i. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ responds as follows:

a) If $ID_i = \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ returns "failure" to $\mathcal{A}_I$.

b) If $ID_i \neq \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ returns SK_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ preforms *Key query* to obtain (ID_i, d_i, v_i, SK_i, PK_i), and returns SK_i to $\mathcal{A}_I$.

6) *Public key replacement query*: With the public key PK_i', $\mathcal{A}_I$ requests $\mathcal{C}$ for the public key replacement of the identity ID_i. Receiving the request from $\mathcal{A}_I$, $\mathcal{C}$ finds out the tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K, and replaces the original public key PK_i with PK_i'.

7) *Signcryption query*: With receivers' identities $L^* = \{ID_1, ID_2, \ldots, ID_n\}$ and the sender's identity ID_S, $\mathcal{A}_I$ queries $\mathcal{C}$ for the signcryption of the plaintext message set $M = \{m_1, m_2, \ldots, m_n\}$. Receiving the query from $\mathcal{A}_I$, $\mathcal{C}$ signcrypts M as follows:

a) If $ID_S \neq \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ finds out the tuple (ID_S, d_S, v_S, SK_S, PK_S) in L-K, performs signcryption algorithm to obtain ($c_0, c_1, \ldots, c_{n-1}, R, V, w, z$), and returns ($c_0, c_1, \ldots, c_{n-1}, R, V, w, z$) to $A_I i_1$.

b) If $ID_S = \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ signcrypts M as follows:

(a) Randomly choose an integer $r \in Z_p^*$, and compute $R = rP$;

(b) Find out the tuple (ID_i, K_i, R, α_i) in list L_1-H_0, randomly choose an integer $\theta \in Z_p^*$, and compute

$$\varphi(x) = \prod_{i=1}^{n}(x - \alpha_i) + \theta(\mathrm{mod}\ p)$$
$$= x^n + c_{n-1}x^{n-1} + \ldots + c_1 x + c_0;$$

(c) Find out the tuple (α_i, R, γ_i) in list L-H_2 and the tuple (α_i, δ_i) in list L_0-H_3, and compute $S = (\gamma_1 || \delta_1 \oplus m_1, \gamma_2 || \delta_2 \oplus m_2, \ldots, \gamma_n || \delta_n \oplus m_n)$;

(d) Find out the tuple (θ_i, β_i) in list L_1-H_3, and compute $V = E_{\beta i}(S || ID_S)$;

(e) Randomly choose an integer $k_S \in Z_p^*$, and compute $w = r^{-1}k_S$;

(f) Find out the tuple (S, θ, $c_0, c_1, \ldots, c_{n-1}$, V, R, w, z) in list L-H_4, set ($c_0, c_1, \ldots, c_{n-1}, R, V, w, z$) as the signcryption ciphertext σ, and return it to $\mathcal{A}_I$.

8) *Designcryption query*: With receivers' identities $L^* = \{ID_1, ID_2, \ldots, ID_n\}$, $\mathcal{A}_I$ queries $\mathcal{C}$ for the designcryption of the signcryption ciphertext $\sigma = (c_0, c_1, \ldots, c_{n-1}, R, V, w, z)$. Receiving the query from $\mathcal{A}_I$, C designcrypts σ as follows:

a) Search for the tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K to obtain SK_i. If there is no the tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K, $\mathcal{C}$ returns "failure" to $\mathcal{A}_I$;

b) Compute $K_i' = (x_i + y_i)R$, find out the tuple (ID_i, K_i', R, α_i) in list L_1-H_0, and compute $\varphi(x) = x^n + c_{n-1}x^{n-1} + \ldots + c_1 x + c_0$ and $\theta' = \varphi(\alpha_i')$;

c) Find out the tuple (θ', β') in list L_1-H_3, and compute $S' || ID_S = D_{\beta'}(V)$;

d) Find out the tuple (S', θ', $c_0, c_1, \ldots, c_{n-1}$, V, R, w, z') in list L-H_4, and judge whether the equation $z' = z$ holds. If

yes, $\mathcal{C}$ computes $H_2(\alpha_i', R)$, finds out the corresponding $H_2(\alpha_i, R) \parallel (H_3(\alpha_i) \oplus m_i)$ in S', computes $m_i = (H_3(\alpha_i) \oplus m_i) \oplus H_3(\alpha_i)$, and returns m_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ returns "failure" to $\mathcal{A}_I$.

Challenge: $\mathcal{A}_I$ chooses the sender's identity ID_S, selects two plaintext message sets $M_0 = \{m_1^0, m_2^0, \ldots, m_n^0\}$ and $M_1 = \{m_1^1, m_2^1, \ldots, m_n^1\}$, where $|m_i^0| = |m_i^1|$ $(1 \leq i \leq n)$, and sends two plaintext message sets and the sender's identity ID_S to $\mathcal{C}$. Receiving $\{M_0, M_1, ID_S\}$ from $\mathcal{A}_I$, $\mathcal{C}$ randomly chooses a bit $\mu \in \{0, 1\}$, and calculates the signcryption ciphertext σ^* as follows:

1) Compute $K_j = b(D_j + l_j T_j)$;

2) Choose $\alpha_j \in Z_p^*$, for $j = 1, 2, \ldots, n$;

3) Choose $\theta \in Z_p^*$ and compute

$$\varphi(x) = \prod_{i=1}^{n}(x - \alpha_i) + \theta(\mathrm{mod}\ p)$$
$$= x^n + c_{n-1}x^{n-1} + \ldots c_1 x + c_0;$$

4) Choose $w \in Z_p^*$, set $\sigma^* = (c_0, c_1, \ldots, c_{n-1}, R_j = b(Q_j + D_j), V = E_\beta(S_\mu || ID_S), w, z = H_4(S_\mu, \theta, c_0, c_1, \ldots, c_{n-1}, V, R_j, w))$, and return σ^* to $\mathcal{A}_I$, where $Q_j = l_j T_j + P_{sys}$, $\beta = H_3(\theta)$ and $S_\mu = (H_2(\alpha_1, R_j) || H_3(\alpha_1) \oplus m_1^\mu, H_2(\alpha_2, R_j) || H_3(\alpha_2) \oplus m_2^\mu, \ldots, H_2(\alpha_n, R_j) || H_3(\alpha_n) \oplus m_n^\mu)$.

Phase 3: $\mathcal{A}_I$ asks $\mathcal{C}$ for the same queries as Phase 2. However, there are the following restrictions:

1) $\mathcal{A}_I$ cannot query for the partial private key of any target identity in L.

