IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 4, 2019, accepted June 20, 2019, date of publication June 24, 2019, date of current version July 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924654

Anonymous Certificateless Multi-Receiver Signcryption Scheme Without Secure Channel

LIAOJUN PANG[1,2], (Member, IEEE), MAN KOU[1], 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 Basic Research Plan in Shaanxi Province of China under Grant 2019JM-129 and Grant 2018JM6064, and in part by the National Cryptography Development Fund under Grant MMJJ20170208.

ABSTRACT The certificateless multi-receiver signcryption scheme provides the sender with the ability to send the same message to multiple authorized receivers contemporaneously, and at the same time, it can avoid the key escrow problem in the existing identity-based multi-receiver signcryption schemes, which makes it to get great attention in the field of one-to-many communication. However, in the existing certificateless multi-receiver signcryption schemes, a secure channel is essential for their key extract algorithm, which brings some troubles in practical applications. On one hand, the security of the partial private key depends on the secure channel. Once the secure channel is broken by an attacker, the user's partial private key may be leaked. On the other hand, maintaining the secure channel increases the economic cost and implementation complexity of the application systems. Motivated by these concerns, we propose a new anonymous certificateless multi-receiver signcryption scheme, in which the key generation center only utilizes a public channel to send the pseudo partial private key to the user during the key extract algorithm, and the designated user can work out the real partial private key from the pseudo partial private key while others cannot. The avoidance of the secure channel improves the security of the proposed scheme and makes the communication system much lighter.

INDEX TERMS Cryptography, certificateless public key cryptography, multi-receiver signcryption, one-to-many communication, key escrow problem, receiver anonymity.

I. INTRODUCTION

Secure multicast [1], which means that the sender can send the same message to multiple receivers securely and simultaneously, provides an efficient communication mechanism for one-to-many communication. As a way to implement the secure multicast, multi-receiver encryption [2] enables the sender to encrypt the plaintext message for multiple receivers in one logical step, and each authorized receiver can independently decrypt the ciphertext correctly. However, the multi-receiver encryption scheme only provides message confidentiality, but does not provide message source verifiability, which limits its application scenarios. As an extension of multi-receiver encryption, multi-receiver signcryption [3] ensures the message confidentiality and provides message source verifiability at one time. With the increasing demand for distributed communication, more and more one-to-many application systems have emerged, such as pay-TV program, remote education and network conference [4]. In this case, the research on multi-receiver encryption/signcryption schemes [5], [6] has become a hotspot in the field of information security.

Beak *et al.* [7] proposed the first identity-based multi-receiver encryption (IBME) scheme, which needs only one pairing computation to encrypt the message for multiple receivers and has higher computational efficiency compared with the traditional one-to-many communication [8]. On its heels, several excellent IBME schemes [9]–[11]

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

were proposed. Combining the idea of signcryption [12], Duan and Cao [3] proposed the first identity-based multi-receiver signcryption (IBMS) scheme and gave the unforgeability security model in their paper. Then, some IBMS schemes [13]–[16] were put forward one by one. In recent years, privacy leakage incidents occur frequently [17], and people's awareness of privacy protection has gradually increased. People do not want it to be known by others which websites they visited or which TV programs they watched [18]. With the emphasis on privacy protection, receiver anonymity was introduced into the IBME/IBMS scheme. Many researchers tried to achieve receiver anonymity in multi-receiver encryption/signcryption schemes by using different methods.

Fan *et al*. [19] put forward the first anonymous IBME scheme, attempting to hide the information of authorized receivers in a Lagrange interpolating polynomial to avoid the disclosure of receivers. Unfortunately, Wang *et al*. [20] demonstrated that this approach cannot truly achieve receiver anonymity because any authorized receiver can judge whether another person is an authorized receiver. Meanwhile, Wang *et al*. [20] proposed an improved IBME scheme to ameliorate receiver anonymity. Regrettably, Li and Pang [21] proved that in Wang *et al*.'s scheme, authorized receivers were still able to determine the identities of other authorized receivers. Later, Tseng *et al*. [22] proposed an anonymous IBME scheme, which uses a modular polynomial to mix and hide the information of authorized receivers, and they re-defined the adversarial model of receiver anonymity under multi-receiver setting because the previous receiver anonymity security model is suitable for single receiver setting. Inspired by this method, Fan and Tseng [23] proposed another anonymous IBME scheme with chosen ciphertext attack (CCA) security. Their scheme provides authentication function for the receivers, but it uses too many bilinear parings operations, resulting in its low efficiency. In 2016, to perfect privacy protection mechanism, Pang *et al*. [24] proposed a completely anonymous IBMS scheme, which offers both receiver anonymity and sender anonymity.

However, all the schemes mentioned above have the key escrow problem, because they are designed based on the identity-based cryptosystem (IBC) [25]. That is to say, in these schemes, the key generation center (KGC) holds the private keys of all users, so it could peek at all users' communication information and could disguise himself as any user to sign a message. To solve the key escrow problem, Al-Riyami and Paterson [26] proposed the certificateless public key cryptography (CL-PKC), which attracts the attention of many scholars and institutions rapidly, and based on Al-Riyami *et al*.'s thought, many certificateless encryption/signcryption schemes [27]–[30] were proposed successively. Selvi *et al*. [31] proposed the first certificateless multi-receiver signcryption (CLMS) scheme, and defined the message confidentiality security model and unforgeability security model of the CLMS scheme. However, Selvi *et al*.'s scheme does not satisfy the message confidentiality under external attacks, so they proposed an improved CLMS scheme [32]. Unfortunately, Miao *et al*. [33] proved that the improved CLMS scheme [32] cannot meet the message confidentiality under external attacks either. In addition, both of Selvi *et al*.'s schemes [31], [32] do not consider receiver anonymity, and they are inefficient during signcryption process because too many bilinear pairings are used.

To reduce the computational complexity, Islam *et al*. [34] proposed a new anonymous certificateless multi-receiver encryption (CLME) scheme, which uses scalar point multiplications on elliptic curve cryptography instead of bilinear pairings and probabilistic map-to-point functions. Besides, Islam *et al*.'s scheme achieves receiver anonymity by using modular polynomial put forward by Tseng *et al*., and defines the receiver anonymity security model of the CLME scheme. However, Hung *et al*. [35] pointed out that Islam *et al*.'s scheme is not suitable for mobile devices since its computation cost of the encryption operation is quadric with the receivers' number, and at the same time, Hung *et al*. [35] proposed another anonymous CLME scheme whose computation cost of the encryption operation is linear with the receivers' number. Regrettably, Hung *et al*.'s scheme uses too many bilinear pairings operations, so it is still low in efficiency. Based on Hung *et al*. scheme, He *et al*. [36] proposed an anonymous CLME scheme, which does not need bilinear pairing operations and thus improves the efficiency in some degree. The three schemes [34]–[36] mentioned above achieve receiver anonymity, but they do not provide source verifiability.

Later, Tseng and Fan [37] proposed a lightweight CLME scheme, which has high computational efficiency and is suitable for device to device communication on the Internet of Things application. Their scheme provides the function of mutual authentication between the receiver and the sender, but it directly puts the list of authorized receivers in ciphertext, exposing the privacy of authorized receivers. In 2018, Win *et al*. [38] proposed an anonymous CLME scheme with CCA secure both in message confidentiality and receiver anonymity, and the scheme provides receivers with the function of verifying the sender. However, a large number of bilinear pairings used in Tseng *et al*.'s scheme results in its low computational efficiency. At the same year, Pang *et al*. [39] proposed an anonymous CLMS scheme, which is more efficient than schemes [37], [38] and offers receiver anonymity and source verifiability. Besides the aforementioned schemes, there are some other outstanding CLME/CLMS schemes [40]–[43] with various properties proposed for different application scenarios in recent years.

In all of the CLME/CLMS schemes mentioned above, the user's private key consists of two parts. One is the secret value chosen by the user himself, which is not known by anyone except the user himself, and the other part is the private key generated by KGC, which is usually sent to the user through a secure channel. It should be noted that the usage of the secure channel makes the privacy of the partial

TABLE 1. Notations.

Name	Meaning
IBC	Identity-based cryptography
IBME	Identity-based multi-receiver encryption
IBMS	Identity-based multi-receiver signcryption
CLMS	Certificateless multi-receiver signcryption
CLME	Certificateless multi-receiver encryption
CL-PKC	Certificateless public key cryptography
CDHP	Computational Diffie-Hellman Problem
ECDLP	Elliptic Curve Discrete Logarithm Problem
KGC	Key generation center
G_p	The addition cycle group of points on elliptic curve
p	Large prime integer
pk_i	Public key of the user ID_i
P	Generator of G_p
sk_i	Private key of he user i, i represents the user's identity
Z_p^*	Non-zero multiplicative group with large prime p

private key depend on the secure channel. Once the secure channel is controlled by an attacker, the user's partial private key is likely to be leaked, which is a terrible security problem for both cryptography and communication systems. In addition, maintaining the secure channel increases the complexity of the communication system and requires additional cost. Improving the security while minimizing the system as much as possible is the pursuit of every system designer [44]. Therefore, it is necessary to propose a certificateless multi-receiver signcryption scheme which does not need any secure channel.

Motivated by these concerns, we proposed a new anonymous certificateless multi-receiver signcryption scheme in this paper. When executing the key extract algorithm in the proposed scheme, KGC and the user transmit all information, including the pseudo partial private key generated by KGC for the user, to each other through public channels. It is easy for the user to extract the real partial private key from the pseudo partial private key, but impossible for the attacker. The elimination of the secure channel brings two benefits, that is, the security of the partial private key is improved and the complexity of the communication system is reduced. In addition, it is proved that the proposed scheme meets message confidentiality, unforgeability and receiver anonymity under the random oracle model.

The rest of this paper is organized as follows: The related hard problems, algorithm model and security models of the proposed scheme are given in Section II. In Section III, the elaboration of the proposed scheme is given. Section IV makes an analysis of the correctness and the security about the proposed scheme. Then, the comparisons between the proposed scheme and the existing CLME/CLMS schemes in terms of efficiency and functions are given in Section V. Finally, the conclusion about this paper is made in Section VI.

In order to facilitate reading and understanding, notations used in this paper are listed in Table 1:

II. PRELIMINARIES

In this section, we shall introduce the hard problems, algorithm model and security models related to the proposed scheme.

A. HARD PROBLEMS

We define that G_p is an additive cyclic group based on a large prime number p, the point P is a generator of G_p and Z_p^* is a nonzero multiplicative group based on the large prime number p. Computational Diffie-Hellman Problem (CDHP) and Elliptic Curve Discrete Logarithm Problem (ECDLP) will be given as follows:

1) **CDHP**: Given P, aP and $bP \in G_p$, where $a, b \in Z_p^*$, computing $abP \in G_p$ is called CDHP.

Definition 1: The probability advantage that CDHP can be solved by a probabilistic polynomial time (PPT) algorithm Ω is defined as

$$\mathrm{Adv}_{\Omega}^{\mathrm{CDHP}} = \Pr[\Omega(P, aP, bP) = abP].$$

CDHP *assumption*: For any PPT algorithm Ω, $\mathrm{Adv}_{\Omega}^{\mathrm{CDHP}}$ is negligible.

2) **ECDLP:** Given P and $xP \in G_p$, where $x \in Z_p^*$, computing the integer x is called ECDLP.

Definition 2: The probability advantage that ECDLP can be solved by any PPT algorithm Ω is defined as

$$\mathrm{Adv}_{\Omega}^{\mathrm{ECDLP}} = \Pr[\Omega(P, xP) = x].$$

ECDLP *assumption*: For any PPT algorithm Ω, $\mathrm{Adv}_{\Omega}^{\mathrm{ECDLP}}$ is negligible.

B. ALGORITHM MODEL

Definition 3: The algorithm model of the proposed scheme, consisting of *Setup*, *Set-Secret-Value*, *Extract-Partial-Private-key*, *Set-Private-Key*, *Set-Public-Key*, *Signcryption* and *De-Signcryption*, is shown as follows:

***Setup*:** With the system security parameter λ as input, KGC runs this algorithm to generate the system's public parameters *Params* and the system master key s. Then, KGC publicizes *Params* and keeps s secret.

***Set-Secret-Value*:** With the user's identity information ID as input, the user runs this algorithm to get his/her own secret value x_{ID} and secret value parameter X_{ID}.

Extract-Partial-Private-Key: With s, *Params* and ID as input, KGC runs this algorithm to get the user's pseudo partial private key u_{ID} and the public key generation parameters D_{ID}.

***Set-Private-Key*:** With *Params*, ID, X_{ID}, D_{ID}, u_{ID} and x_{ID} as input, the user runs this algorithm to get his/her own private key SK_{ID}.

***Set-Public-Key*:** With X_{ID} and D_{ID} as input, the user runs this algorithm to get his/her own public key PK_{ID}.

***Signcryption*:** With the plaintext message M, the sender's private key SK_S, the sender's identity information ID_S, the authorized receivers' public key PK_i $(1 \leq i \leq n)$ and *Params* as input, the sender S runs this algorithm to generate the ciphertext c.

***De-Signcryption*:** With the ciphertext c, the authorized receiver's private key SK_i, the sender's public key PK_S and *Params* as input, each receiver runs this algorithm to get the plaintext message M.

