IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 31, 2018, accepted November 28, 2018, date of publication December 4, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2884798

Efficient Anonymous Certificateless Multi-Receiver Signcryption Scheme Without Bilinear Pairings

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 Foundation of China under Grant 61473214 and Grant 61103178, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM6064, and in part by the National Cryptography Development Fund under Grant MMJJ20170208.

ABSTRACT Certificateless multi-receiver encryption/signcryption (CLME/CLMS) has become a research hotspot in the field of information security. Almost all of the existing CLME/CLMS schemes are constructed based on the bilinear pairing computation, a time-consuming operation, which makes their computational efficiency relatively low. Although there are some CLME schemes constructed on scalar point multiplications on elliptic curve cryptography (ECC) instead of the bilinear pairing computation, too many scalar point multiplications involved still lead to the low computational efficiency. Therefore, there is still room for the CLME/CLMS schemes in efficiency. Motivated by these concerns, an efficient anonymous certificateless multi-receiver signcryption scheme is proposed with its security proved under the random oracle model. The proposed scheme is improved largely in computational efficiency by the idea that it is designed based on scalar point multiplications on ECC instead of the bilinear pairing and the number of scalar point multiplications on ECC is reduced as small as possible.

INDEX TERMS Certificateless cryptography, computational efficiency, elliptic curve cryptography, multi-receiver signcryption.

I. INTRODUCTION

Multi-receiver encryption/signcryption has been considered as an effective and promising way to achieve one-to-many secure communication. The first identity-based multi-receiver encryption (MIBE) scheme was brought forward by Baek *et al.* [1] in 2005. Afterwards, in order to ensure the ciphertext's validity, combining MIBE with Zheng's signcryption [2], Duan and Cao [3] proposed the first multi-receiver identity-based signcryption (MIBS) scheme and gave the unforgeability security model at the same time. Since then, a large number of MIBS schemes [4]–[9], which are suitable for network conferences, paid-TV system and ad-hoc networks, have been proposed.

With the penetration of the Internet in all aspects of our daily life, people are increasingly focusing on their own privacy. For example, when watching a paid-TV program, people may not want others to know the specific program that they are watching, which belongs to their own privacy. Based on this practical need, introducing the receiver anonymity to MIBE, Fan *et al.* [10] put forward the first anonymous MIBE scheme by utilizing Lagrange interpolating polynomial. Unfortunately, both Wang *et al.* [11] and Chien [12] later prove that Fan *et al.*'s scheme fails to achieve the receiver anonymity as they have claimed. Afterwards, a new anonymous MIBS scheme was proposed by Pang and Li [13], in which the concept of decryption fairness is used to describe the characterization and enhancement of the receiver anonymity, but the receiver anonymity is not achieved due to the use of Lagrange interpolating polynomial, either. To truly achieve the receiver anonymity, in 2014, Tseng *et al.* [14] proposed another anonymous MIBE scheme, in which the receiver anonymity is realized by a modular large prime polynomial and the method is considered as one of the most effective ways to achieve the receiver anonymity so far.

Nevertheless, there exits the terrible phenomenon that the number of the involved bilinear pairing operations grows linearly with the number of receivers in Tseng *et al.*'s scheme, which leads to it extremely low in computational efficiency. To further improve efficiency, security and performance, there are a few following anonymous MIBE/MIBS schemes [15]–[19] proposed.

However, for schemes [10]–[19] above, there exists the key escrow problem, which is inherent in all ID-based schemes and means that the key generation center (KGC) could obtain the user's complete private key. Aiming at this problem, Al-Riyami and Paterson [20] put forward the certificateless public key cryptography, in which not only the key escrow problem in ID-based cryptography (IBC) is solved because the user's private key is generated by the user and the key generation center (KGC) together and KGC cannot obtain the user's complete private key, but also the advantage of no certificate in IBC is preserved, which marks the birth of a new cryptosystem, says the certificateless cryptosystem. Subsequently, many certificateless encryption/signcryption schemes [21]–[24] were proposed one by one. Selvi *et al.* proposed a certificateless multi-receiver signcryption (CLMS) scheme [25] in 2008 and its improved version [26] in 2009. Although it has been later proved by Miao *et al.* [27] that Selvi *et al.*'s scheme [26] cannot satisfy message confidentiality under external attacker's attack, it has raised the research upsurge on certificateless multi-receiver encryption/signcryption schemes. In 2005, Islam *et al.* [28] proposed an anonymous certificateless multi-receiver encryption (ACLME) scheme, which does not use the bilinear pairing operations but utilizes scalar point multiplications on elliptic curve cryptography (ECC) and makes an auspicious start in reducing the computational burden. However, the number of the involved scalar point multiplications on ECC remains to be reduced in Islam *et al.*'s scheme. At the same year, Hung *et al.* [29] proposed another ACLME scheme. Unfortunately, Hung *et al.*'s scheme still needs to be improved in efficiency as a result of the use of the bilinear pairing and map-to-point (MTP) hash function, another time-consuming operation. To improve Hung *et al.*'s scheme in efficiency, a new ACLME scheme was proposed by He *et al.* [30]. Regrettably, in He *et al.*'s scheme, it is found that although the bilinear pairing and MTP hash operations are not utilized, the number of the involved scalar point multiplications on ECC is still big, which affects the scheme in computational efficiency.

Further pursuing efficiency and lightweight, other certificateless multi-receiver encryption (CLME) schemes [31], [32] were proposed, successively. Nevertheless, although they are improved in encryption efficiency, the computation efficiency of their decryption process is bad, because the decryption process of scheme [31] utilizes the bilinear pairing and that of scheme [32] utilizes too many scalar point multiplications on ECC. Besides, it is worth noting that schemes [28]–[32] do not provide the sender with signature function, which is impossible to resist the attacker's forgery attack. At present, many certificateless multi-receiver (CLMR) schemes based on applications such as healthcare system [33] and IOT [34] have been proposed, yet they still remain to be improved in efficiency due to the use of the bilinear pairing. Besides, what are worth noting is that schemes [25], [26], [29]–[31], [33], [34] do not achieve decryption fairness and own partial private key verifiability, which fails them to avoid malicious KGC attacks.

To sum up, lots of CLME/CLMS researchers have been pursuing perfection in the computational efficiency, nevertheless, the computational performance of these schemes still remains to be improved. Motivated by those concerns, an efficient anonymous certificateless multi-receiver signcryption scheme without bilinear pairings is proposed, in which not only it is improved in efficiency by using scalar point multiplications on ECC instead of the bilinear pairing and limiting the number of the involved scalar point multiplications as small as possible, but also more security functions have been achieved such as decryption fairness, signature and partial private key verifiability. At the same time, it is proved that the proposed scheme satisfies 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 proposed scheme is minutely described. Besides, Section IV makes an analysis of correctness and security about the proposed scheme. Then the comparison between the proposed scheme and the existing CLME/CLMS schemes in terms of efficiency and functions is given in Section V. Finally, Section VI makes a conclusion about this paper.

II. PRELIMINARIES

In this section, we will give the hard problems, algorithm model and security models related to our proposed scheme.

A. HARD PROBLEMS

We define that p is a large prime number, G_p is the addition cycle group of points on ECC, Z_p^* is a nonzero multiplicative group based on p and P is one generator of G_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 is called a CDHP.

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

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

CDHP *assumption.* For any PPT algorithm A, $\mathrm{Adv}_A^{\mathrm{CDHP}}(k)$ is negligible.

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

Definition 2: The probability advantage that ECDLP can be solved by any probabilistic polynomial time (PPT) algorithm

B is defined as

$$\mathrm{Adv}_B^{\mathrm{ECDLP}}(k) = \Pr[B(P, xP) = x].$$

ECDLP *assumption.* For any PPT algorithm B, $\mathrm{Adv}_B^{\mathrm{ECDLP}}(k)$ 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-Public-Key*, *Set-Private-Key*, *Anony-Signcryption* and *De-Signcryption*, is shown as follows:

Setup: With a security parameter η as input, KGC runs the algorithm to get the master key s and the system's public parameters *Params*, and publishes *Params* while saving s.

Set-Secret-Value: With his/her own identity information ID as input, the user runs the algorithm to get his/her own secret value v_{ID} and secret value parameter V_{ID}.

Extract-Partial-Private-Key: With the master key s, the system's public parameters *Params*, and the user's identity information ID and secret value parameter V_{ID} as input, KGC runs the algorithm to get the user's partial private key y_{ID} and partial public key D_{ID}.

Set-Public-Key: With the system's public parameters *Params*, and his/her own identity information ID, partial private key y_{ID} and partial public key D_{ID} as input, the user runs the algorithm to get his/her own public key $\mathrm{PK}_{\mathrm{ID}}$.

Set-Private-Key: With the system's public parameters *Params*, and his/her own identity information ID, partial private key y_{ID}, public key $\mathrm{PK}_{\mathrm{ID}}$ and secret value v_{ID} as input, the user runs the algorithm to get his/her own private key $\mathrm{SK}_{\mathrm{ID}}$.

Anony-Signcryption: With the system's public parameters *Params*, a plaintext m, the authorized receivers' public key PK_i and his/her own private key SK_S as input, the sender runs the algorithm to generate the ciphertext C = Anony-Signcryption (*Param s*, m, PK_i, SK_S).

De-Signcryption: With the system's public parameters *Params*, the ciphertext C and his/her own private key SK_i as input, every authorized receiver runs the algorithm to get the plaintext m = De-Signcryption (SK_i, C, *Paramss*) and uses the sender's public key PK_S to verify the plaintext's source.