2) A_I cannot query for the private key of the target identity whose public key has been replaced.

3) $\mathcal{A}_I$ cannot query for the designcryption of the signcryption ciphertext σ^*.

Guess: According to the phases performed by $\mathcal{A}_I$ and the challenger $\mathcal{C}$ above, $\mathcal{A}_I$ outputs a bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, $\mathcal{A}_I$ wins ***Game 1***, and $\mathcal{C}$ outputs $abP = R_j\text{-}K_j$ as a solution to the CDHP. Otherwise, $\mathcal{C}$ outputs "failure".

In the interaction process above, it is concluded that $\mathcal{A}_I$ under IND-CLMMRS-CCA can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_{pk} times set public key queries, q_r times public key replacement queries, q_s times signcryption queries and q_{us} times designcryption queries. Therefore, the probability advantage that the CDHP can be solved by the challenger $\mathcal{C}$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$ is

$$\varepsilon' \geq \frac{\varepsilon}{(2nq_s + q_{H_2} + q_{H_1})}\left(1 - \frac{q_s(nq_s + q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right),$$

where t_{pm} is the time of a scalar point multiplication on ECC operation, ε is the non-negligible probability advantage, and t is the probability polynomial time. ■

Theorem 4: IND-CLMMRS-CCA against the adversary $\mathcal{A}_{II}$. Under the random oracle model, if $\mathcal{A}_{II}$ under IND-CLMMRS-CCA can win ***Game 2*** with the non-negligible probability advantage ε in PPT t ($\mathcal{A}_{II}$ can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_{sk} times set private key queries, q_r times public key replacement queries, q_s times signcryption queries and q_{us} times designcryption queries.), the CDHP can be solved by the challenger $\mathcal{C}$ with the probability advantage $\varepsilon' \geq \varepsilon\left(1 - \frac{q_s(nq_s+q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right)$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$, where t_{pm} is the time of a scalar point multiplication on ECC operation.

Proof: Assume that $\mathcal{A}_{II}$ attacks IND-CLMMRS-CCA security of the proposed scheme, $\mathcal{C}$ is a CDHP challenger, and H_0, H_1, H_2, H_3 and H_4 are hash functions defined under the random oracle model. With a set of given elements $\langle P, aP, bP\rangle$, $\mathcal{C}$ hopes to solve the CDHP by interacting with $\mathcal{A}_{II}$. The specific interactions between $\mathcal{A}_{II}$ and $\mathcal{C}$ are shown as follows:

Setup: $\mathcal{C}$ executes the algorithm to generate the system master key $s = \beta \in Z_p^*$ and the public parameters $params = \{p, E(F_p), F_p, G_p, P_{sys} = \beta P, P_1 = aP, E_x, D_x, H_0, H_1, H_2, H_3, H_4\}$, where $a \in Z_p^*$. Then, $\mathcal{C}$ sends s and $params$ to $\mathcal{A}_{II}$.

Phase 1: Receiving s and $params$from $\mathcal{C}$, $\mathcal{A}_{II}$ outputs n target identities $L = \{ID_1, ID_2, \ldots, ID_n\}$ and sends them to $\mathcal{C}$. Then, $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of the same H_i ($i = 0$, 1, 2, 3, 4) *queries* as Phase 1 in ***Theorem 3***, and the challenger $\mathcal{C}$ makes according responds.

Phase 2: $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of the following queries, and $\mathcal{C}$ makes according responds:

1) *Key query*: With ID_i as input, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ gets the tuple (ID_i, d_i, v_i, SK_i, PK_i) from list L-K. Otherwise, $\mathcal{C}$ randomly chooses d_i, l_i, $a_i \in Z_p^*$, sets $l_i = H_0(\mathrm{ID}_i, D_i, T_i)$, computes $T_i = a_iP$, $v_i = l_ia_i + \beta$ and $D_i = P_1\text{-}d_iP$, and computes $PK_i = D_i + l_iT_i$, $x_i = d_iH_1(ID_i, PK_i)$ and $y_i = v_iH_1(ID_i, PK_i)$. Then, $\mathcal{C}$ performs the following steps:

a) If $ID_i = ID_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ sets $d_i = \perp$ as the secret key, v_i as the partial private key, $SK_i = (\perp, y_i)$ as the private key, and PK_i as the public key. Then, $\mathcal{C}$ stores the tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K.

b) If $ID_i \neq ID_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ sets d_i as the secret value, v_i as the partial private key, $SK_i = (a\text{-}x_i, y_i)$ as the private key, and PK_i as the public key. Then, $\mathcal{C}$ stores the tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K.

$\mathcal{C}$ updates the tuple (ID_i, D_i, T_i, l_i) in L_0-H_0.

2) *Set secret value query*: $\mathcal{A}_{II}$ queries $\mathcal{C}$ for the secret value of the identity ID_i. Receiving the query from $\mathcal{A}_{II}$, $\mathcal{C}$ responds as follows:

a) If $ID_i = ID_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ returns "failure" to $\mathcal{A}_{II}$.

b) If $ID_i \neq ID_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ returns d_i to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ preforms *Key query* to obtain (ID_i, d_i, v_i, SK_i, PK_i), and returns d_i to $\mathcal{A}_{II}$.

3) *Extract partial private key query*: $\mathcal{A}_{II}$ queries $\mathcal{C}$ for the partial private key of the identity ID_i. Receiving the query from $\mathcal{A}_{II}$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ returns v_i to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ preforms *Key query* to obtain (ID_i, d_i, v_i, SK_i, PK_i), and returns v_i to $\mathcal{A}_{II}$.

4) *Set public key query*: $\mathcal{A}_{II}$ queries $\mathcal{C}$ for the public key query of the identity ID_i. Receiving the query from $\mathcal{A}_{II}$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ returns PK_i to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ preforms *Key query* to obtain (ID_i, d_i, v_i, SK_i, PK_i), and returns PK_i to $\mathcal{A}_{II}$.

5) *Set private key query:* $\mathcal{A}_{II}$ queries $\mathcal{C}$ for the private key of the identity ID_i. Receiving the query from $\mathcal{A}_{II}$, $\mathcal{C}$ responds as follows:

a) If $ID_i = \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ returns "failure" to $\mathcal{A}_{II}$.

b) If $ID_i \neq \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ first checks whether there is a tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K. If yes, $\mathcal{C}$ returns SK_i to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ preforms *Key query* to obtain (ID_i, d_i, v_i, SK_i, PK_i), and returns SK_i to $\mathcal{A}_{II}$.

