Anonymous Multi-receiver Certificateless Hybrid Signcryption (AMCLHS) for Broadcast Communication

Alia Umrani*, Apurva K Vangujar and Paolo Palmieri

School of Computer Science & IT,
University College Cork, Ireland
a.umrani@cs.ucc.ie, a.vangujar@cs.ucc.ie,
p.palmieri@cs.ucc.ie

Abstract. Confidentiality, authentication, and anonymity are the basic security requirements in broadcast communication, that can be achieved by Digital Signature (DS), encryption, and pseudo-identity PID techniques. Signcryption offers both DS and encryption more efficiently than "sign-then-encrypt,". However, compared to hybrid signcryption, it has higher computational and communication costs. Our paper proposes an Anonymous Multi-receiver Certificateless Hybrid Signcryption (AMCLHS) for secure broadcast communication. AMCLHS combines public-key cryptography and symmetric key to achieve confidentiality, authentication, and anonymity. We provide a simple and efficient construction of a multi-recipient Key Encapsulation Mechanism (mKEM) to create a symmetric session key. This symmetric session key, along with the sender's private key, is used in Data Encapsulation Mechanism DEM to signcrypt the message, thus providing confidentiality and authentication. It also generates identical ciphertext for multiple recipients while keeping their identities private by assigning a PID to each user. Our scheme demonstrate notions for Indistinguishability under Chosen-Ciphertext Attack using Elliptic Curve Computational Diffie-Hellman assumption in random oracle. It also demonstrates security for Existential Unforgeability against Chosen Message Attack using Elliptic Curve Diffie-Hellman assumption.The AMCLHS scheme operates in a multireceiver certificateless environment, preventing the key escrow problem. We show that, compared to existing schemes, our scheme is computationally efficient, provides optimal communication cost, and simultaneously ensures security properties such as confidentiality, authentication, anonymity, non-repudiation, and forward security.

Keywords: mKEM-DEM · Hybrid Signcryption · Certificateless · Multireceiver · Pseudo-Identity · Confidentiality · Authentication · Anonymity.

1 Introduction

Confidentiality, authentication, and anonymity are the basic security requirements in a broadcast communication [19,7]. The current solution to provide for these security requirements are encryption and Digital Signature (DS). However, the traditional

* Alia Umrani and Apurva K Vangujar are supported by PhD scholarships funded by the Science Foundation Ireland Centre under Grant No. SFI 18/CRT/6222

"sign-then-encrypt" approach results in high computational costs. Signcryption, on the other hand, allows both the encryption and signature operations to be performed simultaneously, providing both the confidentiality and authentication more efficiently. Signcryption was first proposed by Zhang et al. [30], as a novel cryptographic primitive. Since then, several signcryption schemes have been introduced in different application scenarios including healthcare, smart cards, and mobile ad-hoc communication [19,20,25,27,3].

Malone-Lee [14] proposed the first Identity (ID)-based signcryption scheme that provides both forward security and public verifiability. However, in ID-based schemes, the Public Key Generator generates the user's private key, leading to the issue of private key escrow. To solve the key escrow problem, Al-Riyami et al. [1] proposed a Certificateless Public Key Cryptography (CLPKC) that does not require the use of certificates and does not have a key escrow problem. In CLPKC, the Key Generation Center (KGC) generates a partial private key of the user by taking user's ID as input. The user then combines partial private key and a secret value to generate the actual private and public key pair. More specifically, the key escrow problem is prevented as the KGC does not have knowledge of the complete private key of the user. Following that, Barbosa and Farshim [2] proposed the first certificateless signcryption scheme that provides confidentiality and unforgeability and is secure under the Random Oracle Model (ROM).

The signcryption methods mentioned above are designed for single receiver scenarios, which are not suitable for broadcast communication with multiple receivers. When sending the same message to multiple recipients, the user has to encrypt a message for each individual recipient, causing an increase in computation time and communication lag. To address this, Yu et al. [28] proposed an ID-based multireceiver signcryption scheme that can encrypt a message for n designated recipients. The security of this scheme has been proven in a ROM. Later on, several ID-based signcryption schemes were proposed however, since ID-based PKC has an inherent key escrow problem, Selvi et al. [22] proposed the first multireceiver certificateless signcryption scheme and proven secure in ROM. Generally, the construction of signcryption can be achieved through two methods: (i) Public key signcryption: With public key signcryption, both message encryption and signing take place in a public key setting [22]. (ii) Hybrid signcryption: Hybrid signcryption provides the advantages of combining symmetric key encryption with asymmetric key signature while ensuring integrity, authentication, and simultaneously non-repudiation [21]. Hybrid signcryption is generally efficient in resource constrained environments than pure asymmetric signcryption because, in asymmetric signcryption alone, large messages are sent with the large public key values. For more reading, we refer to Dent's work [6,5] on Hybrid signcryption schemes.