C. SECURITY MODELS

The security models of the proposed scheme include message confidentiality, unforgeability and receiver anonymity. There are two types of adversaries called Type I adversary ($\mathcal{A}_I$) and Type II adversary ($\mathcal{A}_{II}$) [26] respectively in every security model. $\mathcal{A}_I$ means a malicious adversary who has ability to replace the user's public key, but does not know the system master key s, while $\mathcal{A}_{II}$ means an honest-but-curious KGC who knows the system master key s, but is not allowed to replace the user's public key. The specific security models under different adversaries are shown as follows:

1) MESSAGE CONFIDENTIALITY

The message confidentiality of the proposed scheme is called the indistinguishability of certificateless signcryption against selective multi-receiver chosen ciphertext attack (IND-CLMS-CCA) [34]. We define the following two games called *Game* 1 and *Game* 2 to describe IND-CLMS-CCA against the adversary $\mathcal{A}_I$ and $\mathcal{A}_{II}$, respectively.

***Game*1** (IND-CLMS-CCA-I): This game is the interaction between the adversary $\mathcal{A}_I$ and the challenger $\mathcal{C}$ under IND-CLMS-CCA. Ω is defined as a certificateless anonymous multi-receiver signcryption algorithm. The specific steps are shown as follows:

***Setup*:** $\mathcal{C}$ runs this algorithm to generate the system master key s and the system's public parameters *Params*, and then keeps s secret and sends *Params* to $\mathcal{A}_I$. After receiving *Params*, $\mathcal{A}_I$ selects a set of target multiple identities $L^* = \{ID_1^*, ID_2^*, \ldots, ID_n^*\}$, where n is a positive integer, and then sends L^* to $\mathcal{C}$.

***Hash query*:** $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of queries on hash functions used in the scheme Ω, and $\mathcal{C}$ responds the corresponding hash values to $\mathcal{A}_I$.

***Phase* 1:** $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of queries, then $\mathcal{C}$ responds as follows:

Set-Secret-Value query: When $\mathcal{A}_I$ asks for the secret value of ID, where $ID \notin L^*$, $\mathcal{C}$ runs *Set-Secret-Value* algorithm and returns x_{ID} to $\mathcal{A}_I$.

Extract-Partial-Private-Key query: When $\mathcal{A}_I$ asks for the partial private key of ID, where $ID \notin L^*$, $\mathcal{C}$ runs *Extract-Partial-Private-Key* algorithm and returns y_{ID} to $\mathcal{A}_I$.

Set-Public-Key query: When $\mathcal{A}_I$ asks for the public key of ID, $\mathcal{C}$ runs *Set-Public-Key* algorithm and returns PK_{ID} to $\mathcal{A}_I$.

Public-Key-Replacement query: When $\mathcal{A}_I$ asks $\mathcal{C}$ to replace PK_{ID} of ID with PK'_{ID} chosen by him, $\mathcal{C}$ saves PK'_{ID} as the new public key of ID.

Signcryption query: When $\mathcal{A}_I$ asks $\mathcal{C}$ to signcrypt a plaintext M, $\mathcal{C}$ runs *Signcryption* algorithm and returns c to $\mathcal{A}_I$.

De-Signcryption query: When $\mathcal{A}_I$ asks $\mathcal{C}$ to designcrypt the ciphertext c chosen by him, $\mathcal{C}$ runs *De-Signcryption* algorithm and returns M to $\mathcal{A}_I$.

***Challenge*:** $\mathcal{A}_I$ generates a pair of plaintext $< M_0, M_1 >$ with equal length and sends them to $\mathcal{C}$. $\mathcal{C}$ randomly selects a bit $\beta \in \{0, 1\}$ and computes the ciphertext c^* of M_β, and then returns c^* to $\mathcal{A}_I$.

***Phase* 2:** $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of queries as described in *Phase* 1, but $\mathcal{A}_I$ cannot perform *Set-Secret-Value query* and *Extract-Partial-Private-Key query* on the user whose public key has been replaced, and $\mathcal{A}_I$ cannot perform *De-Signcryption query* on the ciphertext c^*.

***Guess*:** $\mathcal{A}_I$ guesses a bit $\beta^* \in \{0, 1\}$. If $\beta^* = \beta$ holds, $\mathcal{A}_I$ wins *Game* 1. Otherwise, $\mathcal{A}_I$ fails. The advantage of $\mathcal{A}_I$ to win *Game* 1 is defined as:

$$Adv_{\Omega}^{IND\text{-}CLMS\text{-}CCA\text{-}I}(\mathcal{A}_I) = |2\Pr[\beta^* = \beta] - 1|.$$

Definition 4: If for any $\mathcal{A}_I$ under IND-CLMS-CCA, the probability advantage of winning *Game* 1 within time τ meets $Adv_{\Omega}^{IND\text{-}CLMS\text{-}CCA\text{-}I}(\mathcal{A}_I) \leq \varepsilon$, the scheme Ω is said to be (τ, ε)-IND-CLMS-CCA-I secure, where τ is the polynomial time and ε is a negligible probability advantage.

***Game* 2** (IND-CLMS-CCA-II): This game is the interaction between the adversary $\mathcal{A}_{II}$ and the challenger $\mathcal{C}$ under IND-CLMS-CCA. Ω is defined as a certificateless anonymous multi-receiver signcryption algorithm. The specific steps are shown as follows:

***Setup*:** $\mathcal{C}$ runs this algorithm to generate the system master key s and the system's public parameters *Params*, and then sends s and *Params* to $\mathcal{A}_{II}$. After receiving s and *Params*, $\mathcal{A}_{II}$ chooses a set of target multiple identities $L^* = \{ID_1^*, ID_2^*, \ldots, ID_n^*\}$, where n is a positive integer, and then sends L^* to $\mathcal{C}$.

***Hash query*:** This step is the same as *Hash query* in *Game* 1.

***Phase* 1:** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of queries and $\mathcal{C}$ responds accordingly. Among these queries, *Set-Secret-Value query*,*Set-Public-Key query*, *Signcryption query* and *De-Signcryption query* are the same as corresponding queries in *Phase* 1 of *Game* 1. The different queries are shown as follows:

Extract-Partial-Private-Key query: When $\mathcal{A}_{II}$ asks for the partial private key of ID, $\mathcal{C}$ runs *Extract-Partial-Private-Key* algorithm and returns y_{ID} to $\mathcal{A}_{II}$.

Public-Key-Replacement query: When $\mathcal{A}_{II}$ asks $\mathcal{C}$ to replace PK_{ID} of ID with PK'_{ID} chosen by him, where $ID \notin L^*$, $\mathcal{C}$ saves PK'_{ID} as the new public key of ID.

***Challenge*:** $\mathcal{A}_{II}$ generates a pair of plaintext $< M_0, M_1 >$ with equal length and sends them to $\mathcal{C}$. $\mathcal{C}$ randomly selects a bit $\beta \in \{0,1\}$ and computes the ciphertext c^* of M_β, and then returns c^* to $\mathcal{A}_{II}$.

***Phase* 2:** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of queries as described in *Phase* 1, but $\mathcal{A}_{II}$ cannot perform *Set-Secret-Value query* and *Extract-Partial-Private-Key query* on the user whose public key has been replaced, and $\mathcal{A}_{II}$ cannot perform *De-Signcryption query* on the ciphertext c^*.

***Guess*:** $\mathcal{A}_{II}$ guesses a bit $\beta^* \in \{0,1\}$. If $\beta^* = \beta$ holds, $\mathcal{A}_{II}$ wins *Game* 2. Otherwise, $\mathcal{A}_{II}$ fails. The advantage of $\mathcal{A}_{II}$ to win *Game* 2 is defined as:

$$\mathrm{Adv}_{\Omega}^{\text{IND-CLMS-CCA-II}}(\mathcal{A}_{II}) = |2\Pr[\beta^* = \beta] - 1|.$$

Definition 5 : If for any $\mathcal{A}_{II}$ under IND-CLMS-CCA, the probability advantage of winning *Game* 2 within time τ meets $\mathrm{Adv}_{\Omega}^{\text{IND-CLMS-CCA-II}}(\mathcal{A}_{II}) \leq \varepsilon$, the scheme Ω is said to be (τ, ε)-IND-CLMS-CCA-II secure, where τ is the polynomial time and ε is a negligible probability advantage.

2) UNFORGEABILITY

The unforgeability model of the proposed scheme is called the strong existential unforgeability of certificateless signcryption against selective multi-receiver chosen plaintext attack (sEUF-CLMS-CPA) [31]. We define the following two games called *Game*3 and *Game*4 to describe sEUF-CLMS-CPA against the adversary $\mathcal{A}_I$ and $\mathcal{A}_{II}$, respectively.

***Game* 3** (sEUF-CLMS-CPA-I): This game is the interaction between adversary $\mathcal{A}_I$ and challenger $\mathcal{C}$ under sEUF-CLMS-CPA. Ω is defined as a certificateless anonymous multi-receiver signcryption algorithm. The specific steps are shown as follows:

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

***Hash query*:** This step is the same as *Hash query* in *Game* 1.

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

***Forgery*:** With a plaintext M, a sender $\mathrm{ID}_S \in L^*$ and a group of receivers identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$, $\mathcal{A}_I$ forges a ciphertext c^*. If the ciphertext c^* can be decrypted correctly by all receivers in L, $\mathcal{A}_I$ wins *Game* 3. Otherwise, $\mathcal{A}_I$ fails. There is a restriction that ciphertext c^* cannot be generated by the *Signcryption query*.

Definition 6: If for any $\mathcal{A}_I$ under sEUF-CLMS-CPA, the probability advantage of winning *Game* 3 within time τ meets $\mathrm{Adv}_{\Omega}^{\text{sEUF-CLMS-CPA-I}}(\mathcal{A}_I) \leq \varepsilon$, the scheme Ω is said to be (τ, ε)-sEUF-CLMS-CPA-I secure, where τ is the polynomial time and ε is a negligible probability advantage.

***Game* 4** (sEUF-CLMS-CPA-II): This game is the interaction between adversary $\mathcal{A}_{II}$ and the challenger $\mathcal{C}$ under sEUF-CLMS-CPA. Ω is defined as a certificateless anonymous multi-receiver signcryption algorithm. The specific steps are shown as follows:

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

***Hash query*:** This step is the same as *Hash query* in *Game* 1.

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

***Forgery*:** With a plaintext M, a sender $\mathrm{ID}_S \in L^*$ and a group of receivers identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$, $\mathcal{A}_{II}$ forges a ciphertext c^*. If the ciphertext c^* can be decrypted correctly by all receivers in L, $\mathcal{A}_{II}$ wins *Game* 4. Otherwise, $\mathcal{A}_{II}$ fails. There is a restriction that ciphertext c^* cannot be generated by the *Signcryption query*.

Definition 7: If for any $\mathcal{A}_{II}$ under sEUF-CLMS-CPA, the probability advantage of winning *Game* 4 within time τ meets $\mathrm{Adv}_{\Omega}^{\text{sEUF-CLMS-CPA-II}}(\mathcal{A}_{II}) \leq \varepsilon$, the scheme Ω is said to be (τ, ε)-sEUF-CLMS-CPA-II secure, where τ is the polynomial time and ε is a negligible probability advantage.

3) RECEIVER ANONYMITY

The receiver anonymity is called the anonymous indistinguishability of certificateless signcryption against selective multi-receiver chosen ciphertext attack (ANON-IND-CLMS-CCA) [34]. We define the following two games called *Game*5 and *Game*6 to achieve ANON-IND-CLMS-CCA against the adversary $\mathcal{A}_I$ and $\mathcal{A}_{II}$, respectively.

***Game*5** (ANON-IND-CLMS-CCA-I): This game is the interaction between adversary $\mathcal{A}_I$ and challenger $\mathcal{C}$ under ANON-IND-CLMS-CCA. Ω is defined as a certificateless anonymous multi-receiver signcryption algorithm. The specific steps are shown as follows:

***Setup*:** $\mathcal{C}$ runs this algorithm to generate the system master key s and the system's public parameters *Params*, and then keeps s secret and sends *Params* to $\mathcal{A}_I$. After receiving *Params*, $\mathcal{A}_I$ selects a pair of target multiple identities $L^* = \{\mathrm{ID}_0^*, \mathrm{ID}_1^*\}$, and then sends L^* to $\mathcal{C}$.

***Hash query*:** This step is the same as *Hash query* in *Game* 1.

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

***Challenge*:** $\mathcal{A}_I$ chooses a plaintext M and a set of target identities $L = \{\mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, where n is a positive integer, and then sends M and L to $\mathcal{C}$. $\mathcal{C}$ randomly chooses a bit $e \in \{0, 1\}$ and computes the ciphertext c^* with a group of new target identities $L' = \{\mathrm{ID}_e^*, \mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, and then returns the ciphertext c^* to $\mathcal{A}_I$.