6) *Public key replacement query*: With the public keyPK_i', $\mathcal{A}_{II}$ requests $\mathcal{C}$ for the public key replacement of the identity ID_i. Receiving the request from $\mathcal{A}_{II}$, $\mathcal{C}$ responds as follows:

a) If $ID_i = \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ returns "failure" to $\mathcal{A}_{II}$.

b) If $ID_i \neq \mathrm{ID}_j$, for $j \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ finds out the tuple (ID_i, d_i, v_i, SK_i, PK_i) in list L-K, and replaces the original public key PK_i with PK_i'.

7) *Signcryption query*: The step is the same *Signcryption query* as Phase 2 in ***Theorem 3***.

8) *Designcryption query*: The step is the same *Designcryption query* as Phase 2 in ***Theorem 3***.

Challenge: $\mathcal{A}_{II}$ chooses the sender's identity ID_S, selects two plaintext message sets $M_0 = \{m_1^0, m_2^0, \ldots, m_n^0\}$ and $M_1 = \{m_1^1, m_2^1, \ldots, m_n^1\}$, where $|m_i^0| = |m_i^1|$ $(1 \le i \le n)$, and sends two plaintext message sets and the sender's identity ID_S to $\mathcal{C}$. Receiving $\{M_0, M_1, ID_S\}$ from $\mathcal{A}_{II}$, $\mathcal{C}$ randomly chooses a bit $\mu \in \{0, 1\}$, and calculates the signcryption ciphertext σ^* as follows:

1) Compute $K_j = b(D_j + Q_j)$, where $Q_j = l_jT_j + P_{\text{sys}}$ and $D_j = P_1\text{-}u_jP$;

2) Choose $\alpha_j \in Z_p^*$, for $j = 1, 2, \ldots, n$;

3) Choose $\theta \in Z_p^*$ and compute

$$\begin{aligned}\varphi(x) &= \prod_{i=1}^{n}(x - \alpha_i) + \theta(\text{mod } p)\\ &= x^n + c_{n-1}x^{n-1} + \ldots + c_1x + c_0;\end{aligned}$$

4) Choose $w \in Z_p^*$, set $\sigma^* = (c_0, c_1, \ldots, c_{n_{-1}}, R_j = b(P_1\text{-}D_j\text{-}Q_j), V = E_\beta(S_\mu||ID_S), w, z = H_4(S_\mu, \theta, c_0, c_1, \ldots, c_{n_{-1}}, V, R_j, w))$, and return σ^* to $\mathcal{A}_{II}$, where$Q_j = l_jT_j + P_{sys}$, $D_j = P_1\text{-}u_jP$, $\beta = H_3(\theta)$ and $S_\mu = (H_2(\alpha_1, R_j)||H_3(\alpha_1) \oplus m_1^\mu, H_2(\alpha_2, R_j)||H_3(\alpha_2) \oplus m_2^\mu, \ldots, H_2(\alpha_n, R_j)||H_3(\alpha_n) \oplus m_n^\mu)$.

Phase 3: $\mathcal{A}_{II}$ asks $\mathcal{C}$ for the same queries as Phase 2. However, there are the following restrictions:

1) $\mathcal{A}_{II}$ cannot query for the secret value of any target identity in L.

2) A_{II} cannot query for the private key of the target identity whose public key has been replaced.

3) $\mathcal{A}_{II}$ cannot query for the designcryption of the signcryption ciphertext σ^*.

Guess: According to the phases performed by $\mathcal{A}_{II}$ and the challenger $\mathcal{C}$ above, $\mathcal{A}_{II}$ outputs a bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, $\mathcal{A}_{II}$ wins ***Game 2***, and $\mathcal{C}$ outputs $abP = R_j + K_j$ as a solution to the CDHP. Otherwise, $\mathcal{C}$ outputs "failure".

In the interaction process above, it is concluded that $\mathcal{A}_{II}$ under IND-CLMMRS-CCA can ask for at most q_i times H_i queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_s times signcryption queries and q_{us} times designcryption queries. Therefore, the probability advantage that the CDHP can be solved by the challenger $\mathcal{C}$ in the time $t' \le t + O(q_{pk} + nq_s + q_{us})t_{pm}$ is $\varepsilon' \ge \varepsilon\left(1 - \frac{q_s(nq_s + q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right)$, where t_{pm} is the time of a scalar point multiplication on ECC operation, ε is the non-negligible probability advantage, and t is the probability polynomial time. ■

Theorem 5: ANON-IND-CLMMRS-CCA against the adversary $\mathcal{A}_I$. Under the random oracle model, if $\mathcal{A}_I$ under ANON-IND-CLMMRS-CCA can win ***Game 3*** with the non-negligible probability advantage ε in PPT t ($\mathcal{A}_I$ can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_{sk} times set private key queries, q_r times public key replacement queries, q_s times signcryption queries and q_{us} times designcryption queries.), the CDHP can be solved by the challenger $\mathcal{C}$ with the probability advantage $\varepsilon' \ge \frac{\varepsilon}{(2nq_s + q_{H_2} + q_{H_1})}\left(1 - \frac{q_s(nq_s + q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right)$ in the time $t' \le t + O(q_{pk} + nq_s + q_{us})t_{pm}$, where t_{pm} is the time of a scalar point multiplication on ECC operation.

Proof: Assume that $\mathcal{A}_I$ attacks ANON-IND-CLMMRS-CCA security of the proposed scheme, $\mathcal{C}$ is a CDHP challenger, and H_0, H_1, H_2, H_3 and H_4 are hash functions defined under the random oracle model. With a set of given elements $\langle P, aP, bP\rangle$, $\mathcal{C}$ hopes to solve the CDHP by interacting with $\mathcal{A}_I$. The specific interactions between $\mathcal{A}_I$ and $\mathcal{C}$ are shown as follows:

Setup: The step is the same as Setup in ***Theorem 3***.

Phase 1: Receiving *params* from $\mathcal{C}$, $\mathcal{A}_I$ outputs two target identities $L = \{ID_0, ID_1\}$ and sends them to $\mathcal{C}$. Then, $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of the same H_i $(i = 0, 1, 2, 3, 4)$ *queries* as Phase 1 in ***Theorem 3***, and the challenger $\mathcal{C}$ makes according responds.

Phase 2: $\mathcal{A}_I$ asks $\mathcal{C}$ for the same queries as Phase 2 in ***Theorem 3***, and $\mathcal{C}$ makes according responds.