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 (A_I) and Type II adversary (A_{II}) respectively [20] in every security model. A_I means a malicious user who does not know the master key s, but he/she is allowed to replace the user's public key, while A_{II} means an honest-but-curious KGC who knows the master key s, but he/she 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) [25]. IND-CLMS-CCA against A_I (IND-CLMS-CCA-I) and IND-CLMS-CCA against A_{II} (IND-CLMS-CCA-II) will be described by *Game* 1 and *Game* 2, respectively.

***Game* 1** (IND-CLMS-CCA-I): The game is the interaction between the challenger B and A_I under IND-CLMS-CCA, and the specific steps are shown as follows:

Setup: B runs this algorithm to generate the master key s and the system's public parameter *Params*, and then sends *Params* to A_I while keeping s secret. Upon receiving *Params*, A_I outputs a group of target identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$, where n denotes a positive integer.

Phase 1: A_I asks B for a series of adaptive queries, and B responds accordingly:

Set-Secret-Value Query: A_I asks B for *Set-Secret-Value query* on ID. Upon receiving the query, B runs the *Set-Secret-Value* algorithm to get the user's secret value v_{ID} and returns it to A_I.

Extract-Partial-Private-Key Query: A_I asks B for *Extract-Partial-Private-Key query* on ID. Upon receiving the query, B runs the *Extract-Partial-Private-Key* algorithm to get the user's partial private key y_{ID} and returns it to A_I.

Set-Public-Key Query: A_I asks B for *Set-Public-Key query* on ID. Upon receiving the query, B runs the *Set-Public-Key* algorithm to get the user's public key $\mathrm{PK}_{\mathrm{ID}}$ and returns it to A_I.

Set-Private-Key Query: A_I asks B for *Set-Private-Key query* on ID. Upon receiving the query, B runs the *Set-Private-Key* algorithm to get the user's private key $\mathrm{SK}_{\mathrm{ID}}$ and returns it to A_I.

Public-Key-Replacement Query: A_I asks B for *Public-Key-Replacement query* on ID with $\mathrm{PK}'_{\mathrm{ID}}$. Upon receiving the query, B keeps $\mathrm{PK}'_{\mathrm{ID}}$ as the user's new public key.

Anony-Signcryption Query: A_I asks B for *Anony-Signcryption query* with the plaintext m and a series of identity information. Upon receiving the query, B randomly chooses an identity information ID_S, runs the *Anony-Signcryption* algorithm to generate the ciphertext C, and then sends C to A_I.

De-Signcryption Query: A_I asks B for *De-Signcryption query* with the ciphertext C. Upon receiving the query, B runs the *De-Signcryption* algorithm to get the plaintext m, verifies whether m is valid, and then returns m to A_I.

Challenge: A_I randomly chooses a pair of plaintext $< m_0, m_1 >$ with equal length, and sends them to B. Upon receiving $< m_0, m_1 >$, B randomly chooses a bit $\beta \in \{0, 1\}$ and generates the ciphertext C^* with the chosen plaintext m_β, then returns C^* to A_I.

Phase 2: A_I asks B for the same queries as *Phase* 1, but it should be noted that A_I cannot perform *Extract-Partial-Private-Key query* on target identities L and *De-Signcryption query* on C^*, and A_I cannot perform *Set-Private-Key query* on the target identity whose public key has been replaced, either.

Guess: A_I guesses a bit β^*. If $\beta^* = \beta$ holds, A_I wins the game. Otherwise, A_I fails. The probability advantage that A_I wins the game is defined as follows:

$$\mathrm{Adv}^{\text{IND-CLMS-CCA}}(A_I) = \left|2\Pr\left[\beta^* = \beta\right] - 1\right|.$$

Definition 4: If for any A_I under IND-CLMS-CCA, the probability advantage of winning *Game* 1 within time τ meets $\mathrm{Adv}^{\text{IND-CLMS-CCA}}(A_I) \leq \varepsilon$, the scheme is said to be (τ, ε)-IND-CLMS-CCA-I secure, where τ is the polynomial running time and ε is the non-negligible probability advantage.

***Game* 2** (IND-CLMS-CCA-II): The game is the interaction between the challenger B and A_{II} under IND-CLMS-CCA, and the specific steps are shown as follows:

Setup: B runs this algorithm to generate the master key s and the system's public parameter *Params*, and then sends *Params* and s to A_{II}. Upon receiving *Params* and s, A_{II} outputs a group of target identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$, where n denotes a positive integer.

Phase 1: A_{II} asks B for the same adaptive queries as *Phase* 1 in *Game* 1, and B responds accordingly. But it should be noted that A_{II} cannot perform *Public-Key-Replacement query*.

Challenge: A_{II} randomly chooses a pair of plaintext $< m_0, m_1 >$ with equal length, and sends them to B. Upon receiving $< m_0, m_1 >$, B randomly chooses a bit $\beta \in \{0, 1\}$ and generates the ciphertext C^* with the chosen plaintext m_β, then returns C^* to A_{II}.

Phase 2: A_{II} asks B for the same queries as *Phase* 1, but it should be noted that A_{II} cannot perform *Set-Secret-Value query* on target identities L and *De-Signcryption query* on C^*.

Guess: A_{II} guesses a bit β^*. If $\beta^* = \beta$ holds, A_{II} wins the game. Otherwise, A_{II} fails. The probability advantage that A_{II} wins the game is defined as follows:

$$\mathrm{Adv}^{\text{IND-CLMS-CCA}}(A_{II}) = \left|2\Pr\left[\beta^* = \beta\right] - 1\right|$$

Definition 5: If for any A_{II} under IND-CLMS-CCA, the probability advantage of winning *Game* 2 within time τ meets $\mathrm{Adv}^{\text{IND-CLMS-CCA}}\left(A_{II}\right) \leq \varepsilon$, the scheme is said to be (τ, ε)-IND-CLMS-CCA-II secure, where τ is the polynomial running time and ε is the non-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 (SUF-CLMS-CPA) [25]. SUF-CLMS-CPA against A_I (SUF-CLMS-CPA-I) and SUF-CLMS-CPA againstA_{II} (SUF-CLMS-CPA-II) will be described by *Game* 3 and *Game* 4, respectively.

***Game* 3** (SUF-CLMS-CPA-I): The game is the interaction between the challenger B and A_I under SUF-CLMS-CPA, and the specific steps are shown as follows:

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

Attack: A_I asks B for the same adaptive queries as *Phase* 1 in *Game* 1, and B responds accordingly.

Forgery: A_I forges a new ciphertext C^* with a group of target identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$and a plaintext m. If the ciphertext C^* can be decrypted correctly by any receiver in L, A_I wins the game. Otherwise, A_I fails. But it should be noted that C^* cannot be generated by the *Anony-Signcryption query* and other restrictions are the same as *Phase* 2 in *Game* 1.

Definition 6: If for any A_I under SUF-CLMS-CPA, the probability advantage of winning *Game* 3 within time τ meets $\mathrm{Adv}^{\text{SUF-CLMS-CPA}}\left(A_I\right) \leq \varepsilon$, the scheme is said to be (τ, ε)-SUF-CLMS-CPA-I secure, where τ is the polynomial running time and ε is the non-negligible probability advantage.

***Game* 4** (SUF-CLMS-CPA-II): The game is the interaction between the challenger B and A_{II} under SUF-CLMS-CPA, and the specific steps are shown as follows:

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

Attack: A_{II} asks B for the same adaptive queries as *Phase* 1 in *Game* 2, and B responds accordingly.

Forgery: A_{II} forges a new ciphertext C^* with a group of target identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$and a plaintext m. If the ciphertext C^* can be decrypted correctly by any receiver in L, A_{II} wins the game. Otherwise, A_{II} fails. But it should be noted that C^* cannot be generated by the *Anony-Signcryption query* and other restrictions are the same as *Phase* 2 in *Game* 2.

Definition 7: If for any A_{II} under SUF-CLMS-CPA, the probability advantage of winning *Game* 4 within time τ meets $\mathrm{Adv}^{\text{SUF-CLMS-CPA}}\left(A_{II}\right) \leq \varepsilon$, the scheme is said to be (τ, ε)-SUF-CLMS-CPA-II secure, where τ is the polynomial running time and ε is the non-negligible probability advantage.

3) RECEIVER ANONYMITY

The receiver anonymity model of the proposed scheme is called the anonymous indistinguishability of certificateless signcryption against selective multi-receiver, chosen ciphertext attack (ANON-CLMS-CCA) [28]. ANON-CLMS-CCA against A_I (ANON-CLMS-CCA-I) and ANON-CLMS-CCA against A_{II} (ANON-CLMS-CCA-II) will be described by *Game* 5 and *Game* 6, respectively.

***Game* 5**(ANON-CLMS-CCA-I): The game is the interaction between the challenger B and A_I under ANON-CLMS-CCA, and the specific steps are shown as follows:

Setup: B runs this algorithm to generate the master key s and the system's public parameter *Params*, and then sends *Params* to A_I while keeping s secret. Upon receiving *Params*, A_I outputs a group of target identities$L = \{\mathrm{ID}_0, \mathrm{ID}_1\}$.

Phase 1: A_I asks B for the same adaptive queries as *Phase* 1 in *Game* 1, and B responds accordingly.

Challenge: A_I chooses a plaintext m and a group of target identities $L^* = \{\mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, and sends them to B. Upon receiving m and L^*, B randomly chooses a bit $e \in \{0, 1\}$

and generates the ciphertext C^* with a group of new target identities $L^{**} = \{\mathrm{ID}_e, \mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, then returns C^* to A_I.