***Phase* 2:** This step is the same as *Phase* 2 in *Game* 1.

***Guess*:** $\mathcal{A}_I$ guesses a bit $e^* \in \{0, 1\}$. If $e^* = e$ holds, $\mathcal{A}_I$ wins *Game* 5. Otherwise, $\mathcal{A}_I$ fails. The advantage of $\mathcal{A}_I$ to win *Game* 5 is defined as:

$$\mathrm{Adv}_{\Omega}^{\text{ANON-IND-CLMS-CCA-I}}(\mathcal{A}_I) = |2\Pr[e^* = e] - 1|.$$

Definition 8: If for any $\mathcal{A}_I$ under ANON-IND-CLMS-CCA, the probability advantage of winning *Game* 5 within time τ meets $\mathrm{Adv}_{\Omega}^{\text{ANON-IND-CLMS-CCA-I}}(\mathcal{A}_I) \leq \varepsilon$, the scheme Ω is said to be (τ, ε)-ANON-IND-CLMS-CCA-I secure, where τ is the polynomial time and ε is a negligible probability advantage.

***Game*6** (ANON-IND-CLMS-CCA-II): This game is the interaction between adversary $\mathcal{A}_{II}$ and challenger $\mathcal{C}$ under ANON-IND-CLMS-CCA. Ω is defined as a certificateless anonymous multi-receiver signcryption algorithm. The specific steps are shown as follows:

***Setup*:** $\mathcal{C}$ runs this algorithm to generate the system master key s and the system's public parameters *Params*, and then sends s and *Params* to $\mathcal{A}_{II}$. After receiving s and *Params*, $\mathcal{A}_{II}$ chooses a set of target multiple identities $L^* = \{\mathrm{ID}_0^*, \mathrm{ID}_1^*\}$, and then sends L^* to $\mathcal{C}$.

***Hash query*:** This step is the same as *Hash query* in *Game* 1.

***Phase* 1**: This step is the same as *Phase*1 in *Game* 2.

***Challenge*:**$\mathcal{A}_{II}$ chooses a plaintext M and a set of target identities $L = \{\mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, where n is a positive integer, and then sends M and L to $\mathcal{C}$. $\mathcal{C}$ randomly chooses a bit $e \in \{0, 1\}$ and computes the ciphertext c^* with a group of new target identities $L' = \{\mathrm{ID}_e^*, \mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, and then returns the ciphertext c^* to $\mathcal{A}_{II}$.

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

***Guess*:** $\mathcal{A}_{II}$ guesses a bit $e^* \in \{0, 1\}$. If $e^* = e$ holds, $\mathcal{A}_{II}$ wins *Game* 6. Otherwise, $\mathcal{A}_{II}$ fails. The advantage of $\mathcal{A}_{II}$ to win *Game* 6 is defined as:

$$\mathrm{Adv}_{\Omega}^{\text{ANON-IND-CLMS-CCA-II}}(\mathcal{A}_{II}) = |2\Pr[e^* = e] - 1|.$$

Definition 9: If for any $\mathcal{A}_{II}$ under ANON-IND-CLMS-CCA, the probability advantage of winning *Game* 6 within time τ meets $\mathrm{Adv}_{\Omega}^{\text{ANON-IND-CLMS-CCA-II}}(\mathcal{A}_{II}) \leq \varepsilon$, the scheme Ω is said to be (τ, ε)-ANON-IND-CLMS-CCA-II secure, where τ is the polynomial time and ε is a negligible probability advantage.

III. THE PROPOSED SCHEME

The proposed scheme is composed of four algorithms, named *Setup algorithm*, *Key Extract algorithm*, *Signcryption algorithm* and *De-Signcryption algorithm*, shown as follows:

A. SETUP ALGORITHM

This algorithm is run by KGC to generate the system master key and the system's public parameters, and it is composed of the following five steps:

1) With the security parameter λ as input, KGC chooses a large prime number p, determines the finite field F_p with its order large prime number p, selects the secure elliptic curve E_p on the finite field F_p, determines the addition cycle group G_p on the elliptic curve E_p, and selects a generator P on the addition cycle group G_p;

2) Randomly choose a positive integer $s \in Z_p^*$ as the system master key and keep it secret, and then compute the system public key $P_{\mathrm{pub}} = sP$;

3) Select four secure one-way hash functions, as follows:

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

4) Select a secure symmetric encryption algorithm E_k and the corresponding decryption algorithm D_k from the existing symmetric encryption algorithm, such as AES, where k is the symmetric key;

5) Construct and publish the system parameters $Params = < p, F_p, E_p, G_p, P, P_{\mathrm{pub}}, E_k, D_k, H_0, H_1, H_2, H_3 >$, and keep the system master key s secret.

B. KEY EXTRACT ALGORITHM

This algorithm is run by KGC and the user together to extract the user's private key and public key. It is composed of the following four steps:

1) *Set-Secret-Value*

The user ID_i randomly chooses an integer $x_i \in Z_p^*$ as his/her secret value and keeps x_i secret, and then computes $X_i = x_iP$. After that, he/she sends X_i and ID_i to KGC through a public channel.

2) *Extract-Partial-Private-Key*

Upon receiving X_i and ID_i from the user, KGC randomly chooses an integer $d_i \in Z_p^*$, and then computes $D_i = d_iP$ and $u_i = d_i + sH_0(\mathrm{ID}_i, X_i + D_i) + H_0(\mathrm{ID}_i, sX_i) \pmod{p}$, where u_i is the pseudo partial private key of the user. After that, KGC sends u_i and D_i to the user through a public channel.

3) *Set-Private-Key*

Upon receiving u_i and D_i from KGC, the user verifies whether the equation $u_iP = D_i + H_0(\mathrm{ID}_i, X_i + D_i)P_{\mathrm{pub}} + H_0(\mathrm{ID}_i, x_iP_{\mathrm{pub}})P$ holds. If yes, the user extracts his/her partial private key $y_i = u_i\text{-}H_0(\mathrm{ID}_i, x_iP_{\mathrm{pub}})$ and computes his/her private key $\mathrm{SK}_i = x_i + y_i$; otherwise, the user rejects the u_i and D_i, exits the Key Extract algorithm and notifies KGC there is an error.

4) *Set-Public-Key*

(a) The user computes $\mathrm{PK}_i = X_i + D_i$ as his/her public key and sends PK_i to KGC through a public channel.

(b) Upon receiving PK_i from the user, KGC publishes the user's public key PK_i.

C. SIGNCRYPTION ALGORITHM

This algorithm is run by a sender S. Before signcryption, the sender S selects a group of users $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$ as authorized receivers who have extracted their own keys. It is composed of the following six steps:

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

2) Compute $K_i = r\,(\mathrm{PK}_i + H_0(\mathrm{ID}_i, \mathrm{PK}_i)P_{\mathrm{pub}})$ and $\alpha_i = H_1(\mathrm{ID}_i, R, K_i)$;

3) Randomly choose an integer $\theta \in Z_p^*$, and then compute the polynomial:

$$f(x) = \prod_{i=1}^{n} (x - \alpha_i) + \theta \pmod{p}$$
$$= x^n + a_{n-1}x^{n-1} + \ldots + a_1x + a_0, a_i \in Z_p^*;$$

4) Compute $k = H_2(\theta, R)$ and $Z = E_k(M||\ \mathrm{ID}_S)$;

5) Compute $h = H_3(M||\ \mathrm{ID}_S, R, \theta, a_n{-}1, \ldots, a_1, a_0)$ and $v = \mathrm{SK}_S + rh \pmod{p}$;

6) Generate the ciphertext $c =< R, Z, h, v, a_n{-}1, \ldots, a_1, a_0 >$, and then broadcast the ciphertext c to the receivers.

D. DE-SIGNCRYPTION ALGORITHM

This algorithm is run by every receiver R_i, but only authorized receivers can successfully execute *De-Signcryption* algorithm. It is composed of the following five steps:

1) Compute $K_i = \mathrm{SK}_i R$ and $\alpha_i = H_1(\mathrm{ID}_i, R, K_i)$;

2) Restore polynomial $f(x) = x^n + a_n - 1x^n - 1 + \ldots + a_1 x + a_0$ by using polynomial coefficients $< a_n - 1, \ldots, a_1, a_0 >$, and then compute $\theta = f(\alpha_i) \pmod p$;

3) Compute $k = H_2(\theta, R)$ and $M || \mathrm{ID}_S = D_k(Z)$;

4) Compute $h' = H_3(M || \mathrm{ID}_S, R, \theta, a_n - 1, \ldots, a_1, a_0)$, then verify whether the equation $h' = h$ holds. If yes, go to the next step; otherwise, the receiver R_i rejects M and exits the *De-Signcryption* algorithm;

5) With the public key PK_S of ID_S, verify whether the equation $vP = \mathrm{PK}_S + H_0(\mathrm{ID}_S, \mathrm{PK}_S)P_{\mathrm{pub}} + hR$ holds. If yes, the receiver R_i accepts M; otherwise, rejects it.

IV. CORRECTNESS AND SECURITY PROOFS

In this section, we will prove the correctness of the proposed scheme and give the formal proof of the security.

A. CORRECTNESS ANALYSIS

The correctness of the proposed scheme depends on the following two theorems.

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

Proof: The correctness of the user's pseudo partial private key verification is guaranteed by the establishment of the equation $u_i P = D_i + H_0(\mathrm{ID}_i, X_i + D_i)P_{\mathrm{pub}} + H_0(\mathrm{ID}_i, x_i P_{\mathrm{pub}})P$. The deduction that the equation holds is shown as follows:

$$\begin{aligned} u_i P &= (d_i + sH_0(\mathrm{ID}_i, X_i + D_i) + H_0(\mathrm{ID}_i, sX_i))P \\ &= d_i P + sH_0(\mathrm{ID}_i, X_i + D_i)P + H_0(\mathrm{ID}_i, sX_i)P \\ &= D_i + H_0(\mathrm{ID}_i, X_i + D_i)P_{\mathrm{pub}} + H_0(\mathrm{ID}_i, sX_i)P \end{aligned}$$

Through the above derivation, it can be seen that the equation $u_i P = D_i + H_0(\mathrm{ID}_i, X_i + D_i)P_{\mathrm{pub}} + H_0(\mathrm{ID}_i, x_i P_{\mathrm{pub}})P$ holds, so the *Theorem*1 is correct. ■

Theorem 2: The *De-Signcryption algorithm* is correct.

Proof : The correctness of *De-Signcryption algorithm* is guaranteed by the establishment of these two equations $h' = h$ and $vP = \mathrm{PK}_S + H_0(\mathrm{ID}_S, \mathrm{PK}_S)P_{\mathrm{pub}} + hR$. The deductions that these two equations hold are shown in following 1) and 2), respectively.

1) For every receiver R_i, with the ciphertext c^*, he/she can get $K_i = \mathrm{SK}_i R$ and $\alpha_i = H_1(\mathrm{ID}_i, R, K_i)$, and then he/she can compute $\theta = f(\alpha_i) \pmod p$ and $k = H_2(\theta, R)$. With the symmetric key k, the receiver can obtain $M || \mathrm{ID}_S = D_k(Z)$. Finally, he/she can compute $h' = H_3(M || \mathrm{ID}_S, R, \theta, a_n - 1, \ldots, a_1, a_0)$. So the equation $h' = h$ holds.

2) After decrypting out the sender's identity ID_S, the receiver can acquire the public key PK_S. With PK_S, the receiver can verify the validity of the signature, shown as follows:

$$\begin{aligned} vP &= (\mathrm{SK}_S + rh)P \\ &= (x_S + y_S + rh)P \\ &= (x_S + d_S + sH_0(\mathrm{ID}_S, X_S + D_S) + rh)P \\ &= X_S + D_S + H_0(\mathrm{ID}_S, X_S + D_S)P_{\mathrm{pub}} + hR \\ &= \mathrm{PK}_S + H_0(\mathrm{ID}_S, \mathrm{PK}_S)P_{\mathrm{pub}} + hR. \end{aligned}$$

That is to say, the equation $vP = \mathrm{PK}_S + H_0(\mathrm{ID}_S, \mathrm{PK}_S)P_{\mathrm{pub}} + hR$ holds.

Through the derivations of 1) and 2) above, it can be seen that these two equations $h' = h$ and $vP = \mathrm{PK}_S + H_0(\mathrm{ID}_S, \mathrm{PK}_S)P_{\mathrm{pub}} + hR$ hold. As a result, the *De-Signcryption algorithm* is correct.

B. SECURITY PROOFS

Based on security models described in Section 2, we give specific security proofs of the proposed scheme. The message confidentiality is dependent on the establishment of the *Theorem* 3 and *Theorem* 4, the unforgeability relies on the establishment of the *Theorem* 5 and *Theorem* 6, and the receiver anonymity depends on the establishment of the *Theorem* 7 and *Theorem* 8.

Theorem 3: Under IND-CLMS-CCA-I, if there is an adversary $\mathcal{A}_I$ who can win *Game* 1 in probability polynomial time τ with a non-negligible probability advantage ε ($\mathcal{A}_I$ can ask for at most q_i times *Hash queries* H_i $(i = 0, 1, 2, 3)$, q_c times *Create*(ID) *queries*, q_p times *Set-Public-Key queries*, q_r times *Public-Key-Replacement queries*, q_s times *Signcryption queries* and q_d times *De-Signcryption queries*.), the challenger $\mathcal{C}$ can solve CDHP by interacting with the adversary $\mathcal{A}_I$ in time

$$\tau' \leq \tau + (3q_c + 4q_d)\, T_m + T_i$$

with a non-negligible probability advantage

$$\varepsilon' \geq (1 - \frac{q_0 q_c}{p})(1 - \frac{q_1^2}{p})(1 - \frac{q_2^2}{p})(1 - \frac{q_3^2}{p})(1 - \frac{q_d}{p})\varepsilon$$

where T_m is the time spent for executing an elliptic curve scalar point multiplication operation and T_i is the time spent for executing a modular inversion operation.