Challenge: $\mathcal{A}_I$ chooses a plaintext message set $M = \{m_1, m_2, \ldots, m_n\}$, a group of receivers' identities $L^* = \{ID_2, ID_3, \ldots, ID_n\}$, and the sender's identity ID_S. Then, $\mathcal{A}_I$ sends the plaintext message set M, receivers' identities L^* and the sender's identity ID_S to $\mathcal{C}$. Receiving $\{M, ID_S, L^*\}$ from $\mathcal{A}_I$, $\mathcal{C}$ randomly chooses a bit $\mu \in \{0, 1\}$, and calculates the signcryption ciphertext σ^* as follows:

1) Compute $K_j = b(D_j + l_jT_j)$;

2) Choose $\alpha_j \in Z_p^*$, for $j = \mu, 2, 3, \ldots, n$;

3) Choose $\theta \in Z_p^*$ and compute

$$\begin{aligned}\varphi(x) &= (x - \alpha_\mu)\prod_{i=2}^{n}(x - \alpha_i) + \theta(\text{mod } p)\\ &= x^n + c_{n-1}x^{n-1} + \ldots + c_1x + c_0;\end{aligned}$$

4) Choose $w \in Z_p^*$, set $\sigma^* = (c_0, c_1, \ldots, c_{n-1}, R_j = b(Q_j + D_j), V = E_\beta(S||ID_S), w, z = H_4(S, \theta, c_0, c_1, \ldots, c_{n-1}, V, R_j, w))$, and return σ^* to $\mathcal{A}_I$, where $Q_j = l_jT_j + P_{sys}$, $\beta = H_3(\theta)$ and $S = (H_2(\alpha_\mu, R_j)||H_3(\alpha_\mu) \oplus m_1, H_2(\alpha_2, R_j)||H_3(\alpha_2) \oplus m_2, \ldots, H_2(\alpha_n, R_j)||H_3(\alpha_n) \oplus m_n)$.

Phase 3: The step is the same as Phase 3 in ***Theorem 3***.

Guess: According to the phases performed by $\mathcal{A}_I$ and the challenger $\mathcal{C}$ above, $\mathcal{A}_I$ outputs a bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, $\mathcal{A}_I$ wins ***Game 3***, and $\mathcal{C}$ outputs $abP = R_j\text{-}K_j$ as a solution to the CDHP. Otherwise, $\mathcal{C}$ outputs "failure".

In the interaction process above, it is concluded that $\mathcal{A}_I$ under ANON-IND-CLMMRS-CCA can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_{pk} times set public key queries, q_r times public key replacement queries, q_s times signcryption queries and q_{us} times designcryption queries. Therefore, the probability advantage that the CDHP can be solved by the challenger $\mathcal{C}$ in the

time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$ is

$$\varepsilon' \geq \frac{\varepsilon}{(2nq_s + q_{H_2} + q_{H_1})}\left(1 - \frac{q_s(nq_s + q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right),$$

where t_{pm} is the time of a scalar point multiplication on ECC operation, ε is the non-negligible probability advantage, and t is the probability polynomial time. ■

Theorem 6: ANON-IND-CLMMRS-CCA against the adversary $\mathcal{A}_{II}$. Under the random oracle model, if $\mathcal{A}_{II}$ under ANON-IND-CLMMRS-CCA can win ***Game 4*** with the non-negligible probability advantage ε in PPT t ($\mathcal{A}_{II}$ can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_{sk} times set private key queries, q_r times public key replacement queries,q_s times signcryption queries and q_{us} times designcryption queries.), the CDHP can be solved by the challenger $\mathcal{C}$ with the probability advantage $\varepsilon' \geq \varepsilon\left(1 - \frac{q_s(nq_s + q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right)$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$, where t_{pm} is the time of a scalar point multiplication on ECC operation.

Proof: Assume that $\mathcal{A}_{II}$ attacks ANON-IND-CLMMRS-CCA security of the proposed scheme, $\mathcal{C}$ is a CDHP challenger, and H_0, H_1, H_2, H_3 and H_4 are hash functions defined under the random oracle model. With a set of given elements $\langle P, aP, bP\rangle$, $\mathcal{C}$ hopes to solve the CDHP by interacting with $\mathcal{A}_{II}$. The specific interactions between $\mathcal{A}_{II}$ and $\mathcal{C}$ are shown as follows:

Setup: The step is the same as setup in ***Theorem 4***.

Phase 1: Receiving s and *params* from $\mathcal{C}$, $\mathcal{A}_{II}$ outputs two target identities $L = \{ID_0, ID_1\}$ and sends them to $\mathcal{C}$. Then, $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of the same H_i ($i = 0, 1, 2, 3, 4$) *queries* as Phase 1 in ***Theorem 3***, and $\mathcal{C}$ makes according responds.

Phase 2: $\mathcal{A}_{II}$ asks $\mathcal{C}$ for the same queries as Phase 2 in ***Theorem 4***, and $\mathcal{C}$ makes according responds.

Challenge: $\mathcal{A}_{II}$ chooses a plaintext message set $M = \{m_1, m_2, \ldots, m_n\}$, a group of receivers' identities $L^* = \{ID_2, ID_3, \ldots, ID_n\}$ and the sender's identity ID_S. Then, $\mathcal{A}_{II}$ sends the plaintext message set M, receivers' identities L^* and the sender's identity ID_S to $\mathcal{C}$. Receiving $\{M, ID_S, L^*\}$ from $\mathcal{A}_{II}$, $\mathcal{C}$ randomly chooses a bit $\mu \in \{0, 1\}$ and calculates the signcryption ciphertext σ^* as follows:

1) Compute $K_j = b(D_j + Q_j)$, where $Q_j = l_jT_j + P_{\text{sys}}$ and $D_j = P_1\text{-}u_jP$;

2) Choose $\alpha_j \in Z_p^*$, for $j = \mu, 2, 3, \ldots, n$;

3) Choose $\theta \in Z_p^*$ and compute

$$\varphi(x) = (x - \alpha_\mu)\prod_{i=2}^{n}(x - \alpha_i) + \theta(\text{mod } p)$$
$$= x^n + c_{n-1}x^{n-1} + \ldots + c_1x + c_0;$$

4) Choose $w \in Z_p^*$, set $\sigma^* = (c_0, c_1, \ldots, c_{n-1}, R_j = b(P_1\text{-}D_j\text{-}Q_j), V = E_\beta(S||ID_S), w, z = H_4(S, \theta, c_0, c_1, \ldots, c_{n-1}, V, R_j, w))$, and return σ^* to $\mathcal{A}_{II}$, where $Q_j = l_jT_j + P_{sys}$, $D_j = P_1\text{-}u_jP$, $\beta = H_3(\theta)$ and $S = (H_2(\alpha_\mu, R_j)||H_3(\alpha_\mu) \oplus m_1, H_2(\alpha_2, R_j)||H_3(\alpha_2) \oplus m_2, \ldots, H_2(\alpha_n, R_j)||H_3(\alpha_n) \oplus m_n)$.