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

Guess: A_I guesses a bit e^*. If $e^* = e$ holds, A_I wins the game. Otherwise, A_I fails. The probability advantage that A_I wins the game is defined as follows:

$$\mathrm{Adv}^{\text{ANON-CLMS-CCA}}(A_I) = \left|2\Pr\left[e^* = e\right] - 1\right|$$

Definition 8: If for any A_I under ANON-CLMS-CCA, the probability advantage of winning *Game* 5 within time τ meets $\mathrm{Adv}^{\text{ANON-CLMS-CCA}}(A_I) \le \varepsilon$, the scheme is said to be (τ, ε)-ANON-CLMS-CCA-I secure, where τ is the polynomial running time and ε is the non-negligible probability advantage.

Game 6(ANON-CLMS-CCA-II): The game is the interaction between the challenger B and A_{II} under ANON-CLMS-CCA, and the specific steps are shown as follows:

Setup: B runs this algorithm to generate the master key s and the system's public parameter *Params*, and then sends *Params* and s to A_{II}. Upon receiving *Params* and s, A_{II} outputs a set of target identities $L = \{\mathrm{ID}_0, \mathrm{ID}_1\}$.

Phase 1: A_{II} asks B for the same adaptive queries as *Phase* 1 in *Game* 2, and B responds accordingly.

Challenge: A_{II} chooses a plaintext m and a group of target identities $L^* = \{\mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, and sends them to B. Upon receiving m and L^*, B randomly chooses a bit $e \in \{0, 1\}$ and generates the ciphertext C^* with a group of new target identities $L^{**} = \{\mathrm{ID}_e, \mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, then returns C^* to A_{II}.

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

Guess: A_{II} guesses a bit e^*. If $e^* = e$ holds, A_{II} wins the game. Otherwise, A_{II} fails. The probability advantage that A_{II} wins the game is defined as follows:

$$\mathrm{Adv}^{\text{ANON-CLMS-CCA}}(A_{II}) = \left|2\Pr\left[e^* = e\right] - 1\right|$$

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

III. THE PROPOSED SCHEME

The participants of the proposed scheme consist of KGC, the sender S and a set of authorized receivers, $R_1, R_2, \ldots, R_n$, where n is the number of authorized receivers decided by the sender. And the specific scheme includes *Setup algorithm*, *Key Extract algorithm*, *Anony-Signcryption algorithm* and *De-Signcryption algorithm*, shown as follows:

A. SETUP ALGORITHM

Setup algorithm is run by KGC to generate the master key and the system's public parameters, shown as follows:

1) With a security parameter η as input, KGC randomly chooses a prime integer p ($q \ge 2^k$, k is a long integer.), generates an elliptic curve E defined on finite field F_p, and chooses an additive cyclic group G_p on E and its generator P.

2) KGC randomly chooses an integer $s \in Z_p^*$ as the master key and computes $P_{\text{pub}} = sP$ as system's public key.

3) KGC chooses five secure hash functions: H_0:$\{0,1\}^* \times G_p \times Z_p^* \to Z_p^*$; H_1:$\{0,1\}^* \times G_p \to Z_p^*$; H_2:$G_p \times G_p \to Z_p^*$; H_3:$Z_p^* \to Z_p^*$; H_4: $\{0,1\}^* \times Z_p^* \times Z_p^* \times \ldots \times Z_p^* \times G_p \to Z_p^*$.

4) KGC chooses a symmetric encryption function E_k and the corresponding decryption function D_k (such as AES), where k is the symmetric key.

5) KGC publishes the system's public parameters $Params =< p, F_p, E, G_p, P, P_{\text{pub}}, E_k, D_k, H_0, H_1, H_2, H_3, H_4 >$ and keeps the master key s secret.

B. KEY EXTRACT ALGORITHM

Key Extract algorithm is run by the user and KGC together to generate the user's public key and private key, shown as follows:

1) *Set-Secret-Value Algorithm:* The user with the identity information ID_i randomly chooses an integer $v_i \in Z_p^*$, computes $V_i = v_i P$, and sends V_i and his/her own identity information ID_i to KGC through a public channel, where v_i is the user's secret value and V_i is the user's secret value parameter.

2) *Extract-Partial-Private-Key Algorithm:* Upon receiving V_i and ID_i from the user, KGC randomly chooses an integer $d_i \in Z_p^*$ and computes $y_i = H_0(\mathrm{ID}_i, V_i, d_i) + s (\mathrm{mod}\ p)$ and $D_i = H_0(\mathrm{ID}_i, V_i, d_i)P$. Then, KGC sends y_i to the user through a secure channel, and sends D_i to the user through a public channel, where y_i is the user's partial private key, and D_i is the user's partial public key.

3) *Set-Public-Key Algorithm:* Upon receiving y_i and D_i from KGC, the user checks whether the equation $y_i P = D_i + P_{\text{pub}}$ holds. If yes, the user accepts the partial private key y_i and the partial public key D_i, and computes $\mathrm{PK}_i = D_i + H_1(\mathrm{ID}_i, V_i)V_i$ as his/her own public key. Then, the user sends PK_i to KGC for publication. Otherwise, the user rejects the partial private key y_i and the partial public key D_i.

4) *Set-Private-Key Algorithm:* The user computes $\mathrm{SK}_i = H_1(\mathrm{ID}_i, \mathrm{PK}_i)(y_i + H_1(\mathrm{ID}_i, V_i)v_i)(\mathrm{mod} p)$ as his/her own private key.

C. ANONY-SIGNCRYPTION ALGORITHM

With the system's public parameters *Params*, the sender's private key SK_S and the plaintext m as input, the sender S chooses a set of receivers with their identities information $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$ and signcrypts m as follows:

1) Compute $Q_i = \mathrm{PK}_i + P_{\text{pub}}$, where $i = 1, 2, \ldots, n$;

2) Randomly choose an integer $w \in Z_p^*$, and compute $W = wP$, $F_i = wH_1(\mathrm{ID}_i, \mathrm{PK}_i)Q_i$ and $\alpha_i = H_2(F_i, W)$, where $i = 1, 2, \ldots, n$;

3) Randomly choose an integer $\xi \in Z_p^*$ and compute the polynomial

$$f(x) = \prod_{i=1}^{n}(x-\alpha_i)+\xi(\bmod p)$$
$$= a_0 + a_1 x + ... + a_{n-1}x^{n-1} + x^n, \quad a_i \in Z_p^*;$$

4) Compute $k = H_3(\xi)$, $J = E_k(m||\mathrm{ID}_S)$ and $h = H_4(m||\mathrm{ID}_S, \xi, a_0, a_1, \ldots, a_{n-1}, W)$;

5) Compute h^{-1} to make it satisfy the equation $hh^{-1} \equiv 1 \bmod p$, and compute $z = h^{-1}(\mathrm{SK}_S + w)(\bmod p)$;

6) Generate the ciphertext $C =< J, W, z, h, a_0, a_1, \ldots, a_{n-1} >$ and broadcast it to receivers.

D. DE-SIGNCRYPTION ALGORITHM

Upon receiving the ciphertext $C =< J, W, z, h, a_0, a_1, \ldots, a_{n-1} >$, each receiver can decrypt C with his/her own private key SK_i and the system's public parameters *Params* as follows:

1) Compute $F_i = \mathrm{SK}_i W$ and $\alpha_i = H_2(F_i, W)$;

2) Compute $f(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} + x^n$ and $\xi = f(\alpha_i)$;

3) Compute $k = H_3(\xi)$ and $m||\mathrm{ID}_S = D_k(J)$;

4) Compute $h' = H_4(m||\mathrm{ID}_S, \xi, a_0, a_1, \ldots, a_{n-1}, W)$, and check whether the equation $h' = h$ holds. If yes, the receiver continues with the following steps. Otherwise, the receiver rejects m and exits the de-signcryption process.

5) The receiver obtains the sender S's public key PK_S, and judges whether the equation $hzP = H_1(\mathrm{ID}_S, \mathrm{PK}_S)(\mathrm{PK}_S + P_{\mathrm{pub}}) + W$ holds. If yes, the receiver accepts the plaintext m and exits the de-signcryption process. Otherwise, the receiver rejects m and exits the de-signcryption process.

IV. CORRECTNESS AND SECURITY PROOFS

A. CORRECTNESS ANALYSIS

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

Proof: The correctness of the user's partial private key verification is guaranteed by the establishment of the equation $y_i P = D_i + P_{\mathrm{pub}}$, and the deduction that the equation holds is shown as follows:

$$\begin{aligned} y_i P &= (H_0(ID_i, V_i, d_i) + s)P \\ &= H_0(ID_i, V_i, d_i)P + P_{pub} \\ &= D_i + P_{pub} \end{aligned}$$

Through the above derivation, it can be seen that the equation $y_i P = D_i + P_{\mathrm{pub}}$ holds. As a result, the verification of the user's partial private key in *Key Extract Algorithm* is correct.

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

Proof: The correctness of *De-Signcryption algorithm* is guaranteed by establishments of equations $h' = h$ and $hzP = H_1(\mathrm{ID}_S, \mathrm{PK}_S)(\mathrm{PK}_S + P_{\mathrm{pub}}) + W$, and deductions that these two equations hold are shown in the following 1) and 2), respectively.

1) For every receiver R_i, with the ciphertext C, he/she has $F_i = \mathrm{SK}_i W$ and $\alpha_i = H_2(F_i, W)$. Then, with α_i, he/she can compute $\xi = f(\alpha_i)$, and then get $k = H_3(\xi)$ and $m||\mathrm{ID}_S = D_k(J)$. Finally, he/she has $h' = H_4(m||\mathrm{ID}_S, \xi, a_0, a_1, \ldots, a_{n-1}, W)$. So, the equation $h' = h$ holds.