Proof : Assume that within the polynomial time τ, the adversary $\mathcal{A}_I$ can attack the IND-CLMS-CCA-I of the proposed CLMS scheme with a non-negligible probability advantage ε, then there must be a challenger $\mathcal{C}$ who can solve the CDHP by interacting with $\mathcal{A}_I$; that is, for given $< P, aP, bP >$, $\mathcal{C}$ will output abP.

$\mathcal{C}$ maintains the following initial-empty lists in order to achieve the consistency between queries made by $\mathcal{A}_I$:

H_0 list L_0: This list includes the tuple $< \mathrm{ID}_j, \mathrm{PK}_j, l_j >$;

H_1 list L_1: This list includes the tuple $< \mathrm{ID}_j, R_j, K_j, \alpha_j >$;

H_2 list L_2: This list includes the tuple $< \mathrm{ID}_j, \theta_j, R_j, k_j >$;

H_3 list L_3: This list includes the tuple $< \mathrm{ID}_j, M_j || \mathrm{ID}_S, j, R_j, \theta_j, a_j, n - 1, \ldots, a_j, 1, a_j, 0, h_j >$;

List L_C: This list includes the tuple $< \mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j >$.

Setup: $\mathcal{C}$ runs this algorithm to generate the system master key $s = a \in Z_p^*$ and the system's public parameters $Params = < p, F_p, E_p, G_p, P, P_0 = aP, E_k, D_k, H_0, H_1, H_2, H_3 >$, and then keeps s secret and returns $Params$ to $\mathcal{A}_I$. After receiving $Params$, $\mathcal{A}_I$ selects a set of target multiple identities $L^* = \{\mathrm{ID}_1^*, \mathrm{ID}_2^*, \ldots, \mathrm{ID}_n^*\}$ and sends L^* to $\mathcal{C}$, where n is a positive integer.

Hash queries: $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of the following *Hash queries* and $\mathcal{C}$ responds accordingly as follows:

***H*₀-query:** With the tuple $<\mathrm{ID}_j, \mathrm{PK}_j>$ as input, $\mathcal{A}_I$ asks $\mathcal{C}$ for H_0 query. Upon receiving the query, $\mathcal{C}$ searches the list L_0 and responses l_j if the tuple $<\mathrm{ID}_j, \mathrm{PK}_j, l_j>$ is in the list L_0. Otherwise, $\mathcal{C}$ chooses $l_j \in Z_p^*$ and responds l_j to $\mathcal{A}_I$, and then inserts the tuple $<\mathrm{ID}_j, \mathrm{PK}_j, l_j>$ into L_0.

***H*₁-query:** With the tuple $<\mathrm{ID}_j, R_j, K_j>$ as input, $\mathcal{A}_I$ asks $\mathcal{C}$ for H_1 query. Upon receiving the query, $\mathcal{C}$ searches the list L_1 and responses α_j if the tuple $<\mathrm{ID}_j, R_j, K_j, \alpha_j>$ is in the list L_1. Otherwise, $\mathcal{C}$ chooses $\alpha_j \in Z_p^*$ and responds α_j to $\mathcal{A}_I$, and then inserts the tuple $<\mathrm{ID}_j, R_j, K_j, \alpha_j>$ into L_1.

***H*₂-query:** With the tuple $<\mathrm{ID}_j, \theta_j, R_j>$ as input, $\mathcal{A}_I$ asks $\mathcal{C}$ for H_2 query. Upon receiving the query, $\mathcal{C}$ searches the list L_2 and responses k_j if the tuple $<\mathrm{ID}_j, \theta_j, R_j, k_j>$ is in the list L_2. Otherwise, $\mathcal{C}$ chooses $k_j \in Z_p^*$ and responds k_j to $\mathcal{A}_I$, and then inserts the tuple $<\mathrm{ID}_j, \theta_j, R_j, k_j>$ into L_2.

***H*₃-query:** With the tuple $<\mathrm{ID}_j, M_j||\mathrm{ID}_S{,}j, R_j, \theta_j, a_j{,}n{-}1, \ldots, a_j, 1, a_j, 0>$ as input, $\mathcal{A}_I$ asks $\mathcal{C}$ for H_3 query. Upon receiving the query, $\mathcal{C}$ searches the list L_3 and responses h_j if the tuple $<\mathrm{ID}_j, M_j||\mathrm{ID}_S{,}j, R_j, \theta_j, a_j{,}n{-}1, \ldots, a_j, 1, a_j, 0, h_j>$ is in the list L_3. Otherwise, $\mathcal{C}$ chooses $h_j \in Z_p^*$ and responds h_j to $\mathcal{A}_I$, and then inserts the tuple $<\mathrm{ID}_j, M_j||\mathrm{ID}_S{,}j, R_j, \theta_j, a_j{,}n{-}1, \ldots, a_j, 1, a_j, 0, h_j>$ into L_3.

***Phase* 1:** $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of queries, then $\mathcal{C}$ responds as follows:

Create*(ID_j) *query: $\mathcal{A}_I$ asks $\mathcal{C}$ for a*Create*(ID_j) *query*. Upon receiving the query, $\mathcal{C}$ checks whether the tuple $<\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j>$ is in the list L_C. If yes, $\mathcal{C}$ keeps the tuple. Otherwise, $\mathcal{C}$ randomly chooses three integers $x_j, y_j, l_j \in Z_p^*$, sets $l_j = H_0(\mathrm{ID}_j, \mathrm{PK}_j)$, $D_j = y_jP\text{-}l_jP_{\mathrm{pub}}$, $X_j = x_jP$, and then performs as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i^*$ for $i \in \{1, 2, \ldots, n\}$, sets $\mathrm{SK}_j = \perp$ and $\mathrm{PK}_j = X_j + D_j$, and then updates the tuples $<\mathrm{ID}_j, (\perp, \perp), (X_j, D_j)>$ in the list L_C and $<\mathrm{ID}_j, \mathrm{PK}_j, l_j>$ in the list L_0.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$ for $i \in \{1, 2, \ldots, n\}$, sets $\mathrm{SK}_j = x_j + y_j$ and $\mathrm{PK}_j = X_j + D_j$, and then updates the tuples $<\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j>$ in the list L_C and $<\mathrm{ID}_j, \mathrm{PK}_j, l_j>$ in the list L_0.

Set-Secret-Value query: $\mathcal{A}_I$ asks $\mathcal{C}$ for a *Set-Secret-Value query* on the identity ID_j. Upon receiving the query, $\mathcal{C}$ checks if the tuple $<\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j>$ is in the list L_C. If yes, $\mathcal{C}$ returns x_j to $\mathcal{A}_I$; otherwise, $\mathcal{C}$ asks a *Creat*(ID_j) *query* to obtain the tuple $<\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j>$, and then returns x_j to $\mathcal{A}_I$.

Extract-Partial-Private-key query: $\mathcal{A}_I$ asks $\mathcal{C}$ for an *Extract-Partial-Private-key query* on the identity ID_j. Upon receiving the query, $\mathcal{C}$ responds as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$ for $i \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ stops the protocol execution.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$ for $i \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ checks if the tuple $<\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j>$ is in the list L_C. If yes, $\mathcal{C}$ returns y_j to $\mathcal{A}_I$; otherwise, $\mathcal{C}$ asks a *Creat*(ID_j) *query* to obtain the tuple $<\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j>$, and then returns y_j to $\mathcal{A}_I$.

Set-Public-key query: $\mathcal{A}_I$ asks $\mathcal{C}$ for a *Set-Public-key query* on the identity ID_j. Upon receiving the query, $\mathcal{C}$ checks if the tuple $<\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j>$ is in the list L_C. If yes, $\mathcal{C}$ returns PK_j to $\mathcal{A}_I$; otherwise, $\mathcal{C}$ asks a *Creat*(ID_j) *query* to obtain the tuple $<\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j>$, and then returns PK_j to $\mathcal{A}_I$.

Public-Key-Replacement query: If $\mathcal{A}_I$ asks $\mathcal{C}$ to replace PK_j of ID_j with PK_j' chosen by him, $\mathcal{C}$ looks into the list L_C for PK_j and updates PK_j with PK_j'.

Signcryption query: With a plaintext M, an identity ID_S and a group of receivers $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$, $\mathcal{A}_I$ asks $\mathcal{C}$ for a *Signcryption query*. Upon receiving the query, $\mathcal{C}$ judges whether the tuple $<\mathrm{ID}_S, (x_S, d_S), \mathrm{PK}_S>$ is in the list L_C. If yes, $\mathcal{C}$ does the following steps to generate ciphertext c^*; otherwise, $\mathcal{C}$ performs a *Creat*(ID_S) *query* to obtain the private key SK_S and the public key PK_S, then does the following steps to generate ciphertext c^*:

1) Randomly choose an integer $r \in Z_p^*$, then compute $R = rP$, $K_j = r(\mathrm{PK}_j + H_0(\mathrm{ID}_j, \mathrm{PK}_j)P_0)$ and $\alpha_j = H_0(\mathrm{ID}_j, K_j)$;

2) Randomly choose an integer $\theta \in Z_p^*$, and construct an n-order polynomial:

$$\begin{aligned} f(x) &= \prod_{j=1}^{n} \left(x - \alpha_j\right) + \theta \,(\mathrm{mod} p) \\ &= x^n + a_{n-1}x^{n-1} + \ldots + a_1x + a_0, a_i \in Z_p^*; \end{aligned}$$

3) Compute $k = H_1(\theta, R)$, $Z = E_k(M||\mathrm{ID}_S)$ and $h = H_2(M||\mathrm{ID}_S, R, \theta, a_n{-}1, \ldots, a_1, a_0)$;

4) Randomly choose an integer $v \in Z_p^*$;

5) Return the ciphertext $c^* =< R, Z, h, v, a_n{-}1, \ldots, a_1, a_0 >$ to $\mathcal{A}_I$.

De-Signcryption query: With identity ID_j for $j \in \{1, 2, \ldots, n\}$ and ciphertext $c_j =< R_j, Z_j, h_j, v_j, a_j{,}n{-}1, \ldots, a_j, 1, a_j, 0 >$, $\mathcal{A}_I$ asks $\mathcal{C}$ for a *De-Signcryption query*. Upon receiving the query, $\mathcal{C}$ does as follows:

1) Search the list L_3 for a tuple $<\mathrm{ID}_j, M_j||\mathrm{ID}_S{,}j, R_j, \theta_j, a_j{,}n{-}1, \ldots, a_j, 1, a_j, 0, h_j>$. If there is no such tuple, $\mathcal{C}$ outputs *failure* and *aborts* the game. Otherwise, $\mathcal{C}$ obtains $< M_j||\mathrm{ID}_S{,}j, \theta_j >$ from the tuple $<\mathrm{ID}_j, M_j||\mathrm{ID}_S{,}j, R_j, \theta_j, a_j{,}n{-}1, \ldots, a_j, 1, a_j, 0, h_j>$.

2) Search the list L_1 for a tuple $<\mathrm{ID}_j, R_j, K_j, \alpha_j>$ and set the polynomial:

$$\begin{aligned} f(x) &= \prod_{j=1}^{n} (x - \alpha_j) + \theta_j(\mathrm{mod} p) \\ &= x^n + a_{j,n-1}x^{n-1} + \ldots + a_{j,1}x + a_{j,0}, a_{j,i} \in Z_p^*; \end{aligned}$$

3) Search the list L_0 for a tuple $<\mathrm{ID}_j, \mathrm{PK}_j, l_j>$. If there is no such tuple, $\mathcal{C}$ outputs *failure* and *aborts* the game. Otherwise, $\mathcal{C}$ obtains $<\mathrm{ID}_j, \mathrm{PK}_j>$ from the tuple $<\mathrm{ID}_j, \mathrm{PK}_j, l_j>$.

4) Choose the tuple <$\mathrm{ID}_j, \theta_j, R_j, k_j$> from the list L_2 and <$\mathrm{ID}_j, R_j, K_j, \alpha_j$> from the list L_1, and repeatedly check whether $< R_j, K_j >$ is a CDHP tuple or not.

5) If some $< R_t, K_t >$ is a CDHP tuple, compute $\theta_t = f(\alpha_t)$, $k_t = H_2(\theta_t, R_t)$ and $M_t' || \mathrm{ID}_S' = D_{kt}(Z_t)$.

6) Test whether $M_t' = M_j$ holds. If yes, $\mathcal{C}$ returns M_j to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ returns *failure* and aborts the game.