Phase 3: The step is the same as Phase 3 in ***Theorem 4***.

Guess: According to the phases performed by $\mathcal{A}_{II}$ and the challenger $\mathcal{C}$ above, $\mathcal{A}_{II}$ outputs a bit $\mu' \in \{0, 1\}$. If $\mu' = \mu$, $\mathcal{A}_{II}$ wins ***Game 4***, and $\mathcal{C}$ outputs $abP = R_j + K_j$ as a solution to the CDHP. Otherwise, $\mathcal{C}$ outputs "failure".

In the interaction process above, it is concluded that $\mathcal{A}_{II}$ under ANON-IND-CLMMRS-CCA can ask for at most q_i times H_i queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_s times signcryption queries and q_{us} times designcryption queries. Therefore, the probability advantage that the CDHP can be solved by the challenger $\mathcal{C}$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$ is $\varepsilon' \geq \varepsilon\left(1 - \frac{q_s(nq_s + q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right)$, where t_{pm} is the time of a scalar point multiplication on ECC operation, ε is the non-negligible probability advantage, and t is the probability polynomial time. ■

Theorem 7: SUF-CLMMRS-CPA against the forger $\mathcal{F}_I$. Under the random oracle model, if $\mathcal{F}_I$ under SUF-CLMMRS-CPA can win ***Game 5*** with the non-negligible probability advantage ε in PPT t ($\mathcal{F}_I$ can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_{sk} times set private key queries, q_r times public key replacement queries, q_s times signcryption queries and q_{us} times designcryption queries.), the CDHP can be solved by the challenger $\mathcal{C}$ with the probability advantage $\varepsilon' \geq \frac{\varepsilon}{(2nq_s + q_{H_2} + q_{H_1})}\left(1 - \frac{q_s(nq_s + q_{H_0})}{2^n}\right)\left(1 - \frac{q_{us}}{2^n}\right)$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$, where t_{pm} is the time of a scalar point multiplication on ECC operation.

Proof: Assume that $\mathcal{F}_I$ attacks SUF-CLMMRS-CPA security of the proposed scheme, $\mathcal{C}$ is a CDHP challenger, and H_0, H_1, H_2, H_3 and H_4 are hash functions defined under the random oracle model. With a set of given elements $\langle P, aP, bP\rangle$, $\mathcal{C}$ hopes to solve the CDHP by interacting with Φ_I. The specific interactions between $\mathcal{F}_I$ and $\mathcal{C}$ are shown as follows:

Setup: The step is the same as Setup in ***Theorem 3***.

Phase 1: Receiving *params* from $\mathcal{C}$, $\mathcal{F}_I$ outputs n target identities $L = \{ID_1, ID_2, \ldots, ID_n\}$ and sends them to $\mathcal{C}$. Then, $\mathcal{F}_I$ asks $\mathcal{C}$ for a series of the same H_i ($i = 0$, 1, 2, 3, 4) *queries* as Phase 1 in ***Theorem 3***, and $\mathcal{C}$ makes according responds.

Attack: $\mathcal{F}_I$ asks C for the same queries as Phase 2 in ***Theorem 3***, and $\mathcal{C}$ makes according responds.

Forgery: $\mathcal{F}_I$ outputs the forged signcryption ciphertext σ^* and a group of receivers' identities $L^* = \{ID_1, ID_2, \ldots, ID_n\}$. If equations $z' = z$ and $wR = H_1(ID_S, PK_S)(PK_S + P_{\text{pub}})$ hold, the forgery is successful. Then, setting $K_j = b(D_j + l_jT_j)$ and $R_j = b(Q_j + D_j)$, $\mathcal{C}$ outputs $abP = R_j\text{-}K_j$ as a solution to the CDHP, where $Q_j = l_jT_j + P_{sys}$. Otherwise, $\mathcal{C}$ outputs "failure".

In the interaction process, it is concluded that $\mathcal{F}_I$ under SUF-CLMMRS-CPA can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_{pk} times set public key queries, q_r times public key replacement queries, q_s times signcryption queries and q_{us} times designcryption queries. Therefore, the probability advantage that the CDHP can be solved by the challenger $\mathcal{C}$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$ is $\varepsilon' \geq \frac{\varepsilon}{(2nq_s+q_{H_2}+q_{H_1})}\left(1-\frac{q_s(nq_s+q_{H_0})}{2^n}\right)\left(1-\frac{q_{us}}{2^n}\right)$, where t_{pm} is the time of a scalar point multiplication on ECC operation, ε is the non-negligible probability advantage, and t is the probability polynomial time. ■

Theorem 8: SUF-CLMMRS-CPA against the forger $\mathcal{F}_{II}$. Under the random oracle model, if $\mathcal{F}_{II}$ under SUF-CLMMRS-CPA can win ***Game 6*** with the non-negligible probability advantage ε in PPT t (Φ_{II} can ask for at most q_i times H_i queries, q_{sv} times set secret value queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_{sk} times set private key queries, q_r times public key replacement queries, q_s times signcryption queries and q_{us} times designcryption queries.), the CDHP can be solved by the challenger $\mathcal{C}$ with the probability advantage $\varepsilon' \geq \varepsilon\left(1-\frac{q_s(nq_s+q_{H_0})}{2^n}\right)\left(1-\frac{q_{us}}{2^n}\right)$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$, where t_{pm} is the time of a scalar point multiplication on ECC operation.

Proof: Assume that $\mathcal{F}_{II}$ attacks SUF-CLMMRS-CPA security of the proposed scheme, $\mathcal{C}$ is a CDHP challenger, and H_0, H_1, H_2, H_3 and H_4 are hash functions defined under the random oracle model. With a set of given elements $\langle P, aP, bP\rangle$, $\mathcal{C}$ hopes to solve the CDHP by interacting with Φ_{II}. The specific interactions between $\mathcal{F}_{II}$ and $\mathcal{C}$ are shown as follows:

Setup: The step is the same as Setup in ***Theorem 4***.