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

$$\begin{aligned} hzP &= hh^{-1}(SK_S + w)P \\ &= SK_S P + W \\ &= H_1(ID_S, \mathrm{PK}_S)(y_S + H_1(ID_S, V_S)v_S)P + W \\ &= H_1(ID_S, \mathrm{PK}_S)(D_S + H_1(ID_S, V_S)V_S + P_{\mathrm{pub}}) + W \\ &= H_1(ID_S, \mathrm{PK}_S)(\mathrm{PK}_S + P_{\mathrm{pub}}) + W \end{aligned}$$

That is to say, the equation $hzP = H_1(\mathrm{ID}_S, \mathrm{PK}_S)(\mathrm{PK}_S + P_{\mathrm{pub}}) + W$ holds.

Through the derivations of 1) and 2) above, it can be seen that equations $h' = h$ and $hzP = H_1(\mathrm{ID}_S, \mathrm{PK}_S)(\mathrm{PK}_S+P_{\mathrm{pub}})+W$ hold. As a result, the *De-Signcryption algorithm* is correct.

B. SECURITY PROOFS

Based on security models in Section II, the specific security proofs of the proposed scheme are shown below. In *Theorem* 3 and *Theorem* 4, we shall prove that the proposed scheme can achieve IND-CLMS-CCA-I/II security. In *Theorem* 5 and *Theorem* 6, we shall prove that the proposed scheme can achieve SUF-CLMS-CPA-I/II security. In *Theorem* 7 and *Theorem* 8, we shall prove that the proposed scheme can achieve ANON-CLMS-CCA-I/II security.

Theorem 3: IND-CLMS-CCA-I. Under IND-CLMS-CCA, if there is an adversary A_I who can win *Game* 1 in polynomial running time τ with a non-negligible probability advantage ε (A_I can ask for at most q_i Hash queries H_i (i = 0,1,2,3,4), q_c Key queries, q_e Set-Secret-Value queries, q_b Extract-Private-Key queries, q_p Set-Public-Key queries, q_k Set-Private-Key, q_r Public-Key-Replacement queries, q_a Anony-Signcryption queries and q_d De-Signcryption queries.), the challenger B can solve CDHP by interacting with the adversary A_I in time $\tau' \leq \tau + (2q_c + 3q_d)\, O(\tau_s)$ with a non-negligible probability advantage $\varepsilon' \geq 2(\varepsilon - q_d q_4 / 2^k) / nq_2$, where τ_s is the time of an elliptic curve scalar point multiplication operation.

Proof: Assume that an adversary A_I can attack the IND-CLMS-CCA security with a non-negligible probability advantage ε and ask the challenger B for a series of queries under the random oracle model. Given a set of elements $< P, aP, bP >$, the challenger B computes abP to solve CDHP by interacting with the adversary A_I within a time bounded polynomial. And the interaction between the challenger B and the adversary A_I is shown as follows:

Setup: B runs this algorithm to generate the master key $s = a \in Z_p^*$ and the system's public parameter $Params =< F_p, E, G_p, P, P_{\mathrm{pub}} = aP, E_k, D_k, p, H_0, H_1, H_2, H_3, H_4 >$, and then sends *Params* to A_I while keeping s secret. Upon receiving *Params*, A_I outputs a group of target identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$, where n denotes a positive integer. It should be noted that H_0, H_1, H_2, H_3 and H_4 are random oracles controlled by B, and the random oracles interactions between A_I and B are shown as follows:

1) H_0 hash query: With the tuple $< \mathrm{ID}_j, V_j, d_j >$ as input, A_I asks B for H_0 hash query. Upon receiving the query, B checks whether the tuple $< \mathrm{ID}_j, V_j, d_j, \mu_j >$ is in list L_0. If yes, B returns μ_j to A_I. Otherwise, B randomly chooses an integer $\mu_j \in Z_p^*$ and returns it to A_I. Meanwhile, B updates the tuple $< \mathrm{ID}_j, V_j, d_j, \mu_j >$ in list L_0.

2) H_1 hash query: With tuples $< \mathrm{ID}_j, V_j >$ and $<\mathrm{ID}_j, \mathrm{PK}_j >$ as input, A_I asks B for H_1 hash query. Upon receiving the query, B checks whether the tuples $< \mathrm{ID}_j, V_j, \theta_j >$ and $<\mathrm{ID}_j, \mathrm{PK}_j, \delta_j >$ are in list L_1. If yes, B returns θ_j and δ_j to A_I. Otherwise, B randomly chooses two integers $\theta_j, \delta_j \in Z_p^*$ and returns them to A_I. Meanwhile, B updates tuples $< \mathrm{ID}_j, V_j, \theta_j >$ and $<\mathrm{ID}_j, \mathrm{PK}_j, \delta_j >$ in list L_1.

3) H_2 hash query: With the tuple $< F_j, W_j >$ as input, A_I asks B for H_2 hash query. Upon receiving the query, B checks whether the tuple $< F_j, W_j, \alpha_j >$ is in list L_2. If yes, B returns α_j to A_I. Otherwise, B randomly chooses an integer $\alpha_j \in Z_p^*$ and returns it to A_I. Meanwhile, B updates the tuple $< F_j, W_j, \alpha_j >$ in list L_2.

4) H_3 hash query: With the tuple $< \xi_j >$ as input, A_I asks B for H_3 hash query. Upon receiving the query, B checks whether the tuple $< \xi_j, k_j >$ is in list L_3. If yes, B returns k_j to A_I. Otherwise, B randomly chooses an integer $k_j \in Z_p^*$ and returns it to A_I. Meanwhile, B updates the tuple $< \xi_j, k_j >$ in list L_3.

5) H_4 hash query: With the tuple $< m_j||\mathrm{ID}_S, \delta_j, a_{j,0}, a_{j,1}, \ldots, a_{j,n-1}, W_j >$ as input, A_I asks B for H_4 hash query. Upon receiving the query, B checks whether the tuple $< m_j||\mathrm{ID}_S, \delta_j, a_{j,0}, a_{j,1}, \ldots, a_{j,n-1}, W_j, h_j >$ is in list L_4. If yes, B returns h_j to A_I. Otherwise, B randomly chooses an integer $h_j \in Z_p^*$ and returns it to A_I. Meanwhile, B updates the tuple $< m_j||\mathrm{ID}_S, \delta_j, a_{j,0}, a_{j,1}, \ldots, a_{j,n-1}, W_j, h_j >$ in list L_4.

Phase 1: A_I asks B for a series of adaptive queries, and B responds accordingly as follows:

1) Key query: B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ is in list L_C. If yes, B keeps the tuple. Otherwise, B performs as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B randomly chooses two integers $d_j, v_j \in Z_p^*$, sets $V_j = v_jP$ and $\mathrm{SK}_j \leftarrow \perp$, computes $D_j = H_0(\mathrm{ID}_i, V_i, d_i)P$ and $\mathrm{PK}_j = D_j + H_1(\mathrm{ID}_j, V_j)V_j$, and then updates tuples $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ in list L_C and $< \mathrm{ID}_j, V_j, \theta_j >$ in list L_1, respectively.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B randomly chooses two integers $y_j, v_j \in Z_p^*$, sets $V_j = v_jP$, computes $D_j = y_jP - P_{\mathrm{pub}}$, $\mathrm{PK}_j = D_j + H_1(\mathrm{ID}_j, V_j)V_j$ and $\mathrm{SK}_j = H_1(\mathrm{ID}_j,\mathrm{PK}_j)(y_j + H_1(\mathrm{ID}_j, V_j)v_j)$ (mod$p$), and then updates tuples $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, d_j, v_j, y_j >$ in list L_C and $< \mathrm{ID}_j, V_j, \theta_j >$ in list L_1, respectively.

2) Set-Secret-Value query: A_I asks B for Set-Secret-Value query on ID_j. Upon receiving the query, B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ is in list L_C. If yes, B returns v_j to A_I. Otherwise, B performs key query to obtain the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$, and returns v_j to A_I.

3) Extract-Partial-Private-Key query: A_I asks B for Extract-Partial-Private-Key query on ID_j. Upon receiving the query, B responds as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B returns "failure" to A_I.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, d_j, v_j, y_j >$ is in list L_C. If yes, B returns y_j to A_I. Otherwise, B performs key query to obtain the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ and returns y_j to A_I.

4) Set-Public-Key query: A_I asks B for Set-Public-Key query on ID_j. Upon receiving the query, B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ is in list L_C. If yes, B returns PK_j to A_I. Otherwise, B performs key query to obtain the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ and returns PK_j to A_I.

5) Set-Private-Key query: A_I asks B for Set-Private-Key query on ID_j. Upon receiving the query, B responds as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B returns "failure" to A_I.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, d_j, v_j, y_j >$ is in list L_C. If yes, B returns SK_j to A_I. Otherwise, B performs key query to obtain the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ and returns SK_j to A_I.

6) Public-Key-Replacement query: A_I asks B for Public-Key-Replacement query on ID_j with PK'_j. Upon receiving the query, B searches for the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, d_j, v_j, y_j >$ in list L_C and replaces PK_j with PK'_j. Then, B updates the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ in list L_C.

7) Anony-Signcryption query: A_I asks B for Anony-Signcryption query on the plaintext m and the identity information ID_S. Upon receiving the query, B judges whether $\mathrm{ID}_S \neq \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$. If yes, B performs Set-Private-Key query to obtain the private key SK_S, generates the ciphertext C, and returns C to A_I. Otherwise, B performs as follows:

a) Randomly choose an integer $w \in Z_p^*$, and compute $W = wP$, $F_j = wH_1(\mathrm{ID}_j, \mathrm{PK}_j)(\mathrm{PK}_j + P_{\mathrm{pub}})$ and $\alpha_j = H_2(F_j, W)$, where $j = 1, 2, \ldots, n$;

b) Randomly choose an integer $\xi \in Z_p^*$, and construct the polynomial

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

c) Compute $k = H_3(\xi)$, $J = E_k(m||\mathrm{ID}_S)$ and $h = H_4(m||\mathrm{ID}_S, \xi, a_0, a_1, \ldots, a_{n-1}, W)$;

d) Randomly choose an integer $z \in Z_p^*$;

e) Return the ciphertext $C =< J, W, z, h, a_0, a_1, \ldots, a_{n-1} >$ to A_I.