***Challenge*:** $\mathcal{A}_I$ chooses a pair of plaintext $< M_0, M_1 >$ with equal length, and sends them to $\mathcal{C}$. Upon receiving $< M_0, M_1 >$, $\mathcal{C}$ randomly chooses a bit $\beta \in \{0,1\}$ and calculates the ciphertext c^* with the chosen plaintext M_β as follows:

1) Set $R = b(Q_i + X_i)$, $K_i = b\mathrm{PK}_i$ and $\mathrm{PK}_i = X_i + D_i$, where $Q_i = D_i + l_i P_0$;

2) Choose $\alpha_i \in Z_p^*$, for $i \in \{1, 2, \ldots, n\}$;

3) Choose an integer $\theta \in Z_p^*$ and construct a polynomial:

$$f(x) = \prod_{i=1}^{n} (x - \alpha_i) + \theta \ (\mathrm{mod} p)$$
$$= x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0, a_i \in Z_p^*;$$

4) Compute $k = H_2(\theta, R)$, $Z = E_k(M_\beta)$, $h = H_3(M_\beta || \mathrm{ID}_S, R, \theta, a_n - 1, \ldots, a_1, a_0)$;

5) Choose $v \in Z_p^*$;

6) Return the ciphertext $c^* =< R, Z, h, v, a_n - 1, \ldots, a_1, a_0 >$ to $\mathcal{A}_I$.

***Phase* 2:** $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of queries as described in *Phase* 1, but $\mathcal{A}_I$ cannot perform *Set-Secret-Value query* and *Extract-Partial- Private-Key query* on the user whose public key has been replaced, and $\mathcal{A}_I$ cannot perform *De-Signcryption query* on the ciphertext c^*.

***Guess*:** $\mathcal{A}_I$ guesses a bit $\beta^* \in \{0,1\}$. If $\beta^* = \beta$ holds, $\mathcal{A}_I$ wins *Game* 1, and $\mathcal{C}$ outputs $abP = l_i^{-1}(R_i\text{-}K_i)$ as the solution to CDHP. Otherwise, $\mathcal{A}_I$ fails and $\mathcal{C}$ outputs *failure*.

In summary, during the process that $\mathcal{A}_I$ asks the challenger $\mathcal{C}$ for queries, the successful probability of q_k times *Creat*(ID) *queries* is $(1 - \frac{q_0}{p})^{q_c}$, the successful probabilities of q_i times *Hash queries* H_i $(i = 1, 2, 3)$ are $(1 - \frac{q_1}{p})^{q_1}$, $(1 - \frac{q_2}{p})^{q_2}$ and $(1 - \frac{q_3}{p})^{q_3}$ respectively, and the successful probability of q_d times *De-Signcryption queries* is $1 - \frac{q_d}{p}$. Note $(1 - \frac{q_0}{p})^{q_c} \geq (1 - \frac{q_0 q_c}{p})$, $(1 - \frac{q_1}{p})^{q_1} \geq (1 - \frac{q_1^2}{p})$, $(1 - \frac{q_2}{p})^{q_2} \geq (1 - \frac{q_2^2}{p})$ and $(1 - \frac{q_3}{p})^{q_3} \geq (1 - \frac{q_3^2}{p})$. If $\mathcal{A}_I$ has the non-negligible probability advantage ε to win *Game* 1 within probability polynomial time τ, $\mathcal{C}$ has the non-negligible probability advantage:

$$\varepsilon' \geq (1 - \frac{q_0 q_c}{p})(1 - \frac{q_1^2}{p})(1 - \frac{q_2^2}{p})(1 - \frac{q_3^2}{p})(1 - \frac{q_d}{p})\varepsilon$$

to solve CDHP within probability polynomial time:

$$\tau' \leq \tau + (3q_c + 4q_d)\, T_m + T_i.$$

■

Theorem 4: Under IND-CLMS-CCA-II, if there is an adversary $\mathcal{A}_{II}$ who can win *Game* 2 in probability polynomial time τ with a non-negligible probability advantage ε($\mathcal{A}_{II}$ can ask for at most q_i times *Hash queries* H_i $(i = 0, 1, 2, 3)$, q_c times *Create* (ID) *queries*, q_p times *Set-Public-Key queries*, q_r times *Public-Key-Replacement queries*, q_s times *Signcryption queries* and q_d times *De-Signcryption queries*.), the challenger $\mathcal{C}$ can solve CDHP by interacting with the adversary $\mathcal{A}_{II}$ in time

$$\tau' \leq \tau + (3q_c + 4q_d)T_m$$

with a non-negligible probability advantage

$$\varepsilon' \geq (1 - \frac{q_0 q_c}{p})(1 - \frac{q_1^2}{p})(1 - \frac{q_2^2}{p})(1 - \frac{q_3^2}{p})(1 - \frac{q_d}{p})\varepsilon$$

where T_m is the time spent for executing an elliptic curve scalar point multiplication operation.

Proof: Assume that within the polynomial time τ, the adversary $\mathcal{A}_{II}$ can attack the IND-CLMS-CCA-II of the proposed CLMS scheme with a non-negligible probability advantage ε, then there must be a challenger $\mathcal{C}$ who can solve the CDHP by interacting with $\mathcal{A}_{II}$; that is, for given $< P, aP, bP>$, $\mathcal{C}$ will output abP.

Similar to ***Theorem* 3**, $\mathcal{C}$ maintains the lists $L_i (i = 0, 1, 2, 3)$ and L_C.

***Setup*:** $\mathcal{C}$ randomly chooses two integers $s, a \in Z_p^*$ and generates the system's public parameters $Params = < p, F_p, E_p, G_p, P, P_{\mathrm{pub}} = sP, P_0 = aP, E_k, D_k, H_0, H_1, H_2, H_3 >$, then returns system master key s and *Params* to $\mathcal{A}_{II}$. After receiving s and *Params*, $\mathcal{A}_{II}$ selects a set of target multiple identities $L^* = \{\mathrm{ID}_1^*, \mathrm{ID}_2^*, \ldots, \mathrm{ID}_n^*\}$ and sends L^* to $\mathcal{C}$, where n is a positive integer.

***Hash queries*:** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of *Hash queries* as described in ***Theorem* 3**.

***Phase* 1:** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of the following queries and $\mathcal{C}$ responses accordingly:

***Create* (ID$_j$):** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a *Create*(ID_j) *query*. Upon receiving the query, $\mathcal{C}$ checks whether the tuple <$\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j$> is in the list L_C. If yes, $\mathcal{C}$ keeps the tuple. Otherwise, $\mathcal{C}$ randomly chooses three integers $x_j, d_j, l_j \in Z_p^*$, sets $l_j = H_0(\mathrm{ID}_j, \mathrm{PK}_j)$, $D_j = d_j P$, $y_j = d_j + a l_j$, $X_j = x_j P$, and then performs as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$ for $i \in \{1, 2, \ldots, n\}$, sets $\mathrm{SK}_j = \perp$ and $\mathrm{PK}_j = X_j + D_j$, and then updates tuples <$\mathrm{ID}_j, (\perp, y_j), \mathrm{PK}_j$> in the list L_C and <$\mathrm{ID}_j, \mathrm{PK}_j, l_j$> in the list L_0.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$ for $i \in \{1, 2, \ldots, n\}$, sets $\mathrm{SK}_j = x_j + y_j$ and $\mathrm{PK}_j = X_j + D_j$, and then updates tuples <$\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j$> in list L_C and <$\mathrm{ID}_j, \mathrm{PK}_j, l_j$> in the list L_0.

***Set-Secret-Value query*:** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a *Set-Secret-Value query* on the identity ID_j. Upon receiving the query, $\mathcal{C}$ responds as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$ for $i \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ stops the protocol execution.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$ for $i \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ checks if the tuple <$\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j$> is in the list L_C. If yes, $\mathcal{C}$ returns x_j to $\mathcal{A}_{II}$; otherwise, $\mathcal{C}$ asks a *Creat*(ID_j) *query* to obtain the tuple <$\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j$>, and then returns x_j to $\mathcal{A}_{II}$.

***Extract-Partial-Private-key query*:** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for an *Extract-Partial-Private-key query* on identity ID_j. Upon receiving the query, $\mathcal{C}$ checks if the tuple <$\mathrm{ID}_j, (x_j, y_j), \mathrm{PK}_j$>

is in the list L_C. If yes, $\mathcal{C}$ returns y_j to $\mathcal{A}_{II}$; otherwise, $\mathcal{C}$ asks a *Creat*(ID$_j$) *query* to obtain the tuple <ID$_j$, (x_j, y_j), PK$_j$ >, and then returns d_j to $\mathcal{A}_{II}$.

Set-Public-key query: This query is the same as performed in ***Theorem* 3**.

***Public-Key-Replacement query*:** If $\mathcal{A}_{II}$ asks $\mathcal{C}$ to replace PK$_j$ of ID$_j$ with PK$'_j$ chosen by him, $\mathcal{C}$ performs as follows:

a) If ID$_j$ = ID$_i$ for $i \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ stops the protocol execution.

b) If ID$_j \neq$ ID$_i$ for $i \in \{1, 2, \ldots, n\}$, $\mathcal{C}$ looks into the list L_C for PK$_j$ and updates PK$_j$ with PK$'_j$.

***Signcryption query*:** This query is the same as performed in ***Theorem* 3**.

***De-Signcryption query*:** This query is the same as performed in ***Theorem* 3**.

***Challenge*:** $\mathcal{A}_{II}$ chooses a pair of plaintext $< M_0, M_1 >$ with equal length, and sends them to $\mathcal{C}$. Upon receiving $< M_0, M_1 >$, $\mathcal{C}$ randomly chooses a bit $\beta \in \{0, 1\}$ and calculates the ciphertext c^* with the chosen plaintext M_β as follows:

1) Set $R = b(P_0\text{-}D_i\text{-}Q_i)$, $K_j = b(Q_i + D_i)$, where $Q_i = X_i + l_iP_0$ and $D_i = P_0\text{-}x_iP$;

2) Choose $\alpha_i \in Z_p^*$, for $i \in \{1, 2, \ldots, n\}$;

3) Choose an integer $\theta \in Z_p^*$ and construct a polynomial:

$$f(x) = \prod_{i=1}^{n} (x - \alpha_i) + \theta \pmod{p}$$
$$= x^n + a_{n-1}x^{n-1} + \ldots + a_1x + a_0, a_i \in Z_p^*;$$

4) Computes $k = H_2(\theta, R)$, $Z = E_k(M_\beta)$, $h = H_3(M_\beta||\text{ID}_S, R, \theta, a_n{-}1, \ldots, a_1, a_0)$;

5) Choose $v \in Z_p^*$;

6) Return the ciphertext $c^* =< R, Z, h, v, a_n{-}1, \ldots, a_1, a_0 >$ to $\mathcal{A}_{II}$.

***Phase* 2:** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of queries as described in *Phase* 1, but $\mathcal{A}_{II}$ cannot perform *Set-Secret-Value query* and *Extract-Partial-Private-Key query* on the user whose public key has been replaced, and $\mathcal{A}_{II}$ cannot perform *De-Signcryption query* on the ciphertext c^*.

***Guess*:** $\mathcal{A}_{II}$ guesses a bit $\beta^* \in \{0, 1\}$. If $\beta^* = \beta$ holds, $\mathcal{A}_{II}$ wins *Game* 2, and $\mathcal{C}$ outputs $abP = K_i + R$ as the solution to CDHP. Otherwise, $\mathcal{A}_{II}$ fails and $\mathcal{C}$ outputs *failure*.

In summary, during the process that $\mathcal{A}_{II}$ asks the challenger $\mathcal{C}$ for queries, the successful probability of q_k times *Creat*(ID)*queries* is $(1 - \frac{q_0}{p})^{q_c}$, the successful probabilities of q_i times *Hash queries* H_i $(i = 1, 2, 3)$ are $(1-\frac{q_1}{p})^{q_1}$, $(1-\frac{q_2}{p})^{q_2}$ and $(1 - \frac{q_3}{p})^{q_3}$ respectively, and the successful probability of q_d times *De-Signcryption queries* is $1-\frac{q_d}{p}$. Note $(1-\frac{q_0}{p})^{q_c} \geq (1 - \frac{q_0q_c}{p})$, $(1 - \frac{q_1}{p})^{q_1} \geq (1 - \frac{q_1^2}{p})$, $(1 - \frac{q_2}{p})^{q_2} \geq (1 - \frac{q_2^2}{p})$ and $(1-\frac{q_3}{p})^{q_3} \geq (1-\frac{q_3^2}{p})$. If $\mathcal{A}_{II}$ has the non-negligible probability advantage ε to win *Game* 2 within probability polynomial time τ, $\mathcal{C}$ has the non-negligible probability advantage:

$$\varepsilon' \geq (1 - \frac{q_0q_c}{p})(1 - \frac{q_1^2}{p})(1 - \frac{q_2^2}{p})(1 - \frac{q_3^2}{p})(1 - \frac{q_d}{p})\varepsilon$$

to solve CDHP within probability polynomial time:

$$\tau' \leq \tau + (3q_c + 4q_d)T_m.$$

Theorem 5: Under sEUF-CLMS-CPA-I, if there is an adversary $\mathcal{A}_I$ who can win *Game* 3 in probability polynomial time τ with a non-negligible probability advantage $\varepsilon(\mathcal{A}_I$ can ask for at most q_i times *Hash queries* H_i $(i = 0, 1, 2, 3)$, q_c times *Create*(ID) *queries*, q_p times *Set-Public-Key queries*, q_r times *Public-Key-Replacement queries*, q_s times *Signcryption queries* and q_d times *De-Signcryption queries*.), the challenger $\mathcal{C}$ can solve CDHP by interacting with the adversary $\mathcal{A}_I$ in time

$$\tau' \leq \tau + (3q_c + 4q_s)\,T_m + T_i$$

with a non-negligible probability advantage

$$\varepsilon' \geq (1 - \frac{q_0q_c}{p})(1 - \frac{q_1^2}{p})(1 - \frac{q_2^2}{p})(1 - \frac{q_3^2}{p})(1 - \frac{q_s}{p})\varepsilon$$

where T_m is the time spent for executing an elliptic curve scalar point multiplication operation and T_i is the time spent for executing a modular inversion operation.