Phase 1: Receiving s and *params* from $\mathcal{C}$, $\mathcal{F}_{II}$ outputs n target identities $L = \{ID_1, ID_2, \ldots, ID_n\}$ and sends them to $\mathcal{C}$. Then, $\mathcal{F}_{II}$ asks $\mathcal{C}$ for a series of the same H_i (i = 0, 1, 2, 3, 4) *queries* as Phase 1 in ***Theorem 3***, and $\mathcal{C}$ makes according responds.

Attack: $\mathcal{F}_{II}$ asks C for the same queries as Phase 2 in ***Theorem 4***, and $\mathcal{C}$ makes according responds.

Forgery: $\mathcal{F}_{II}$ outputs the forged signcryption ciphertext σ^* and a group of receivers' identities $L^* = \{ID_1, ID_2, \ldots, ID_n\}$. If equations $z' = z$ and $wR = H_1(ID_S, PK_S)(PK_S + P_{\text{pub}})$ hold, the forgery is successful. Then, setting $K_j = b(D_j + Q_j)$ and $R_j = b(P_1\text{-}D_j\text{-}Q_j)$, $\mathcal{C}$ outputs $abP = R_j + K_j$ as a solution to the CDHP, where $Q_j = l_jT_j + P_{\text{sys}}$ and $D_j = P_1\text{-}u_jP$. Otherwise, $\mathcal{C}$ outputs "failure".

In the interaction process, it is concluded that $\mathcal{F}_{II}$ under SUF-CLMMRS-CPA can ask for at most q_i times H_i queries, q_e times extract partial private key queries, q_{pk} times set public key queries, q_s times signcryption queries and q_{us} times designcryption queries. Therefore, the probability advantage that the CDHP can be solved by the challenger $\mathcal{C}$ in the time $t' \leq t + O(q_{pk} + nq_s + q_{us})t_{pm}$ is $\varepsilon' \geq \varepsilon\left(1-\frac{q_s(nq_s+q_{H_0})}{2^n}\right)\left(1-\frac{q_{us}}{2^n}\right)$, where t_{pm} is the time of a scalar point multiplication on ECC operation, ε is the non-negligible probability advantage, and t is the probability polynomial time.

V. COMPARISON AND ANALYSIS OF FUNCTIONS AND EFFICIENCY

Because schemes [12], [15], [20], [21], [23], [25], [28], [30]–[32] have a higher similarity to the proposed scheme in functions or Cryptographic foundation, in order to show our scheme's advantages, we shall compare our scheme with them in terms of functions and efficiency in the following.

A. COMPARISON AND ANALYSIS OF FUNCTIONS

We will compare schemes [12], [15], [20], [21], [23], [25], [28], [30]–[32] with our proposed scheme in terms of functions, shown in TABLE 2.

From TABLE 2, we can see: (1) Schemes [12], [15], [20], [21], [23], and [25] do not meet the receiver anonymity and the decryption fairness, which not only leads the receivers' identities information to be revealed, but also makes authorized receivers decrypt the signcryption ciphertext unfairly. Although schemes [28] and [31] satisfy the receiver anonymity, they do not meet the decryption fairness. (2) Schemes [12], [15], [20], and [21] suffer from the public key certificate management burden because they are constructed based on the PKI-based cryptography, which causes huge expenses in maintaining PKI. Although there exists no the public key certificate management burden in schemes [23] and [28], there exists the key escrow problem as a result of the use of IBC, which means that it is possible for schemes [23] and [28] to be attacked by malicious KGC. Because Wang *et al.*'s scheme [25] is a heterogeneous scheme, and it can shift between the PKI-based cryptography and IBC, there exist both the public key certificate management burden and the key escrow problem in Wang *et al.*'s scheme. (3) Although schemes [30]–[32] are free from the public key certificate management burden and the key escrow problem, and schemes [30] and [32] satisfy both the receiver anonymity and the decryption fairness, they cannot meet the requirements of the multi-message and multi-receiver signcryption scheme.

Through the analyses above, it can be seen that compared with schemes [12], [15], [20], [21], [23], [25], [28], and [30]–[32], the proposed scheme meets more functions shown in TABLE 2, and thus it is more practical for applications.

B. COMPARISON AND ANALYSIS OF EFFICIENCY

For the sake of convenient analyses, we first define some symbols to denote the computational complexity of different mathematical operations which are used in encryption/signcryption processes and decryption /designcryption processes, shown in TABLE 3 (The data are

TABLE 2. Comparison of functions.

Schemes	Receiver anonymity	Decryption fairness	No certificate management burden	No key escrow problem	Multi-message and multi-receiver signcryption
Hassan *et al.*'s [12]	×	×	×	√	√
Han *et al.*'s [15]	×	×	×	√	√
Nizamud *et al.*'s [20]	×	×	×	√	√
Rahman *et al.*'s [21]	×	×	×	√	√
Qiu *et al.*'s [23]	×	×	√	×	√
Wang *et al.*'s [25]	×	×	×	×	√
Niu *et al.*'s [28]	√	×	√	×	√
Islam *et al.*'s [30]	√	√	√	√	×
Hung *et al.*'s [31]	√	×	√	√	×
Pang *et al.*'s [32]	√	√	√	√	×
Our proposed	√	√	√	√	√

TABLE 3. Notations and mathematical operations' computational complexity.

Symbols	Symbols' definition
T_m	Time for a modular multiplication operation;
T_b	Time for a bilinear pairing operation, $T_b \approx 87T_m$;
T_{be}	Time for a pairing-based exponentiation operation, $T_{be} \approx 43.5T_m$;
T_{pm}	Time for a scalar point multiplication on ECC operation, $T_{pm} \approx 29T_m$;
T_{dm}	Time for a divisor multiplication on hyper elliptic curve operation, $T_{dm} \approx 14.5T_m$;
T_{pa}	Time for a point addition on ECC operation, $T_{pa} \approx 0.12T_m$;
T_e	Time for a modular exponentiation operation, $T_e \approx 240T_m$;
T_h	Time for a map to point hash function operation, $T_h \approx 29T_m$;
T_i	Time for a modular inversion operation, $T_i \approx 11.6T_m$;

TABLE 4. Comparison of computational complexity.