8) De-Signcryption query: A_I asks B for De-Signcryption query on the ciphertext C. Upon receiving the query, B randomly chooses an identity information ID_j, and judges whether $\mathrm{ID}_j = \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$. If yes, B returns "failure" to A_I. Otherwise, B performs as follows:

a) Search for the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j >$ in list L_C to obtain SK_j, and compute $F_j = \mathrm{SK}_jW$ and $\alpha_j = H_2(F_j, W)$;

b) Compute $f(x) = a_0 + a_1x + \ldots + a_{n-1}x^{n-1} + x^n$, and obtain ξ by $f(x)$ and α_j;

c) Compute $k = H_3(\xi)$ and $m||\mathrm{ID}_S = D_k(J)$;

d) Judge whether the equation $hzP = H_1(\mathrm{ID}_S,\mathrm{PK}_S)(\mathrm{PK}_S + P_{\mathrm{pub}}) + W$ holds. If yes, B returns m to A_I. Otherwise, B returns "failure" to A_I.

Challenge: A_I randomly chooses a pair of plaintext $< m_0, m_1 >$ with equal length, and sends them to B. Upon receiving $< m_0, m_1 >$, B randomly chooses a bit $\beta \in \{0, 1\}$ and generates the ciphertext C^* with the chosen plaintext m_β as follows:

a) Set $W_i = b\mathrm{PK}_i$, $F_i = b(\mathrm{PK}_i + P_{\mathrm{pub}})$ and $\alpha_i = H_2(F_i, W_i)$, where $i = 1, 2, \ldots, n$;

b) Randomly choose an integer $\xi \in Z_p^*$ and construct the polynomial

$$\begin{aligned} f(x) &= \prod_{j=1}^{n} \left(x - \alpha_j\right) + \xi (\mathrm{mod}\ p) \\ &= a_0 + a_1 x \ldots + a_{n-1}x^{n-1} + x^n,\ a_j \in Z_p^*; \end{aligned}$$

c) Compute $k = H_3(\xi)$, $J^* = E_k(m_\beta||\mathrm{ID}_S)$ and $h^* = H_4(m_\beta||\mathrm{ID}_S, \xi, a_0, a_1, \ldots, a_{n-1}, W_i)$;

d) Randomly choose an integer $z \in Z_p^*$;

e) Return the ciphertext $C^* =< J^*, W_i, z, h, a_0, a_1, \ldots, a_{n-1} >$ to A_I.

Phase 2: A_I asks B for the same queries as *Phase* 1, but it should be noted thatA_I cannot perform *De-Signcryption query* on C^*.

Guess: A_I guesses a bit β^*. If $\beta^* = \beta$ holds, A_I wins the game, and B outputs $abP = W_i$-F_i as the solution to CDHP. Otherwise, B outputs "failure".

Through the discussion above, it is concluded that during de-signcryption queries, H_4 hash could provide a valid ciphertext, so the probability that a valid ciphertext is rejected is not greater than ${}^{q_4}/_{2^k}$. Since A_I asks B for q_d de-signcryption queries during the attack process, the probability advantage that B decrypts the ciphertext successfully is $\varepsilon_d \geq \varepsilon - {}^{q_4 q_d}/_{2^k}$. And during the guess process, H_2 hash satisfies CDHP, so the correct probability that B computes abP is at least $\varepsilon_g = {}^{2}/_{nq_2}$. Therefore, the probability advantage that B can solve CDHP by interacting with the adversary A_I is $\varepsilon' \geq \varepsilon_d \varepsilon_g \geq 2(\varepsilon - q_d q_4 / 2^k) / nq_2$ within running time $\tau' \leq \tau + (2q_c + 3q_d)\, O(\tau_s)$, where τ_s is the time of an elliptic curve scalar point multiplication operation. ∎

Theorem 4: IND-CLMS-CCA-II. Under IND-CLMS-CCA, if there is an adversary A_{II} who can win *G*ame 2 in polynomial running time τ with a non-negligible probability advantage ε (A_{II} can ask for at most q_i Hash queries H_i (i = 0,1,2,3,4), q_c Key queries, q_e Set-Secret-Value queries, q_b Extract-Private-Key queries, q_p Set-Public-Key queries, q_k Set-Private-Key, q_a Anony-Signcryption queries and q_d De-Signcryption queries.), the challenger B can solve CDHP by interacting with the adversary A_{II} in time $\tau' \leq \tau + (3q_c + 3q_d)\, O(\tau_s)$ with a non-negligible probability advantage $\varepsilon' \geq 2(\varepsilon - q_d q_4 / 2^k) / nq_2$, where τ_s is the time of an elliptic curve scalar point multiplication operation.

Proof: Assume that an adversary A_{II} can attack the IND-CLMS-CCA security with a non-negligible probability advantage ε and ask the challenger B for a series of queries under the random oracle model. Given a set of elements $< P, aP, bP >$, the challenger B computes abP to solve CDHP by interacting with the adversary A_{II} within a time bounded polynomial. And the interaction between the challenger B and the adversary A_{II} is shown as follows:

Setup: B runs this algorithm to generate the master key $s \in Z_p^*$ and the system's public parameter $Params =< F_p, E, G_p, P, K = aP, P_{\mathrm{pub}}, E_k, D_k, p, H_0, H_1, H_2, H_3, H_4 >$, and then sends $Params$ and s to A_{II}, where $a \in Z_p^*$. Upon receiving $Params$ and s, A_{II} outputs a group of target identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$, where n denotes a positive integer. It should be noted that H_0, H_1, H_2, H_3 and H_4 are random oracles controlled by B, and the random oracles interactions between A_{II} and B are the same as *Setup* in ***Theorem* 3**.

Phase 1: A_{II} asks B for a series of adaptive queries, and B responds accordingly as follows:

1) Key query: B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ is in list L_C. If yes, B keeps the tuple. Otherwise, B performs as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B randomly chooses two integers d_j, $v_j \in Z_p^*$, computes $y_j = H_0(\mathrm{ID}_j, V_j, d_j) + s(\mathrm{mod} p)$ and $\mathrm{PK}_j = H_0(\mathrm{ID}_j, V_j, d_j)P + H_1(\mathrm{ID}_j, V_j)V_j$, and then updates tuples $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ in list L_C and $< \mathrm{ID}_j, V_j, \theta_j >$ in list L_1, respectively, where $V_j = v_jP$ and $\mathrm{SK}_j \leftarrow \perp$.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B randomly chooses two integers d_j, $v_j \in Z_p^*$, and computes $y_j = H_0(\mathrm{ID}_j, V_j, d_j) + s(\mathrm{mod} p)$, $\mathrm{PK}_j = H_0(\mathrm{ID}_j, V_j, d_j)P + H_1(\mathrm{ID}_j, V_j)V_j$ and $\mathrm{SK}_j = H_1(\mathrm{ID}_j, \mathrm{PK}_j)(y_j + H_1\ (\mathrm{ID}_j, V_j)v_j)(\mathrm{mod} p)$, and then updates tuples $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ in list L_C and $< \mathrm{ID}_j, V_j, \theta_j >$ in list L_1, respectively, where $V_j = v_jP$.

2) Set-Secret-Value query: A_{II} asks B for Set-Secret-Value query on ID_j. Upon receiving the query, B responds as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B returns "failure" to A_{II}.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ is in list L_C. If yes, B returns v_j to A_{II}. Otherwise, B performs key query to obtain the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$, and returns v_j to A_{II}.

3) Extract-Partial-Private-Key query: A_{II} asks B for Extract-Partial-Private-Key query on ID_j. Upon receiving the query, B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ is in list L_C. If yes, B returns y_j to A_{II}. Otherwise, B performs key query to obtain the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ and returns y_j to A_{II}.

4) Set-Public-Key query: A_{II} asks B for Set-Public-Key query on ID_j. Upon receiving the query, B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ is in list L_C. If yes, B returns PK_j to A_{II}. Otherwise, B performs key query to obtain the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ and returns PK_j to A_{II}.

5) Set-Private-Key query: A_{II} asks B for Set-Private-Key query on ID_j. Upon receiving the query, B responds as follows:

a) If $\mathrm{ID}_j = \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B returns "failure" to A_{II}.

b) If $\mathrm{ID}_j \neq \mathrm{ID}_i$, for $i = 1, 2, \ldots, n$, B checks whether the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ is in list L_C. If yes, B returns SK_j to A_{II}. Otherwise, B performs key query to obtain the tuple $<\mathrm{ID}_j, \mathrm{SK}_j, \mathrm{PK}_j, v_j, y_j>$ and returns SK_j to A_{II}.

6) Anony-Signcryption query: The step is the same as Anony-Signcryption query in ***Theorem* 3**.

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

Challenge: A_{II} randomly chooses a pair of plaintext $< m_0, m_1 >$ with equal length, and sends them to B. Upon receiving $< m_0, m_1 >$, B randomly chooses a bit $\beta \in \{0, 1\}$ and generates the ciphertext C^* with the chosen plaintext m_β as follows:

a) Set $W_i = b(\mathrm{PK}_i + Y)$, $F_i = b(\mathrm{PK}_i + P_{\mathrm{pub}})$ and $\alpha_i = H_2(F_i, W_i)$, where $Y = K + P_{\mathrm{pub}}$ and $i = 1, 2, \ldots, n$;

b) Randomly choose an integer $\xi \in Z_p^*$ and construct the polynomial

$$f(x) = \prod_{j=1}^{n} \left(x - \alpha_j\right) + \xi (\mathrm{mod}\ p)$$
$$= a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} + x^n, \quad a_j \in Z_p^*$$

c) Compute $k = H_3(\xi)$, $J^* = E_k(m_\beta || \mathrm{ID}_S)$ and $h^* = H_4(m_\beta || \mathrm{ID}_S, \xi, a_0, a_1, \ldots, a_{n-1}, W_i)$;

d) Randomly choose an integer $z \in Z_p^*$;

e) Return the ciphertext $C^* =< J^*, W_i, z, h, a_0, a_1, \ldots, a_{n-1} >$ to A_{II}.