Proof: Assume that within the polynomial time τ, the adversary $\mathcal{A}_I$ can attack the sEUF-CLMS-CPA-I of the proposed CLMS scheme with a non-negligible probability advantage ε, then there must be a challenger $\mathcal{C}$ who can solve the CDHP by interacting with $\mathcal{A}_I$; that is, for given $< P, aP, bP>$, $\mathcal{C}$ will output abP.

Similar to ***Theorem*3**, $\mathcal{C}$ maintains the lists $L_i(i = 0, 1, 2, 3)$ and L_C.

***Setup*:** $\mathcal{C}$ runs this algorithm to generate the system master key $s = a \in Z_p^*$ and the system's public parameters *Params*=$< p, F_p, E_p, G_p, P, P_0 = aP, E_k, D_k, H_0, H_1, H_2, H_3 >$, then keeps s secret and returns *Params* to $\mathcal{A}_I$. After receiving *Params*, $\mathcal{A}_I$ chooses a group of target identities $L^* = \{\text{ID}_1^*, \text{ID}_2^*, \ldots, \text{ID}_n^*\}$ and sends L^* to $\mathcal{C}$, where n is a positive integer.

***Hash queries*:** $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of *Hash queries* as described in ***Theorem*3**.

***Attack*:** $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of queries as *Phase* 1 in ***Theorem*3**.

***Forgery*:** With a plaintext M, a sender ID$_S \in L^*$ and a group of receivers identities $L = \{\text{ID}_1, \text{ID}_2, \ldots, \text{ID}_n\}$, $\mathcal{A}_I$ forges the ciphertext $c^* =< R^*, Z^*, h^*, v^*, a_n{-}1^*, \ldots, a_1^*, a_0^*>$. If the ciphertext c^* is forged successfully, equations $h^{*\prime} = h^*$ and $v^*P = PK_S + H_0(\text{ID}_S, \text{PK}_S)P_{\text{pub}} + h^*R^*$ hold. Setting $R = b\text{PK}_i$, $K_i = b(\text{PK}_i + l_iP_0)$, the challenger $\mathcal{C}$ outputs $abP = l_i^{-1}(K_i\text{-}R)$. There is a restriction that the ciphertext c^* cannot be generated by *Signcryption query*.

In summary, during the process that $\mathcal{A}_I$ asks the challenger $\mathcal{C}$ for queries, the successful probability of q_k times *Creat*(ID)*queries* is $(1 - \frac{q_0}{p})^{q_c}$, the successful probabilities of q_i times *Hash queries* H_i $(i = 1, 2, 3)$ are $(1-\frac{q_1}{p})^{q_1}$, $(1-\frac{q_2}{p})^{q_2}$ and $(1 - \frac{q_3}{p})^{q_3}$ respectively, and the successful probability of q_s times *Signcryption queries* is $1 - \frac{q_s}{p}$. Note $(1 - \frac{q_0}{p})^{q_c} \geq (1 - \frac{q_0q_c}{p})$, $(1 - \frac{q_1}{p})^{q_1} \geq (1 - \frac{q_1^2}{p})$, $(1 - \frac{q_2}{p})^{q_2} \geq (1 - \frac{q_2^2}{p})$ and

$(1-\frac{q_3}{p})^{q_3} \geq (1-\frac{q_3^2}{p})$. If $\mathcal{A}_I$ has the non-negligible probability advantage ε to win *Game* 3 within probability polynomial time τ, $\mathcal{C}$ has the non-negligible probability advantage:

$$\varepsilon' \geq (1-\frac{q_0 q_c}{p})(1-\frac{q_1^2}{p})(1-\frac{q_2^2}{p})(1-\frac{q_3^2}{p})(1-\frac{q_s}{p})\varepsilon$$

to solve CDHP within probability polynomial time:

$$\tau' \leq \tau + (3q_c + 4q_s)\, T_m + T_i.$$

■

Theorem 6: Under sEUF-CLMS-CPA-II, if there is an adversary $\mathcal{A}_{II}$ who can win *Game* 4 in probability polynomial time τ with a non-negligible probability advantage ε ($\mathcal{A}_{II}$ can ask for at most q_i times *Hash queries* H_i $(i = 0, 1, 2, 3)$, q_c times *Create*(ID) *queries*, q_p times *Set-Public-Key queries*, q_r times *Public-Key-Replacement queries*, q_s times *Signcryption queries* and q_d times *De-Signcryption queries*.), the challenger $\mathcal{C}$ can solve CDHP by interacting with the adversary $\mathcal{A}_I$ in time

$$\tau' \leq \tau + (3q_c + 4q_s)T_m$$

with a non-negligible probability advantage

$$\varepsilon' \geq (1-\frac{q_0 q_c}{p})(1-\frac{q_1^2}{p})(1-\frac{q_2^2}{p})(1-\frac{q_3^2}{p})(1-\frac{q_s}{p})\varepsilon$$

where T_m is the time spent for executing an elliptic curve scalar point multiplication operation.

Proof: Assume that within the polynomial time τ, the adversary $\mathcal{A}_{II}$ can attack the sEUF-CLMS-CPA-II of the proposed CLMS scheme with a non-negligible probability advantage ε, then there must be a challenger $\mathcal{C}$ who can solve the CDHP by interacting with $\mathcal{A}_{II}$; that is, for given $< P, aP, bP>$, $\mathcal{C}$ will output abP.

Similar to ***Theorem*3**, $\mathcal{C}$ maintains the lists $L_i(i = 0, 1, 2, 3)$ and L_C.

Setup: $\mathcal{C}$ randomly chooses two integers $s, a \in Z_p^*$ and generates the system's public parameters *Params*$=< p, F_p, E, G_p, P, P_{\text{pub}} = sP, P_0 = aP, E_k, D_k, H_0, H_1, H_2, H_3 >$, then returns system master key s and *Params* to $\mathcal{A}_{II}$. After receiving *Params*, $\mathcal{A}_{II}$ chooses a group of target identities $L^* = \{\text{ID}_1^*, \text{ID}_2^*, \ldots, \text{ID}_n^*\}$ and sends L^* to $\mathcal{C}$, where n is a positive integer.

Hash queries: $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of *Hash queries* as described in ***Theorem*3**.

Attack: $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of queries as *Phase* 1 in ***Theorem*4**.

Forgery: With a plaintext M, a sender $\text{ID}_S \in L^*$ and a group of receivers identities $L = \{\text{ID}_1, \text{ID}_2, \ldots, \text{ID}_n\}$, $\mathcal{A}_{II}$ forges the ciphertext $c^* =< R^*, Z^*, h^*, v^*, a_n{-}1^*, \ldots, a_1^*, a_0^* >$. If the ciphertext c^* is forged successfully, equations $h^{*\prime} = h^*$ and $v^*P = PK_S + H_0(\text{ID}_S, \text{PK}_S)P_{\text{pub}} + h^*R^*$ hold. Setting $R = b(\text{PK}_i + P_0)$, $K_i = b(\text{PK}_i + l_iP_0)$, the challenger $\mathcal{C}$ outputs $abP = (l_i\text{-}1)^{-1}(K_i\text{-}R)$. There is a restriction that the ciphertext c^* cannot be generated by *Signcryption query*.

In summary, during the process that $\mathcal{A}_{II}$ asks the challenger $\mathcal{C}$ for queries, the successful probability of q_k times *Creat*(ID) *queries* is $(1-\frac{q_0}{p})^{q_c}$, the successful probabilities of q_i times *Hash queries* H_i $(i = 1, 2, 3)$ are $(1-\frac{q_1}{p})^{q_1}$, $(1-\frac{q_2}{p})^{q_2}$ and $(1-\frac{q_3}{p})^{q_3}$ respectively, and the successful probability of q_s times *Signcryption queries* is $1-\frac{q_s}{p}$. Note $(1-\frac{q_0}{p})^{q_c} \geq (1-\frac{q_0 q_c}{p})$, $(1-\frac{q_1}{p})^{q_1} \geq (1-\frac{q_1^2}{p})$, $(1-\frac{q_2}{p})^{q_2} \geq (1-\frac{q_2^2}{p})$ and $(1-\frac{q_3}{p})^{q_3} \geq (1-\frac{q_3^2}{p})$. If $\mathcal{A}_{II}$ has the non-negligible probability advantage ε to win *Game* 4 within probability polynomial time τ, $\mathcal{C}$ has the non-negligible probability advantage:

$$\varepsilon' \geq (1-\frac{q_0 q_c}{p})(1-\frac{q_1^2}{p})(1-\frac{q_2^2}{p})(1-\frac{q_3^2}{p})(1-\frac{q_s}{p})\varepsilon$$

to solve CDHP within probability polynomial time:

$$\tau' \leq \tau + (3q_c + 4q_s)T_m.$$

■

Theorem 7: Under ANON-IND-CLMS-CCA-I, if there is an adversary $\mathcal{A}_I$ who can win *Game* 5 in probability polynomial time τ with a non-negligible probability advantage ε ($\mathcal{A}_I$ can ask for at most q_i times *Hash queries* H_i $(i = 0, 1, 2, 3)$, q_c times *Create*(ID) *queries*, q_p times *Set-Public-Key queries*, q_r times *Public-Key-Replacement queries*, q_s times *Signcryption queries* and q_d times *De-Signcryption queries*.), the challenger $\mathcal{C}$ can solve CDHP by interacting with the adversary $\mathcal{A}_I$ in time

$$\tau' \leq \tau + (3q_c + 4q_d)\, T_m + T_i$$

with a non-negligible probability advantage

$$\varepsilon' \geq (1-\frac{q_0 q_c}{p})(1-\frac{q_1^2}{p})(1-\frac{q_2^2}{p})(1-\frac{q_3^2}{p})(1-\frac{q_d}{p})\varepsilon$$

where T_m is the time spent for executing an elliptic curve scalar point multiplication operation and T_i is the time spent for executing a modular inversion operation.

Proof : Assume that within the polynomial time τ, the adversary $\mathcal{A}_I$ can attack the ANON-CLMS-CCA-I of the proposed CLMS scheme with a non-negligible probability advantage ε, then there must be a challenger $\mathcal{C}$ who can solve the CDHP by interacting with $\mathcal{A}_I$; that is, for given $< P, aP, bP>$, $\mathcal{C}$ will output abP.

Similar to ***Theorem* 3**, $\mathcal{C}$ maintains the lists $L_i(i = 0, 1, 2, 3)$ and L_C.

Setup: $\mathcal{C}$ runs this algorithm to generate the system master key $s = a \in Z_p^*$ and the system's public parameters *Params* $=< p, F_p, E_p, G_p, P, P_0 = aP, E_k, D_k, H_0, H_1, H_2, H_3 >$, and then keeps s secret and returns *Params* to $\mathcal{A}_I$. After receiving *Params*, $\mathcal{A}_I$ selects a pair of target multiple identities $L^* = \{\text{ID}_0^*, \text{ID}_1^*\}$, and then sends L^* to $\mathcal{C}$.

Hash queries: $\mathcal{A}_I$ asks $\mathcal{C}$ for a series of *Hash queries* as described in ***Theorem*3**

***Phase* 1:** This phase is the same as *Phase* 1 in ***Theorem*3**.

Challenge: $\mathcal{A}_I$ chooses a plaintext M and a set of target identities $L = \{\text{ID}_2, \text{ID}_3, \ldots, \text{ID}_n\}$, where n is a positive integer, and then sends M and L to $\mathcal{C}$. $\mathcal{C}$ randomly chooses a bit $e \in\{0,1\}$ and computes the ciphertext c^* with a group of new target identities $L' = \{ID_e^*, ID_2, ID_3, \ldots, ID_n\}$ as follows:

1) Set $R = b(Q_i + X_i)$, $K_i = bPK_i$, $PK_i = X_i + D_i$, where $Q_i = D_i + l_i P_0$;

2) For $i \in \{2, 3, \ldots, n\}$, choose a tuple $<ID_i, R_i, K_i>$ and compute $\alpha_i = H_1(ID_i, R_i, K_i)$;

3) Choose $\alpha, \theta \in Z_p^*$, and construct a polynomial:

$$f(x) = \prod_{i=1}^{n}(x - \alpha_i) + \theta \,(\text{mod} p) = x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0,\ a_i \in Z_p^*;$$

4) Computes $k = H_2(\theta, R)$, $Z = E_k(M)$, $h = H_3(M||ID_S, R, \theta, a_n-1, \ldots, a_1, a_0)$;

5) Choose $v \in Z_p^*$;

6) Return the ciphertext $c^* =< R, Z, h, v, a_n-1, \ldots, a_1, a_0 >$ to $\mathcal{A}_I$.