TABLE IV
COMPARISON OF COMPUTATIONAL COMPLEXITY

Schemes	Encryption/Signcryption	Decryption/Designcryption
Hassan *et al.*'s [12]	$nT_{pm} \approx 29nT_m$	$2T_{pm}+T_m \approx 59T_m$
Han *et al.*'s [15]	$(2n+1)T_{pm}+nT_h \approx (87n+29)T_m$	$T_{pm}+T_h+2T_b \approx 232T_m$
Nizamud *et al.*'s [20]	$nT_{pm}+T_{pm} \approx (29n+29)T_m$	$3T_{pm} \approx 87T_m$
Rahman *et al.*'s [21]	$(n+2)T_{dm} \approx (14.5n+29)T_m$	$3T_{dm} \approx 43.5T_m$
Qiu *et al.*'s [23]	$nT_b+nT_h+T_{pm} \approx (116n+29)T_m$	$T_b+3T_{pm}+T_{pa} \approx 174.12T_m$
Wang *et al.*'s [25]	$(n+1)T_{pm}+(n+1)T_e \approx (269n+269)T_m$	$2T_b+T_e \approx 414T_m$
Niu *et al.*'s [28]	$(n+2)T_{pm}+2nT_b+2nT_{be}+T_h \approx (290n+87)T_m$	$T_{pm}+4T_b+T_{pa}+T_h \approx 406.12T_m$
Islam *et al.*'s [30](*)	$(2n+1)T_{pm}+2nT_{pa} \approx (58.24n+29)T_m$	$T_{pm} \approx 29T_m$
Hung *et al.*'s [31](*)	$(n+1)T_{pm}+nT_{be}+nT_b+nT_h \approx (188.5n+29)T_m$	$T_{pm}+T_b \approx 116T_m$
Pang *et al.*'s [32](*)	$(n+1)T_{pm}+nT_{pa} \approx (29.12n+29)T_m$	$3T_{pm}+2T_{pa} \approx 87.24T_m$
Our proposed	$(n+1)T_{pm}+nT_{pa} \approx (29.12n+29)T_m$	$3T_{pm}+T_{pa} \approx 87.12T_m$

(*) indicates that the scheme does not meet the requirements of the multi-message and multi-receiver signcryption scheme; n indicates that the number of receivers.

from [21] and [30]). It is worth noting that only the mathematical operations in TABLE 3 are considered, because compared with those in TABLE 3, the computational time of other mathematical operations can be ignored.

The comparison results of computational complexity between the proposed scheme and schemes [12], [15], [20], [21], [23], [25], [28], [30]–[32] are shown in TABLE 4.

From TABLE 4, we can see that both in encryption/signcryption processes and in decryption /designcryption processes, compared with schemes [15], [23], [25], [28], [31], and [32], the proposed scheme is low in computational complexity. Although the proposed scheme is slightly higher than schemes [12], [20], [21], and [30] in computational complexity, which results from the newly added functions summarized in TABLE 2, we think that this deficiency is acceptable when considering the overall advantages of the proposed scheme. From a comprehensive perspective, compared with schemes [12], [15], [20], [21], [23], [25], [28], and [30]–[32], the proposed scheme is relatively high in efficiency.

VI. CONCLUSION

Aiming at the public key certificate management burden and the key escrow problem in the existing multi-message

and multi-receiver signcryption schemes, we introduce the concept of CLC-PKC into the designing of the multi-message and multi-receiver signcryption and propose a certificateless multi-message and multi-receiver signcryption scheme. Compared with most existing multi-message and multi-receiver signcryption schemes, the proposed scheme is free from the public key certificate management burden and the key escrow problem because it is constructed only based on CLC-PKC. Besides, it is improved in efficiency because the bilinear pairing operations are not used and the number of applied scalar point multiplication on ECC operations is limited as small as possible. At the same time, it achieves the receiver anonymity. The proposed scheme can be applied to ad-hoc networks to ensure that communication is efficient and secure and the receiver's privacy is protected. However, our scheme's security is proved under the random oracle model, which is not universal in reality more or less. Hence, we will do further research on security under the standard model to make the multi-message and multi-receiver scheme more practical.

REFERENCES

[1] L. C. Ma, X. F. Liu, Q. Q. Pei, and Y. Xiang, "Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing," *IEEE Trans. Services Comput.*, to be published. doi: 10.1109/TSC.2018.2825986.

[2] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving efficient and secure data acquisition for cloud-supported Internet of things in smart grid," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1934–1944, Dec. 2017.

[3] M. Elhoseny, A. Abdelaziz, A. S. Salama, A. M. Riad, K. Muhammad, and A. K. Sangaiah, "A hybrid model of Internet of things and cloud computing to manage big data in health services applications," *Future Gener. Comput. Syst.*, vol. 86, pp. 1383–1394, Sep. 2018.

[4] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology—CRYPTO* (Lecture Notes Comput. Science), vol. 773. Berlin, Germany: Springer, 2001, pp. 480–491.

[5] L. Pang, L. Gao, H. Li, and Y. Wang, "Anonymous multi-receiver ID-based signcryption scheme," *IET Inf. Secur.*, vol. 9, no. 3, pp. 194–201, May 2015.

[6] Z. Guan *et al.*, "APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *J. Netw. Comput. Appl.*, vol. 125, pp. 82–92, Jan. 2019.

[7] Z. Guan, Y. Zhang, L. Zhu, L. Wu, and S. Yu, "EFFECT: An efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid," *Sci. China Inf. Sci.*, vol. 62, no. 3, Mar. 2019, Art. no. 32103.

[8] V. Hindumathi and K. R. L. Reddy, "Adaptive priority-based fair-resource allocation for MIMO-OFDM multicast networks," *Int. J. Netw. Virtual Organisations*, vol. 20, no. 1, pp. 73–89, Jan. 2019.

[9] M. Seo and K. Kim, "Electronic funds transfer protocol using domain-verifiable signcryption scheme," in *Information Security and Cryptology—ICISC* (Lecture Notes Computer Science), vol. 1787. Berlin, Germany: Springer, 1999, pp. 269–277.

[10] R. Gao, J. Zeng, and L. Deng, "Efficient certificateless anonymous multi-receiver encryption scheme without bilinear parings," *Math. Prob. Eng.*, vol. 2018, Jul. 2018, Art. no. 1486437. doi: 10.1155/2018/1486437.

[11] D. H. Elkamchouchi, "A chaotic public key multi-message multi-recipients signcryption scheme (CPK-MM-MR-SS)," in *Proc. 12th World Multi-Conf. Systemics, Cybern. Inform./14th Int. Conf. Inf. Syst. Anal. Synth.*, Orlando, FL, USA, 2008, pp. 30–34.