Phase 2: A_{II} asks B for the same queries as *Phase* 1, but it should be noted that A_{II} cannot perform De-Signcryption query on C^*.

Guess: A_{II} guesses a bit β^*. If $\beta^* = \beta$ holds, A_{II} wins the game, and B outputs $abP = F_i$-W_i as the solution to CDHP. Otherwise, B outputs "failure".

Through the discussion above, it is concluded that during de-signcryption queries, H_4 hash could provide a valid ciphertext, so the probability that a valid ciphertext is rejected is not greater than $^{q_4}/_{2^k}$. Since A_{II} asks B for q_d de-signcryption queries during the attack process, the probability advantage that B decrypts the ciphertext successfully is $\varepsilon_d \geq \varepsilon - {}^{q_4 q_d}/_{2^k}$. And during the guess process, H_2 hash satisfies CDHP, so the correct probability that B computes abP is at least $\varepsilon_g = {}^2/_{nq_2}$. Therefore, the probability advantage that B can solve CDHP by interacting with the adversary A_{II} is $\varepsilon' \geq \varepsilon_d \varepsilon_g \geq 2(\varepsilon - q_d q_4 / 2^k) / nq_2$ within running time $\tau' \leq \tau + (3q_c + 3q_d)\, O(\tau_s)$, where τ_s is the time of an elliptic curve scalar point multiplication operation. ■

Theorem 5: SUF-CLMS-CPA-I. Under SUF-CLMS-CPA, if there is an adversary A_I who can win *Game* 3 in polynomial running time τ with a non-negligible probability advantage ε (A_I can ask for the same queries as A_I in ***Theorem* 3**), the challenger B can solve CDHP by interacting with the adversary A_I in time $\tau' \leq \tau + (2q_c + 2q_a)\, O(\tau_s)$ with a non-negligible probability advantage $\varepsilon' \geq (\varepsilon - q_a / 2^k) / 2$, where τ_s is the time of an elliptic curve scalar point multiplication operation.

Proof: Assume that an adversary A_I can attack the SUF-CLMS-CPA security with a non-negligible probability advantage ε and ask the challenger B for a series of queries under the random oracle model. Given a set of elements $< P, aP, bP>$, the challenger B computes abP to solve CDHP by interacting with the adversary A_I within a time bounded polynomial. The interaction between the challenger B and the adversary A_I is shown as follows:

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

Attack: A_I asks B for the same adaptive queries as *Phase* 1 in ***Theorem* 3**.

Forgery: A_I forges a new ciphertext $C^* =< J, W, z, h, a_0, a_1, \ldots, a_{n-1} >$ with a group of target identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, ID_n\}$ and a plaintext m. If equations $h = h'$ and $hzP = H_1(\mathrm{ID}_S, \mathrm{PK}_S)(\mathrm{PK}_S + P_{\mathrm{pub}}) + W$ hold, the ciphertext C^* is forged successfully. And setting $\mathrm{PK}_i' = \mathrm{b}^{-1}\mathrm{PK}_i$ and $F_i = b(\mathrm{PK}_i' + P_{\mathrm{pub}})$, B computes $F_i = \mathrm{PK}_i + abP$, and outputs $abP = F_i$-PK_i as the solution to CDHP. Otherwise, B outputs "failure".

Through the discussion above, it is concluded that during q_a signcryption queries, its successful probability advantage is at least $\varepsilon_a = \varepsilon - {}^{q_a}/_{2^k}$. And during the forger process, the correct probability that B computes abP is at least $\varepsilon_g = {}^1/_2$. Therefore, the probability advantage that B can solve CDHP by interacting with the adversary A_I is $\varepsilon' \geq \varepsilon_a \varepsilon_g = (\varepsilon - q_a / 2^k) / 2$ within running time $\tau' \leq \tau + (2q_c + 2q_a)\, O(\tau_s)$, where τ_s is the time of an elliptic curve scalar point multiplication operation. ■

Theorem 6: SUF-CLMS-CPA-II. Under SUF-CLMS-CPA, if there is an adversary A_{II} who can win *Game* 4 in polynomial running time τ with a non-negligible probability advantage ε (A_{II} can ask for the same queries as A_{II} in ***Theorem* 4**), the challenger B can solve CDHP by interacting with the adversary A_{II} in time $\tau' \leq \tau + (3q_c + 2q_a)\, O(\tau_s)$ with a non-negligible probability advantage $\varepsilon' \geq (\varepsilon - q_a / 2^k) / 2$, where τ_s is the time of an elliptic curve scalar point multiplication operation.

Proof: Assume that an adversary A_{II} can attack the SUF-CLMS-CPA security with a non-negligible probability advantage ε and ask the challenger B for a series of queries under the random oracle model. Given a set of elements $< P, aP, bP>$, the challenger B computes abP to solve CDHP by interacting with the adversary A_{II} within a time bounded polynomial. The interaction between the challenger B and the adversary A_{II} is shown as follows:

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

Attack: A_{II} asks B for the same adaptive queries as *Phase* 1 in ***Theorem* 4**.

Forgery: A_{II} forges a new ciphertext $C^* =< J, W, z, h, a_0, a_1, \ldots, a_{n-1} >$ with a set of target identities $L = \{\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_n\}$ and a plaintext m. If equations $h = h'$ and $hzP = H_1(\mathrm{ID}_S, \mathrm{PK}_S)(\mathrm{PK}_S + P_{\mathrm{pub}}) + W$ hold, the ciphertext C^* is forged successfully. And setting $\mathrm{PK}_i' = \mathrm{b}^{-1}\mathrm{PK}_i$ and $F_i = b(\mathrm{PK}_i' + K)$, B computes $F_i = \mathrm{PK}_i + abP$, and outputs

$abP = F_i$-PK_i as the solution to CDHP. Otherwise, B outputs "failure".

Through the discussion above, it is concluded that during q_a signcryption queries, its successful probability advantage is at least $\varepsilon_a = \varepsilon - {}^{q_a}/_{2^k}$. And for the forger process, the correct probability that B computes abP is at least $\varepsilon_g = {}^1/_2$. Therefore, the probability advantage that B can solve CDHP by interacting with the adversary A_{II} is $\varepsilon' \geq \varepsilon_a\varepsilon_g = \left(\varepsilon - q_a/2^k\right)/2$ within running time $\tau' \leq \tau + (3q_c + 2q_a)\, O(\tau_s)$, where τ_s is the time of an elliptic curve scalar point multiplication operation. ■

Theorem 7: ANON-CLMS-CCA-I. Under ANON-CLMS-CCA, if there is an adversary A_I who can win *Game* 5 in polynomial running time τ with a non-negligible probability advantage ε (A_I can ask for the same queries as A_I in ***Theorem* 3**), the challenger B can solve CDHP by interacting with the adversary A_I in time $\tau' \leq \tau + (2q_c + 3q_d)\, O(\tau_s)$ with a non-negligible probability advantage $\varepsilon' \geq \left(\varepsilon - q_d q_4/2^k\right)/nq_2$, where τ_s is the time of an elliptic curve scalar point multiplication operation.

Proof: Assume that an adversary A_I can attack the ANON-CLMS-CCA security with a non-negligible advantage ε and ask the challenger B for a series of queries under the random oracle model. Given a set of elements $< P$, aP, $bP>$, B computes abP to solve CDHP by interacting with the adversary A_I within a time bounded polynomial. The interaction between the challenger B and the adversary A_I is shown as follows:

Setup: B runs this algorithm to generate the master key $s = a \in Z_p^*$ and the system's public parameter $Params =< F_p, E, G_p, P, P_{\mathrm{pub}} = aP, E_k, D_k, p, H_0, H_1, H_2, H_3, H_4 >$ and then sends *Params* to A_I while keeping s secret. Upon receiving *Params*, A_I outputs a group of target identities $L = \{\mathrm{ID}_0, \mathrm{ID}_1\}$. It should be noted that H_0, H_1, H_2, H_3 and H_4 are random oracles controlled by B, and the random oracles interactions between A_I and B are the same as *Setup* in ***Theorem* 3**.

Phase 1: A_I asks B for the same adaptive queries as *Phase* 1 in ***Theorem* 3**.

Challenge: A_I chooses a plaintext m and a group of target identities $L^* = \{\mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, and sends them to B. Upon receiving m and L^*, B randomly chooses a bit $e \in \{0, 1\}$ and generates the ciphertext C^* with a group of new target identities $L^{**} = \{\mathrm{ID}_e, \mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$as follows:

a) Set $W_i = b\mathrm{PK}_i$, $F_i = b(\mathrm{PK}_i + P_{\mathrm{pub}})$ and $\alpha_i = H_2(F_i, W_i)$, where $i = e, 2, 3, \ldots, n$;

b) Randomly choose an integer $\xi \in Z_p^*$ and construct the polynomial

$$\begin{aligned} f(x) &= \prod_{j=1}^{n}\left(x - \alpha_j\right) + \xi (\bmod p) \\ &= a_0 + a_1 x \ldots + a_{n-1}x^{n-1} + x^n, \quad a_j \in Z_p^* \end{aligned}$$

c) Compute $k = H_3(\xi)$, $J^* = E_k(m||\mathrm{ID}_S)$ and $h^* = H_4(m||\mathrm{ID}_S, \xi, a_0, a_1, \ldots, a_{n-1}, W_i)$;

d) Randomly choose an integer $z \in Z_p^*$;

e) Return the ciphertext $C^* =< J^*$, W_i, z, h, a_0, $a_1, \ldots, a_{n-1} >$ to A_I.