***Phase* 2:** This phase is the same as *Phase* 2 in ***Theorem* 3**.

***Guess*:** $\mathcal{A}_I$ guesses a bit $e^* \in\{0,1\}$. If $e^* = e$ holds, $\mathcal{A}_I$ wins *Game* 5, and $\mathcal{C}$ outputs $abP= l_i^{-1}(R_i\text{-}K_i)$ as the solution to CDHP. Otherwise, $\mathcal{A}_I$ fails and $\mathcal{C}$ outputs *failure*.

In summary, during the process that $\mathcal{A}_I$ asks the challenger $\mathcal{C}$ for queries, the successful probability of q_k times *Creat*(ID)*queries* is $(1 - \frac{q_0}{p})^{q_c}$, the successful probabilities of q_i times *Hash queries* H_i $(i = 1, 2, 3)$ are $(1-\frac{q_1}{p})^{q_1}$, $(1-\frac{q_2}{p})^{q_2}$ and $(1 - \frac{q_3}{p})^{q_3}$ respectively, and the successful probability of q_d times *De-Signcryption queries* is $1-\frac{q_d}{p}$. Note $(1-\frac{q_0}{p})^{q_c} \geq (1 - \frac{q_0 q_c}{p})$, $(1 - \frac{q_1}{p})^{q_1} \geq (1 - \frac{q_1^2}{p})$, $(1 - \frac{q_2}{p})^{q_2} \geq (1 - \frac{q_2^2}{p})$ and $(1-\frac{q_3}{p})^{q_3} \geq (1-\frac{q_3^2}{p})$. If $\mathcal{A}_I$ has the non-negligible probability advantage ε to win *Game* 5 within probability polynomial time τ, $\mathcal{C}$ has the non-negligible probability advantage:

$$\varepsilon' \geq (1 - \frac{q_0 q_c}{p})(1 - \frac{q_1^2}{p})(1 - \frac{q_2^2}{p})(1 - \frac{q_3^2}{p})(1 - \frac{q_d}{p})\varepsilon$$

to solve CDHP within probability polynomial time:

$$\tau' \leq \tau + (3q_c + 4q_d)\,T_m + T_i.$$

■

Theorem 8: Under ANON-IND-CLMS-CCA-II, if there is an adversary $\mathcal{A}_{II}$ who can win *Game* 6 in probability polynomial time τ with a non-negligible probability advantage ε ($\mathcal{A}_I$ can ask for at most q_i times *Hash queries* H_i $(i = 0, 1, 2, 3)$, q_c times *Create*(ID) *queries*, q_p times *Set-Public-Key queries*, q_s times *Signcryption queries* and q_d times *De-Signcryption queries*.), the challenger $\mathcal{C}$ can solve CDHP by interacting with the adversary $\mathcal{A}_{II}$ in time

$$\tau' \leq \tau + (3q_c + 4q_d)\,T_m$$

with a non-negligible probability advantage

$$\varepsilon' \geq (1 - \frac{q_0 q_c}{p})(1 - \frac{q_1^2}{p})(1 - \frac{q_2^2}{p})(1 - \frac{q_3^2}{p})(1 - \frac{q_d}{p})\varepsilon$$

where T_m is the time spent for executing an elliptic curve scalar point multiplication operation.

Proof: Assume that within the polynomial time τ, the adversary $\mathcal{A}_{II}$ can attack the ANON-CLMS-CCA-II of the proposed CLMS scheme with a non-negligible probability advantage ε, then there must be a challenger $\mathcal{C}$ who can solve the CDHP by interacting with $\mathcal{A}_{II}$; that is, for given $< P, aP, bP>$, $\mathcal{C}$ will output abP.

Similar to ***Theorem*3**, $\mathcal{C}$ maintains the lists $L_i(i = 0, 1, 2, 3)$ and L_C.

***Setup*:** $\mathcal{C}$ randomly chooses two integers $s, a \in Z_p^*$ and generates the system's public parameters $Params = < p, F_p, E_p, G_p, P, P_{pub} = sP, P_0 = aP, E_k, D_k, H_0, H_1, H_2, H_3 >$, and then returns system master key s and $Params$ to $\mathcal{A}_{II}$. After receiving $Params$, $\mathcal{A}_{II}$ selects a pair of target multiple identities $L^* = \{ID_0^*, ID_1^*\}$, and then sends L^* to $\mathcal{C}$.

***Hash queries*:** $\mathcal{A}_{II}$ asks $\mathcal{C}$ for a series of *Hash queries* as described in ***Theorem*3**.

***Phase* 1:** This phase is the same as *Phase* 1 in ***Theorem* 4**.

***Challenge*:** $\mathcal{A}_{II}$ chooses a plaintext M and a set of target identities $L = \{ID_2, ID_3, \ldots, ID_n\}$, where n is a positive integer, and then sends M and L to $\mathcal{C}$. $\mathcal{C}$ randomly chooses a bit $e \in \{0, 1\}$ and computes the ciphertext c^* with a group of new target identities $L' = \{ID_e^*, ID_2, ID_3, \ldots, ID_n\}$ as follows:

1) Set $R = b(P_0\text{-}D_i\text{-}Q_i)$, $K_j = b(Q_i + D_i)$, where $Q_i = X_i + l_i P_0$ and $D_i = P_0\text{-}x_i P$;

2) For $i \in \{2, 3, \ldots, n,\}$, choose a tuple $<ID_i, R_i, K_i>$ and compute $\alpha_i = H_1(ID_i, R_i, K_i)$;

3) Choose $\alpha, \theta \in Z_p^*$, and construct a polynomial:

$$f(x) = \prod_{i=1}^{n}(x - \alpha_i) + \theta \,(\text{mod} p) = x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0,\ a_i \in Z_p^*;$$

4) Compute $k = H_2(\theta, R)$, $Z = E_k(M)$, $h = H_3(M||ID_S, R, \theta, a_n-1, \ldots, a_1, a_0)$;

5) Choose $v \in Z_p^*$;

6) Return the ciphertext $c^* =< R, Z, h, v, a_n-1, \ldots, a_1, a_0 >$ to $\mathcal{A}_{II}$.

***Phase* 2:** This phase is the same as *Phase* 2 in ***Theorem* 4**.

***Guess*:** $\mathcal{A}_{II}$ guesses a bit $e^* \in\{0,1\}$. If $e^* = e$ holds, $\mathcal{A}_I$ wins*Game* 6, and $\mathcal{C}$ outputs $abP= K_i + R$ as the solution to CDHP. Otherwise, $\mathcal{A}_{II}$ fails and $\mathcal{C}$ outputs *failure*.

In summary, during the process that $\mathcal{A}_{II}$ asks the challenger $\mathcal{C}$ for queries, the successful probability of q_k times *Creat*(ID)*queries* is $(1 - \frac{q_0}{p})^{q_c}$, the successful probabilities of q_i times *Hash queries* H_i $(i = 1, 2, 3)$ are $(1-\frac{q_1}{p})^{q_1}$, $(1-\frac{q_2}{p})^{q_2}$ and $(1 - \frac{q_3}{p})^{q_3}$ respectively, and the successful probability of q_d times *De-Signcryption queries* is $1-\frac{q_d}{p}$. Note $(1-\frac{q_0}{p})^{q_c} \geq (1 - \frac{q_0 q_c}{p})$, $(1 - \frac{q_1}{p})^{q_1} \geq (1 - \frac{q_1^2}{p})$, $(1 - \frac{q_2}{p})^{q_2} \geq (1 - \frac{q_2^2}{p})$ and $(1-\frac{q_3}{p})^{q_3} \geq (1-\frac{q_3^2}{p})$. If $\mathcal{A}_{II}$ has the non-negligible probability advantage ε to win *Game* 6 within probability polynomial time τ, $\mathcal{C}$ has the non-negligible probability advantage:

$$\varepsilon' \geq (1 - \frac{q_0 q_c}{p})(1 - \frac{q_1^2}{p})(1 - \frac{q_2^2}{p})(1 - \frac{q_3^2}{p})(1 - \frac{q_d}{p})\varepsilon$$

TABLE 2. Symbols' definition.

Symbols	Symbols' definition
T_m	Time of calculating a modular multiplication operation.
T_b	Time of calculating a bilinear pairing operation, $T_b \approx 87T_m$.
T_e	Time of calculating a modular exponentiation operation, $T_e \approx 240T_m$.
T_h	Time of calculating a map-to-point hash function operation, $T_h \approx 29T_m$.
T_i	Time of calculating a modular inversion operation, $T_i \approx 11.6T_m$.
T_{be}	Time of calculating a bilinear pairing exponentiation operation, $T_{be} \approx 43.5T_m$.
T_{pa}	Time of calculating a point addition operation, $T_{pa} \approx 0.12T_m$.
T_{pm}	Time of calculating a scalar point multiplication, $T_{pm} \approx 29T_m$.

TABLE 3. Comparison of efficiency.

Schemes	Encryption/Signcryption	Decryption/De-Signcryption
Fan *et al.*[23]	$(n+2)T_{pm}+nT_b+nT_h+T_{be} \approx (145n+101.5)T_m$	$T_{pm}+3T_b+T_h+T_{be} \approx 362.5T_m$
Selvi *et al.*[31]	$(n+1)T_{pm}+T_i+(n+1)T_b+(n+1)T_{be} \approx (159.5n+171.1)T_m$	$T_{pm}+T_{pa}+2T_b+T_{be} \approx 246.62T_m$
Islam *et al.*[34]	$(2n+1)T_{pm}+2nT_{pa} \approx (58.24n+29)T_m$	$T_{pm} \approx 29T_m$
Hung *et al.*[35]	$(n+1)T_{pm}+nT_b+nT_h+nT_{be} \approx (188.5n+29)T_m$	$T_{pm}+T_b \approx 116T_m$
He *et al.*[36]	$(3n+1)T_{pm}+nT_{pa} \approx (87.12n+29)T_m$	$2T_{pm} \approx 58T_m$
Tseng *et al.*[37]	$(2n+3)T_{pm}+2nT_b+nT_h+T_{be} \approx (261n+103.5)T_m$	$2T_{pm}+6T_b+T_h+T_{be} \approx 652.5T_m$
Win *et al.*[38]	$(n+2)T_{pm}+nT_{pa} \approx (29.12\text{n}+58)T_m$	$4T_{pm}+4T_{pa}+2T_i \approx 139.68T_m$
Pang *et al.*[39]	$(n+1)T_{pm}+nT_{pa} \approx (29.12n+29)T_m$	$2T_{pm}+T_{pa} \approx 58.12T_m$
Zhu *et al.*[41]	$(2n+3)T_{pm}+nT_{pa}+T_b \approx (58.12n+174)T_m$	$2T_b+T_{pm}+T_i \approx 214.6T_m$
Gao *et al.*[42]	$(2n+1)T_{pm}+2nT_{pa} \approx (58.24n+29)T_m$	$2T_{pm} \approx 58T_m$
Our scheme	$(2n+1)T_{pm}+nT_{pa} \approx (58.12n+29)T_m$	$4T_{pm}+2T_{pa} \approx 106.24T_m$

n indicates the number of receivers.

to solve CDHP within probability polynomial time:

$$\tau' \leq \tau + (3q_c + 4q_d)T_m.$$

∎

V. EFFICIENCY ANALYSIS AND FUNCTIONAL COMPARISON

To show the advantages of our scheme, we will compare our scheme with schemes [23], [31], [34]–[39], [41] and [42] in terms of computational efficiency and functions, because these schemes are similar to our scheme in functions or cryptographic foundation.

A. EFFICIENCY ANALYSIS

In order to express the computational efficiency of each scheme conveniently, we define some symbols to represent the time spent on different computations, shown in TABLE 2 (The data are from [34]). The time spent on the encryption/signcryption and decryption/de-signcryption algorithms of each scheme only includes the computational operations listed in TABLE 2, because the computational operations which are not listed take so little time as to be negligible. The time spent on encryption/signcryption process and decryption/de-signcryption process is shown in TABLE 3, from where we can see the performance of each scheme in terms of computational complexity.

In TABLE 4, we give the simulation running time of each scheme under $n = 5, 10, 15, 20$ and 25, where n is the number of authorized receivers. The simulation is implemented on a Inter(R) Core(TM)2 Duo 2.93GHz processor and 2.00GB RAM using Windows XP and JDK Visual C++ 6.0, the length of private key is 128bits and symmetric encryption algorithm is AES. To facilitate simulation, we use SHA-1 to implement H_0, H_1, H_2, and H_3. For example, "$\{0,1\}^*$" and "G_p" are two input parameters of H_0, and we can compute SHA-1($\{0,1\}^*||G_p$) as the output of H_0. We adopt the similar way to implement for the hash functions in the related schemes.

From TABLE 3 and TABLE 4, we can see that our scheme is more efficient than schemes [23], [31], [34]–[37], [41], and [42] in encryption/signcryption process, and it is more efficient than schemes [23], [31], [35], [37], [38], and [41] in decryption/de-signcryption process. However, our scheme is more inefficient than schemes [38] and [39] in encryption/signcryption process, and it is more inefficient than schemes [34], [36], [39], and [42] in decryption/de-signcryption process, because we have increased some computation to avoid the use of secure channels. Although the computational complexity of our scheme is higher than that of some schemes, the extra calculation costs are considered acceptable when considering the costs spent on maintaining

TABLE 4. Simulation data of running time.