[12] M. E. Hassan and A. A. H. Esam, "An efficient public key multi-messages multi-recipients elliptic curve signcryption (PK-MM-ECS) scheme," in *Proc. Nat. Radio Sci. Conf.*, Tanta, Egypt, Mar. 2008, pp. 1–10.

[13] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Gener. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.

[14] Y. Han and X. Gui, "Multi-recipient signcryption for secure group communication," in *Proc. 4th IEEE Conf. Ind. Electron. Appl.*, Xi'an, China, May 2009, pp. 161–165.

[15] Y. Han and X. Gui, "Adaptive secure multicast in wireless networks," *Int. J. Commun. Syst.*, vol. 22, no. 9, pp. 1213–1239, Sep. 2009.

[16] Y. Han, X. Gui, X. Yang, and H. Yang, "Parallel multi-recipient signcryption for multicast networks," in *Proc. 2nd Int. Workshop Educ. Technol. Comput. Sci.*, Wuhan, China, Mar. 2010, pp. 128–131.

[17] H. Li, Z. Han, L. Wang, and L. Pang, "Blind proxy re-signature scheme based on isomorphisms of polynomials," *IEEE Access*, vol. 6, pp. 53869–53881, 2018.

[18] A. Kumar and M. M. Ansari, "Multi message signcryption based on chaos with public verifiability," *Int. J. Sci. Technol. Res.*, vol. 2, no. 5, pp. 194–198, May 2013.

[19] M. Miao, J. Wang, S. Wen, and J. Ma, "Publicly verifiable database scheme with efficient keyword search," *Inf. Sci.*, vol. 475, pp. 18–28, Feb. 2019.

[20] A. I. U. N. Din, A. Waheed, and N. U. Amin, "An efficient multi message multi receiver signcryption scheme with forward secrecy on elliptic curves," IACR Cryptologic ePrint Arch., Las Vegas, NV, USA, Tech. Rep. 20150702:075730, 2015, p. 655. [Online]. Available: https://eprint.iacr.org/2015/655.pdf

[21] A. U. Rahman *et al.*, "A lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme based on hyper elliptic curve," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 5, pp. 160–167, 2018.

[22] J. Dai and Q. Zhou, "A PKI-based mechanism for secure and efficient access to outsourced data," in *Proc. Int. Conf. Netw. Digit. Soc.*, Wenzhou, China, May 2010, pp. 640–643.

[23] J. Qiu, J. Bai, X.-C. Song, and S.-M. Hou, "Secure and efficient multi-message and multi-receiver ID-based signcryption for rekeying in ad hoc networks," *J. Chongqing Univ.*, vol. 12, no. 2, pp. 91–96, Jun. 2013.

[24] C. Meshram, S. G. Meshram, and C.-C. Lee, "Constructing provably secure ID-based beta cryptographic scheme in random oracle," *Int. J. Netw. Secur.*, vol. 20, no. 3, pp. 568–574, May 2018.

[25] C. Wang, C. Liu, Y. Li, H. Qiao, and L. Chen, "Multi-message and multi-receiver heterogeneous signcryption scheme for ad-hoc networks," *Inf. Secur. J., A Global Perspective*, vol. 26, no. 3, pp. 136–152, May 2017.

[26] R. Boussada, M. E. Elhdhili, and L. A. Saidane, "Toward privacy preserving in IoT e-health systems: A key escrow identity-based encryption scheme," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2018, pp. 1–7.

[27] S. Niu, Z. Li, and C. Wang, "Privacy-preserving multi-party aggregate signcryption for heterogeneous systems," in *Cloud Computing and Security* (Lecture Notes Computer Science), vol. 10603. Berlin, Germany: Springer, 2017, pp. 216–229.

[28] S. Niu, L. Niu, X. Yang, C. Wang, and X. Jia, "Heterogeneous hybrid signcryption for multi-message and multi-receiver," *PLoS ONE*, vol. 12, no. 9, Sep. 2017, Art. no. e0184407.

[29] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Adv. Cryptology—ASIACRYPT* (Lecture Notes Computer Science), vol. 2894. Berlin, Germany: Springer, 2003, pp. 452–473.

[30] S. H. Islam, M. K. Khan, and A. M. Al-Khouri, "Anonymous and provably secure certificateless multireceiver encryption without bilinear pairing," *Secur. Commun. Netw.*, vol. 8, no. 13, pp. 2214–2231, Sep. 2015.

[31] Y.-H. Hung, S.-S. Huang, Y.-M. Tseng, and T.-T. Tsai, "Efficient anonymous multireceiver certificateless encryption," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2602–2613, Dec. 2017.

[32] L. Pang, M. Kou, M. Wei, and H. Li, "Efficient anonymous certificateless multi-receiver signcryption scheme without bilinear pairings," *IEEE Access*, vol. 6, pp. 78123–78135, 2018. doi: 10.1109/ACCESS.2018.2884798.

[33] H. Li and L. Pang, "Cryptanalysis of Wang et al.'s improved anonymous multi-receiver identity-based encryption scheme," *IET Inf. Secur.*, vol. 8, no. 1, pp. 8–11, Jan. 2014.

[34] S. S. D. Selvi, S. S. Vivek, D. Shukla, and P. R. Chandrasekaran, "Efficient and provably secure certificateless multi-receiver signcryption," in *Provable Security* (Lecture Notes Comput. Science), vol. 5324. Berlin, Germany: Springer, 2008, pp. 52–67.

LIAOJUN PANG (M'09) was born in 1978. He received the bachelor's and master's degrees in computer science and technology and the Ph.D. degree in cryptography from Xidian University, China, in 2000, 2003, and 2006, respectively, where he is currently a Full Professor with the State Key Laboratory of Integrated Services Networks, School of Life Science and Technology. He is also a Visiting Scholar with the Department of Computer Science, Wayne State University, USA. His research interests include the Internet security, cryptography, secure mobile agent systems, and e-commerce security technology.

MENGMENG WEI was born in 1993. She is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Services Networks, School of Life Science and Technology, Xidian University, China. Her research interests include cryptography and information theory.

HUIXIAN LI was born in 1977. She received the Ph.D. degree in cryptography from the Dalian University of Technology. She is currently an Associate Professor with the School of Computer Science and Engineering, Northwestern Polytechnical University. She is also a Visiting Scholar with the Department of Computer Science, Wayne State University, USA. Her research interests include information security, cryptography, and security technologies for mobile health care systems.