Phase 2: A_I asks B for the same queries as *Phase* 2, but it should be noted that A_I cannot perform De-Signcryption query on C^*.

Guess: A_I guesses a bit e^*. If $e^* = e$ holds, A_I wins the game, and B outputs $abP = W_i$-F_i as the solution to CDHP. Otherwise, B outputs "failure".

Through the discussion above, it is concluded that during de-signcryption queries, H_4 hash could provide a valid ciphertext, so the probability that a valid ciphertext is rejected is not greater than ${}^{q_4}/_{2^k}$. Since A_I asks B for q_d de-signcryption queries during the attack process, the probability advantage that B decrypts the ciphertext successfully is $\varepsilon_d \geq \varepsilon - {}^{q_4 q_d}/_{2^k}$. And during the guess process, H_2 hash satisfies CDHP, so the correct probability that B computes abP is at least $\varepsilon_g = {}^1/_{nq_2}$. Therefore, the probability advantage that B can solve CDHP by interacting with the adversary A_I is $\varepsilon' \geq \varepsilon_d\varepsilon_g \geq \left(\varepsilon - q_d q_4/2^k\right)/nq_2$ within running time $\tau' \leq \tau + (2q_c + 3q_d)\, O(\tau_s)$, where τ_s is the time of an elliptic curve scalar point multiplication operation. ■

Theorem 8: ANON-CLMS-CCA-II. Under ANON-CLMS-CCA, if there is an adversary A_{II} who can win *Game* 6 in polynomial running time τ with a non-negligible advantage ε (A_{II} can ask for the same queries as A_{II} in ***Theorem* 4**), the challenger B can solve CDHP by interacting with the adversary A_{II} in time $\tau' \leq \tau + (3q_c + 3q_d)\, O(\tau_s)$ with a non-negligible probability advantage $\varepsilon' \geq \left(\varepsilon - q_d q_4/2^k\right)/nq_2$, where τ_s is the time of an elliptic curve scalar point multiplication operation.

Proof: Assume that an adversary A_{II} can attack the ANON-CLMS-CCA security with a non-negligible advantage ε and ask the challenger B for a series of queries under the random oracle model. Given a set of elements $< P$, aP, $bP>$, B computes abP to solve CDHP by interacting with the adversary A_{II} within a time bounded polynomial. The interaction between the challenger B and the adversary A_{II} is shown as follows:

Setup: B runs this algorithm to generate the master key $s \in Z_p^*$ and the system's public parameter $Params =< F_p$, $E, G_p, P, K = aP, P_{\mathrm{pub}}, E_k, D_k, p, H_0, H_1, H_2, H_3, H_4 >$, and then sends *Params* and s to A_{II}, where $a \in Z_p^*$. Upon receiving *Params* and s, A_{II} outputs a group of target identities $L = \{\mathrm{ID}_0, \mathrm{ID}_1\}$. It should be noted that H_0, H_1, H_2, H_3 and H_4 are random oracles controlled by B, and the random oracles interactions between A_{II} and B are the same as *Setup* in ***Theorem* 3**.

Phase 1: A_{II} asks B for the same adaptive queries as *Phase* 1 in ***Theorem* 4**.

Challenge: A_{II} chooses a plaintext m and a group of target identities $L^* = \{\mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$, and sends them to B. Upon receiving m and L^*, B randomly chooses a bit $e \in \{0,1\}$ and generates the ciphertext C^* with a group of new target identities $L^{**} = \{\mathrm{ID}_e, \mathrm{ID}_2, \mathrm{ID}_3, \ldots, \mathrm{ID}_n\}$as follows:

TABLE 1. Symbols' definition.

Symbols	Symbols' definition
T_m	It refers to the calculation time of modular multiplication operation.
T_i	It refers to the calculation time of modular inversion operation, $T_i \approx 11.6T_m$.
T_b	It refers to the calculation time of bilinear pairing operation, $T_b \approx 87T_m$.
T_e	It refers to the calculation time of modular exponentiation operation, $T_e \approx 240T_m$.
T_{bx}	It refers to the calculation time of bilinear pairing exponentiation operation, $T_{bx} \approx 43.5T_m$.
T_{pm}	It refers to the calculation time of scalar point multiplications on ECC operation, $T_{pm} \approx 29T_m$.
T_h	It refers to the calculation time of MTP hash function operation, $T_h \approx 29T_m$.
T_{pa}	It refers to the calculation time of point addition on ECC operation, $T_{pa} \approx 0.12T_m$.

TABLE 2. Comparison of efficiency.

Schemes	Signcryption/Encryption	De-signcryption/Decryption
Selvi *et al.*[25]	$(n+1)T_e+(n+1)T_{pm}\approx(269n+69)T_m$	$T_{pm}+T_{bx}+2T_b+T_{pa}\approx 246.62T_m$
Selvi *et al.*[26]	$(n+2)T_{pm}+T_i+2nT_{bx}+2nT_b\approx(290n+69.6)T_m$	$3T_b+T_{bx}+T_{pm}+T_{pa}\approx 333.62T_m$
Islam *et al.*[28]	$(2n+1)T_{pm}+2nT_{pa}\approx(58.24n+29)T_m$	$T_{pm}\approx 29T_m$
Hung *et al.*[29]	$(n+1)T_{pm}+nT_{bx}+nT_b+nT_h\approx(188.5n+29)T_m$	$T_{pm}+T_b\approx 116T_m$
He *et al.*[30]	$(3n+1)T_{pm}+nT_{pa}\approx(87.12n+29)T_m$	$2T_{pm}\approx 58T_m$
Zhu *et al.*[31]	$(2n+3)T_{pm}+nT_{pa}+T_b\approx(58.12n+174)T_m$	$2T_b+T_{pm}+T_i\approx 214.6T_m$
Win *et al.*[32]	$(n+2)T_{pm}+nT_{pa}\approx(29.12n+58)T_m$	$4T_{pm}+4T_{pa}+2T_i\approx 139.68T_m$
Ma *et al.*[33]	$(n+2)T_{pm}+nT_{bx}+nT_b+nT_h\approx(145n+58.24)T_m$	$T_{pm}+2T_h+T_b\approx 203T_m$
Our scheme	$(n+1)T_{pm}+nT_{pa}\approx(29.12n+29)T_m$	$2T_{pm}+T_{pa}\approx 58.12T_m$

n indicates the number of receivers.

a) Set $W_i = b(\mathrm{PK}_i + Y)$, $F_i = b(\mathrm{PK}_i + P_{\mathrm{pub}})$ and $\alpha_i = H_2(F_i, W_i)$, where $Y = K + P_{\mathrm{pub}}$ and $i = e, 2, 3, \ldots, n$;

b) Randomly choose an integer $\xi \in Z_p^*$ and construct the polynomial

$$f(x) = \prod_{j=1}^{n} (x - \alpha_j) + \xi (\mathrm{mod}\ p)$$
$$= a_0 + a_1 x + \ldots + a_{n-1}x^{n-1} + x^n, \quad a_j \in Z_p^*$$

c) Compute $k = H_3(\xi)$, $J^* = E_k(m||\mathrm{ID}_S)$ and $h^* = H_4(m||\mathrm{ID}_S, \xi, a_0, a_1, \ldots, a_{n-1}, W_i)$;

d) Randomly choose an integer $z \in Z_p^*$;

e) Return the ciphertext $C^* =< J^*, W_i, z, h, a_0, a_1, \ldots, a_{n-1} >$ to A_{II}.

Phase 2: A_{II} asks B for the same queries as *Phase* 2, but it should be noted that A_{II} cannot perform De-Signcryption query on C^*.

Guess: A_{II} guesses a bit e^*. If $e^* = e$ holds, A_{II} wins the game, and B outputs $abP = F_i$-W_i as the solution to CDHP. Otherwise, B outputs "failure".

Through the discussion above, it is concluded that during de-signcryption queries, H_4 hash could provide a valid ciphertext, so the probability that a valid ciphertext is rejected is not greater than $q_4/2^k$. Since A_{II} asks B for q_d de-signcryption queries during the attack process, the probability advantage that B decrypts the ciphertext successfully is $\varepsilon_d \geq \varepsilon - q_4 q_d/2^k$. And during the guess process, H_2 hash satisfies CDHP, so the correct probability that B computes abP is at least $\varepsilon_g = 1/nq_2$. Therefore, the probability advantage that B can solve CDHP by interacting with the adversary A_{II} is $\varepsilon' \geq \varepsilon_d \varepsilon_g \geq (\varepsilon - q_d q_4 / 2^k)/nq_2$ within running time $\tau' \leq \tau + (3q_c + 3q_d)\, O(\tau_s)$, where τ_s is the time of an elliptic curve scalar point multiplication operation. ■

V. EFFICIENCY ANALYSIS AND FUNCTIONAL COMPARISON

In order to evaluate our scheme, we will make comparisons between our scheme and the existing ones [25], [26], [28]–[33] in terms of computational efficiency and functions, because these schemes [25], [26], [28]–[33] are based on certificateless cryptography and they are similar to our scheme in some functions.

A. EFFICIENCY ANALYSIS

For ease of analysis, we define some symbols in TABLE 1, and the corresponding data are from [28]. It is worth noting that we only consider these operations' time defined in TABLE 1, and other operations' time is not considered

TABLE 3. Comparison of functions.