Schemes	Cost of encryption/signcryption (*ms*)					Cost of decryption/de-signcryption (*ms*)
	n=5	*n*=10	*n*=15	*n*=20	*n*=25	
Fan *et al.*[23]	229.32	430.41	631.50	832.59	1033.68	100.43
Selvi *et al.*[31]	268.97	490.06	711.14	932.22	1153.30	68.72
Islam *et al.*[34]	93.56	176.25	258.93	341.62	424.31	11.11
Hung *et al.*[35]	274.33	537.79	801.25	1064.70	1328.16	35.58
He *et al.*[36]	135.77	260.67	385.57	510.47	635.36	19.83
Tseng *et al.*[37]	389.78	750.79	1111.79	1472.80	1833.80	180.14
Win *et al.*[38]	62.69	102.84	142.98	184.12	225.27	41.39
Pang *et al.*[39]	56.61	99.16	141.70	183.24	223.78	19.14
Zhu *et al.*[41]	196.30	356.38	516.46	676.54	836.61	59.41
Gao *et al.*[42]	94.96	175.45	256.94	338.42	419.91	16.71
Our scheme	92.20	173.52	254.84	336.16	417.48	35.41

TABLE 5. Comparison of functions.

schemes	no key escrow problem	receiver anonymity	source verifiability	decryption fairness	partial private key verifiability	no secure channel
Fan *et al.* [23]	No	Yes	No*	Yes	No	No
Selvi *et al.*[31]	Yes	No	Yes	No	No	No
Islam *et al.*[34]	Yes	Yes	No	Yes	Yes	No
Hung *et al.*[35]	Yes	Yes	No	No	Yes	No
He *et al.*[36]	Yes	Yes	No	No	No	No
Tseng *et al.* [37]	Yes	Yes	No*	Yes	No	No
Win *et al.*[38]	Yes	No	Yes	No	Yes	No
Pang *et al.*[39]	Yes	Yes	Yes	Yes	Yes	No
Zhu et al.[41]	Yes	No	No	No	No	No
Gao *et al.*[42]	Yes	Yes	No	Yes	Yes	No
Our scheme	Yes	Yes	Yes	Yes	Yes	Yes

No* indicates that the scheme claims to have source verifiability, but in fact, it cannot really verify the message source.

the secure channel, because it is well known that maintaining a secure channel requires a lot in practical applications.

B. FUNCTIONAL COMPARISON

The comparisons of functions between our scheme and schemes [23], [31], [34]–[39], [41] and [42] are shown in TABLE 5, from where we can see the performance of each scheme in terms of functions.

From TABLE 5, we can see that only scheme [23] has key escrow problem because it is based on IBC. In terms of privacy protection, our scheme and schemes [23], [34]–[37], [39], and [42] provide receiver anonymity so that no one except the sender knows the authorized receivers, whereas schemes [31], [38], and [41] do not consider receiver anonymity. Our scheme and schemes [31], [38], and [39] offer source verifiability to resist the forgery of attackers. However, schemes [23] and [37] fails to implement source verifiability function as their claimed, because there is a lack of sender's signature to the message. Schemes [34]–[36], [41] and [42] even do not take source verifiability into account. In addition, our scheme and schemes [23], [34], [37], [39], and [42] achieve decryption fairness, while other schemes do not. Besides, our scheme and schemes [34], [35], [38], [39], and [42] have partial private key verifiability which ensures the correctness of the user's partial private key, but other schemes do not have partial private key verifiability. Finally, we can see that only our scheme does not use the secure channel to transmit the partial private key. To sum up, our scheme has more functions than the existing similar schemes.

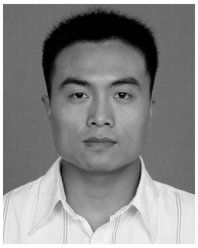
VI. CONCLUSION

In this paper, in order to solve the problem that the key extract algorithm relies on a secure channel in the existing certificateless multi-receiver signcryption schemes, we proposed a new anonymous certificateless multi-receiver signcryption scheme, which transmits the partial private key through a public channel. The proposed scheme not only has higher security of the private key, but also reduces the system complexity of the practical applications because it avoids maintaining a secure channel. Therefore, whether in security, efficiency or functions, the proposed scheme is more suitable for practical applications. Although the computational complexity of our scheme is higher than that of some schemes, the extra calculation costs are considered acceptable compared with the costs of maintaining a secure channel.

Despite this, finding the new design method to improve the computation efficiency may be our next work.

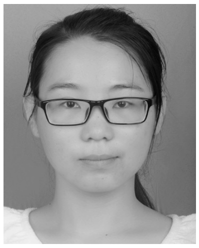
REFERENCES

[1] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Netw.*, vol. 8, nos. 2–3, pp. 153–167, 2002.
[2] M. Bellare, A. Boldyreva, and S. Micali, "Public-key encryption in a multi-user setting: Security proofs and improvements," in *Advances in Cryptology—EUROCRYPT*, vol. 1807. Berlin, Germany: Springer, 2000, pp. 259–274.
[3] S. Duan and Z. Cao, "Efficient and provably secure multi-receiver identity-based signcryption," in *Information Security and Privacy*, vol. 4058. Berlin, Germany: Springer, 2006, pp. 195–206.
[4] L. Pang, H. Li, and Q. Pei, "Improved multicast key management of Chinese wireless local area network security standard," *IET Commun.*, vol. 6, no. 9, pp. 1126–1130, Jun. 2012.
[5] L. H. E. Fadil, A. Moumen, and M. Bouye, "Anonymous multi-receiver public key encryption based on lucas sequences," *Preprints*, to be published. doi: 10.20944/preprints201810.0751.v1.
[6] A. Chillali and E. L. Fadil, "Anonymous multi-receiver public key encryption based on third order linear sequences," in *Proc. 2nd Int. Conf. Appl. Math. (ICAM)*, vol. 2074, no. 1, p. 020013, 2019.
[7] J. Baek, R. Safavi-Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 3286. Berlin, Germany: Springer, 2005, pp. 380–397.
[8] X. Boyen, "Multipurpose identity-based signcryption," in *Advances in Cryptology—CRYPTO*, vol. 2729. Berlin, Germany: Springer, 2003, pp. 383–399.
[9] X. Du, Y. Wang, J. Ge, and Y. Wang, "An ID-based broadcast encryption scheme for key distribution," *IEEE Trans. Broadcast.*, vol. 51, no. 2, pp. 264–266, Jun. 2005.
[10] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 4833. Berlin, Germany: Springer, pp. 200–215.
[11] Y. M. Tseng, T. T. Tsai, and Y. T. Wu, "Efficient revocable multi-receiver ID-based encryption," *Inf. Technol. Control.*, vol. 42, no. 2, pp. 159–169, 2013.
[12] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption)," in *Advances in Cryptology—CRYPTO*. Santa Barbara, CA, USA: Springer, 1997, pp. 165–179.
[13] S. Lal and P. Kushwah, "Anonymous ID based signcryption scheme for multiple receivers," in *IACR Cryptology ePrint Archive*. Las Vegas, NV, USA, 2009.
[14] 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.
[15] X. Zhang, C. Xu, and J. Xue, "Efficient multi-receiver identity-based signcryption from lattice assumption," *Int. J. Electron. Secur. Digit. Forensics*, vol. 10, no. 1, pp. 20–28, 2018.
[16] Z. Yu, Z. Jing, H. Yang, and C. Gu, "ID-based multi-receiver signcryption scheme in the standard model," *Int. J. Internet Protoc. Technol.*, vol. 10, no. 1, pp. 4–12, Mar. 2017.
[17] A. Srivastava and G. Geethakumari, "Measuring privacy leaks in online social networks," in *Proc. Jayachamarajendra College Eng. (ICACCI)*, Mysore, India, 2013, pp. 2095–2100.
[18] L. Pang, H. Li, L. Gao, and Y. Wang, "Completely anonymous multi-recipient signcryption scheme with public verification," *PLoS ONE*, vol. 8, no. 5, p. e63562, May 2013.
[19] C.-I. Fan, L.-Y. Huang, and P.-H. Ho, "Anonymous multireceiver identity-based encryption," *IEEE Trans. Comput.*, vol. 59, no. 9, pp. 1239–1249, Jan. 2010.
[20] H. Wang, Y. Zhang, H. Xiong, and B. Qin, "Cryptanalysis and improvements of an anonymous multi-receiver identity-based encryption scheme," *IET Inf. Secur.*, vol. 6, no. 1, pp. 20–27, Mar. 2012.
[21] 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.
[22] Y. M. Tseng, Y. H. Huang, and H. J. Chang, "Privacy-preserving multi-receiver ID-based encryption with provable security," *Int. J. Commun. Syst.*, vol. 27, no. 7, pp. 1034–1050, Jul. 2014.
[23] C.-I. Fan and Y.-F. Tseng, "Anonymous multi-receiver identity-based authenticated encryption with CCA security," *Symmetry*, vol. 7, no. 4, pp. 1856–1881, Dec. 2015.
[24] L. Pang, X. Yan, H. Zhao, Y. Hu, and H. Li, "A novel multi-receiver signcryption scheme with complete anonymity," *PLoS ONE*, vol. 11, no. 11, p. e0166173, Nov. 2016.
[25] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology* (Lecture Notes in Computer Science). Santa Barbara, CA, USA: Springe, 1985, pp. 47–53.
[26] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT (LNCS)*, vol. 2894. Berlin, Germany: Springer, 2003, pp. 452–473.
[27] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from asiacrypt 2003," in *Proc. Int. Conf. Crypto. Netw. Secur.*, in Lecture Notes in Computer Science, vol. 3810. Berlin, Germany: Springer, 2005, pp. 13–25.
[28] Z. F. Zhang, D. S. Wong, J. Xu, and D. G. Feng, "Certificateless public-key signature: Security model and efficient construction," in *Proc. Int. Conf. Appl. Crypto. Netw. Secur.*, in Lecture Notes in Computer Science, vol. 3989. Berlin, Germany: Springer, 2006, pp. 293–308.
[29] X. Y. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signature revisited," in *Information Security and Privacy*, in Lecture Notes in Computer Science, vol. 4586, Berlin, Germany: Springer, 2007, pp. 308–322.
[30] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proc. ACM, Symp. Inf., Comput. Commun. Secur.*, Tokyo, Japan, 2008, pp. 369–372.
[31] 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 in Computer Science), vol. 5324. Berlin, Germany: Springer, 2008, pp. 52–67.
[32] S. S. D. Selvi, S. S. Vivek, and C. P. Rangan, "A note on the certificateless multi-receiver signcryption scheme," in *IACR Cryptology ePrint Archive*. Las Vegas, NV, USA, 2009. [Online]. Available: http://eprint.iacr.org/2009/308.pdf
[33] S. Q. Miao, F. T. Zhang, and L. Zhang, "Cryptanalysis of a certificateless multi-receiver signcryption scheme," in *Proc. Int. Conf.*, Mines, NJ, USA, 2010, pp. 593–597.
[34] 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.
[35] 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.
[36] D. He, H. Wang, L. Wang, J. Shen, and X. Yang, "Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices," *Soft Comput.*, vol. 21, no. 22, pp. 6801–6810, Nov. 2017.
[37] Y. F. Tseng and C.-I. Fan, "Provably CCA-secure anonymous multi-receiver certificateless authenticated encryption," *J. Inf. Sci. Eng.*, vol. 34, no. 6, pp. 1517–1541, Nov. 2018.
[38] E. K. Win, T. Yoshihisa, Y. Ishi, T. Kawakami, Y. Teranishi, and S. Shimojo, "Lightweight and secure certificateless multi-receiver encryption based on ECC," *J. Inf. Process.*, vol. 26, pp. 612–624, Jan. 2018.
[39] 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.
[40] H. Li, X. Chen, L. Pang, and W. Shi, "Quantum attack-resistent certificateless multi-receiver signcryption scheme," *PLoS ONE*, vol. 8, no. 6, p. e49141, Jun. 2013.
[41] J. Zhu, L. L. Chen, X. Zhu, and L. Xie, "A new efficient certificateless multi-receiver public key encryption scheme," *Int. J. Comput. Sci. Issues*, vol. 13, no. 6, pp. 1–7, Nov. 2016.
[42] R. Gao, J. Zeng, and L. Z. Deng, "Efficient certificateless anonymous multi-receiver encryption scheme without bilinear parings," *Math. Problems Eng.*, vol. 2018, 2018, Art. no. 1486437.
[43] S. Ullah, L. Marcenaro, and B. Rinner, "Secure smart cameras by aggregate-signcryption with decryption fairness for multi-receiver IoT applications," *Sensors*, vol. 19, no. 2, p. 327, Jan. 2019.
[44] L. Pang, H. Li, L. He, A. Alramadhan, and Y. M. Wang, "Secure and efficient lightweight RFID authentication protocol based on fast tag indexing," *Int. J. Commun. Syst.*, vol. 27, no. 11, pp. 3244–3254, Nov. 2014.



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. He was a Visiting Scholar with the Department of Computer Science, Wayne State University, USA. He is currently a Full Professor with the State Key Laboratory of Integrated Services Networks, School of Life Science and Technology, Xidian University. His research interests include the Internet security, cryptography, secure mobile agent systems, and e-commerce security technology.

MAN KOU was born in 1993. She is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University, China. Her research interest includes network and information security.



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.