Schemes	Decryption fairness	Receiver anonymity	Partial private key verifiability	Signature
Selvi *et al.*[25]	No	No	No	Yes
Selvi *et al.*[26]	No	No	No	Yes
Islam *et al.*[28]	Yes	No	Yes	No
Hung *et al.*[29]	No	Yes	No	No
He *et al.*[30]	No	Yes	No	No
Zhu *et al.*[31]	No	No	No	No
Win *et al.*[32]	No	No	Yes	No
Ma *et al.*[33]	No	Yes	No	No
Our scheme	Yes	Yes	Yes	Yes

because their runtime can be negligible compared with that of operations defined in TABLE 1.

The comparisons of computational efficiency between our scheme and these schemes [25], [26], [28]–[33] in signcryption/encryption and de-signcryption/decryption are shown in TABLE 2.

From TABLE 2, we can see that compared with schemes [25], [26], [28]–[33], our scheme is the highest in computational efficiency in terms of signcryption/encryption process. In de-signcryption/decryption, our scheme is more efficient than schemes [25], [26], [29], [31]–[33], but more inefficient than schemes [28], [30]. The reason is that our scheme has the step to verify the message source, but these schemes [28], [30] do not.

B. FUNCTIONAL COMPARISON

The comparison of functions between our scheme and these schemes [25], [26], [28]–[33] is shown in the following TABLE 3.

From TABLE 3, we can see that only the scheme [28] and our scheme meet decryption fairness, which ensures that all authorized receivers have the same ability to decrypt the received ciphertext, while schemes [25], [26], [29]–[33] do not. In addition, in order to protect the receivers' privacy, schemes [29], [30], [33] and our scheme achieve the receiver anonymity, which means that no one except the sender knows the authorized receivers' identities. However, schemes [25], [26], [28], [31]–[32] do not take the receiver anonymity into account, which reveals the receivers' identities in their ciphertext directly. Schemes [28], [32] and our scheme possess the partial private key verifiability, which prevents the malicious KGC from producing fake partial private key to deceive users. Nevertheless, because the partial private key verifiability is unavailable in schemes [25], [26], [29]–[31], [33], they have no ability to prevent the malicious KGC's attack. Schemes [25], [26] and our scheme realize the signature function to ensure message's reliability, which avoids the situation that the attacker impersonates the sender's identity to send the message. But schemes [28]–[33] do not consider the function, and it is possible for the attacker to personate the sender's identity to do something bad. In short, compared with schemes [25], [26], [28]–[33], our scheme has more functions, and is more secure and more suitable for practical applications.

VI. CONCLUSION

In this paper, we propose an efficient anonymous certificateless multi-receiver signcryption scheme without bilinear pairings. Compared with existing CLME/CLMS schemes, the proposed scheme not only is high in computational efficiency, because the bilinear pairing and MTP hash function are not used and the number of scalar point multiplications on ECC is limited as small as possible, but also has more security functions such as decryption fairness, partial private key verifiability and signature. It has been proved to be secure in message confidentiality, unforgeability and receiver anonymity under the random oracle model. Therefore, whether in efficiency, functions, or in security, the proposed scheme is more in line with practical needs in application.

REFERENCES

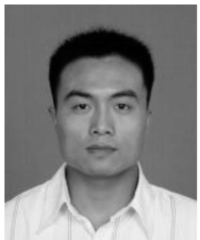
[1] J. Baek, R. Safavi-Naini, and W. Susilo, "Efficient multi-receiver identity-based encryption and its application to broadcast encryption," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 3386. Berlin, Germany: Springer, 2005, pp. 380–397.

[2] Y. L. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) ≪ cost(signature) + cost(encryption)," in *Advances in Cryptology—CRYPTO*, vol. 1294. Berlin, Germany: Springer, 1997, pp. 165–179.

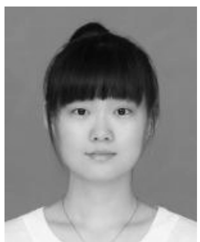
[3] S. Duan and Z. Cao, "Efficient and provably secure multi-receiver identity-based signcryption," in *Information Security and Privacy* (Lecture Notes in Computer Science), vol. 4058. Berlin, Germany: Springer, 2006, pp. 195–206.

[4] Y. Yu, B. Yang, X. Huang, and M. Zhang, "Efficient identity-based signcryption scheme for multiple receivers," in *Autonomic and Trusted Computing* (Lecture Notes in Computer Science), vol. 4610. Berlin, Germany: Springer, 2007, pp. 13–21.

[5] S. S. D. Selvi, S. S. Vivek, R. Srinivasan, and C. P. Rangan, "An efficient identity-based signcryption scheme for multiple receivers," in *Advances in Information and Computer Security* (Lecture Notes in Computer Science), vol. 5824. Berlin, Germany: Springer, 2009, pp. 71–88.
[6] S. Khullar, V. Richhariya, and V. Richhariya, "An efficient identity based multi-receiver signcryption scheme using ECC," *Int. J. Advancements Res. Technol.*, vol. 2, no. 4, pp. 189–193, Apr. 2013.
[7] 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.
[8] 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.
[9] Y.-M. Tseng, T.-T. Tsai, S.-S. Huang, and H.-Y. Chien, "Efficient anonymous multi-receiver ID-based encryption with constant decryption cost," in *Proc. Int. Conf. Inf. Sci., Electron. Elect. Eng.*, Sapporo, Japan, Apr. 2014, pp. 131–137.
[10] 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, Sep. 2010.
[11] 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.
[12] H.-Y. Chien, "Improved anonymous multi-receiver identity-based encryption," *Comput. J.*, vol. 55, no. 4, pp. 439–446, Apr. 2012.
[13] L. J. Pang and H. X. Li, "nMIBAS: A novel multi-receiver ID-based anonymous signcryption with decryption fairness," *Comput. Inf.*, vol. 32, no. 3, pp. 441–460, Mar. 2013.
[14] Y.-M. Tseng, Y.-H. Huang, and H.-J. Chang, "Privacy-preserving multireceiver ID-based encryption with provable security," *Int. J. Commun. Syst.*, vol. 27, no. 7, pp. 1034–1050, Jul. 2014.
[15] J. H. Zhang and J. Mao, "An improved anonymous multi-receiver identity-based encryption scheme," *Int. J. Commun. Syst.*, vol. 28, no. 4, pp. 645–658, Mar. 2015.
[16] L. J. 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.
[17] R. Skowronski, "ADON—Anonymous data exchange in hybrid opportunistic networks using utility functions and bloom filters," in *Proc. Int. Conf. Inf. Netw.*, Da Nang, Vietnam, Jan. 2017, pp. 558–563.
[18] Y. Sun, H. Zhu, and X. Feng, "A novel and concise multi-receiver protocol based on chaotic maps with privacy protection," *Int. J. Netw. Secur.*, vol. 19, no. 3, pp. 371–382, May 2017.
[19] K. He, J. Weng, Y. J. Mao, and H. Yuan, "Anonymous identity-based broadcast encryption technology for smart city information system," *Pers. Ubiquitous Comput.*, vol. 21, no. 5, pp. 841–853, Oct. 2017.
[20] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 2894. Berlin, Germany: Springer, 2003, pp. 452–473.
[21] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from asiacrypt 2003," in *Cryptology and Network Security* (Lecture Notes in Computer Science), vol. 3810. Berlin, Germany: Springer, 2005, pp. 13–25.
[22] Z. Zhang, D. S. Wong, J. Xu, and D. Feng, "Certificateless public-key signature: Security model and efficient construction," in *Applied Cryptography and Network Security* (Lecture Notes in Computer Science), vol. 3989. Berlin, Germany: Springer, 2006, pp. 293–308.
[23] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signature revisited," in *Information Security and Privacy* (Lecture Notes in Computer Science), vol. 4586. Berlin, Germany: Springer, 2007, pp. 308–322.
[24] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proc. ACM Symp. Inf., Comput. Commun. Secur.*, Tokyo, Japan, 2008, pp. 369–372.
[25] 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.
[26] S. S. D. Selvi, S. S. Vivek, and C. P. Rangan, "A note on the certificateless multi-receiver signcryption scheme," IACR Cryptol. ePrint Arch., Las Vegas, NV, USA, Tech. Rep., 2009, vol. 2009, p. 308. [Online]. Available: http://eprint.iacr.org/2009/308.pdf
[27] S. Miao, F. Zhang, and L. Zhang, "Cryptanalysis of a certificateless multi-receiver signcryption scheme," in *Proc. Int. Conf. MINES*, Nanjing, China, Nov. 2010, pp. 593–597.
[28] S. K. 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, Nov. 2015.
[29] 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.
[30] 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.
[31] 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.
[32] E. K. Win, T. Yoshihisa, Y. Ishi, T. Kawakami, Y. Teranishi, and S. Shimojo, "A lightweight multi-receiver encryption scheme with mutual authentication," in *Proc. IEEE 41st Annu. COMPSAC*, Turin, Italy, Jul. 2017, pp. 491–497.
[33] M. Ma, D. He, M. K. Khan, and J. Chen, "Certificateless searchable public key encryption scheme for mobile healthcare system," *Comput. Elect. Eng.*, vol. 65, pp. 413–424, Jan. 2018.
[34] M. Ma, D. He, N. Kumar, K.-K. R. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 759–767, Feb. 2018.



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 the Xidian University of China, in 2000, 2003, and 2006, respectively. He is currently a Full Professor with State Key Laboratory of Integrated Services Networks, School of Life Science and Technology, Xidian University, and also a Visiting Scholar with the Department of Computer Science, Wayne State University, USA. His research interests include Internet security, cryptography, secure mobile agent system, 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 interests include 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 Telecommunications Engineering, 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, and 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.