In this paper, we propose an anonymous certificateless hybrid signcryption based on $(\mathsf{mKEM} - \mathsf{DEM})$ for broadcast communication. For confidentiality, we prove Indistinguishability under Chosen-Ciphertext Attack *(IND-CCA2-I)* for Type-I adversary, and *(IND-CCA2-II)* for Type-II adversary, based on Elliptic Curve based Computational Diffie Hellman (ECCDH) hard assumption. For unforgeability, we prove Existential Unforgeability against Chosen Message Attack *(EUF-CMA-I)* for Type-I adversary, and *(EUF-CMA-II)* for Type-II adversary, respectively, based on Elliptic Curve Discrete Logarithm (ECDL) hard assumption. Additionally, to ensure anonymity, each

user is assigned a Pseudo-Identity (PID). We additionally demonstrate the security for non-repudiation, and forward security. Finally, we compare our scheme with existing multireceiver certificateless hybrid signcryption schemes, demonstrating its efficiency in terms of computation cost, communication cost, and security requirements. In comparison to existing schemes listed at the end of the paper, our scheme demonstrates higher efficiency, with the signcryption cost increasing linearly with the number of designated receivers, while the unsigncryption cost remains constant. Our scheme simultaneously satisfy all the security requirements in terms of confidentiality, unforgeability, anonymity, non-repudiation, and forward security.

1.1 Our Contributions

The objective of this paper is to provide an anonymous certificateless hybrid signcryption scheme by utilizing $\mathsf{mKEM} - \mathsf{DEM}$. Our main contributions are as follows:

1. We propose a multi-recipient Key Encapsulation Mechanism (mKEM) - Data Encapsulation Mechanism (DEM) based Anonymous Multireceiver Certificateless Hybrid Signcryption (AMCLHS). The AMCLHS uses a combination of symmetric key and public-key cryptography to signcrypt an arbitrary-length message in broadcast communication.
2. The AMCLHS scheme achieves anonymity for each receiver by assigning a PID to each user (sender and receiver) and enables the sender to signcrypt an identical message for multiple receivers while keeping their identities anonymous.
3. The scheme operates in a multireceiver certificateless environment, preventing the key escrow problem. We achieve confidentiality by demonstrating security against *IND-CCA2* Type-I and Type-II adversaries and unforgeability by demonstrating security against *EUF-CMA* Type-I and Type-II adversaries, respectively. The security is demonstrated using ECCDH and ECDL hard assumptions under the ROM.
4. We compare the proposed scheme with existing multireceiver hybrid signcryption schemes in terms of computation cost, communication cost, and security requirements. We show that, compared to existing multireceiver schemes, our scheme is computationally more efficient and has optimal communication costs, with signcryption cost is linear to the number of designated receivers, while the unsigncryption cost remains constant. Our scheme simultaneously achieves non-repudiation, and forward security.

The remainder of the paper is organized as follows: The related work is provided in Section 2. Section 3 describes the preliminaries and mathematical assumptions. In Section 4, we introduce the AMCLHS framework and security model of the scheme in Section 5. Section 6 introduces the proposed AMCLHS scheme and in Section 7, we provide the security analysis under the hardness assumption. Section 8 provide the performance analysis and comparison of the proposed scheme. Lastly, in Section 9, we conclude the work.

2 Related Work

2.1 Certificateless Signcryption

Signcryption was first introduced by Zheng et al. [30] in 1997 combining the signature and encryption to provide authentication and confidentiality more efficiently than sign-then-encrypt. Several ID-based signcryption schemes have been proposed, however, the key issue with ID-based signcryption is the presence of a key escrow problem. To address this, Barbosa and Farshim [2] proposed the first certificateless signcryption scheme that provides both confidentiality and authentication and is secure under the ROM. Chen et al. [3] and Cui et al. [4] proposed a certificateless signcryption scheme for the Internet of Medical Things without pairings and the Internet of Vehicles, respectively. The schemes provides confidentiality and authentication and proves security for *IND-CCA* and *EUF-CMA* under ECDL and Computational Diffie-Hellman (CDH) assumptions. Similarly, a certificateless signcryption scheme without ROM was proposed by ZHOU et al. [31] that achieves confidentiality and unforgeability however, does not provide anonymity. Kasyoka et al. [11] proposed a certificateless signcryption for wireless sensor networks. Additionally, Cui et al. [4] presented a pairing-free certificateless signcryption scheme for the Internet of Vehicles. Li et al. [13] proposed a signcryption scheme for resource-constrained smart terminals in cyber-physical power systems. However, all the aforementioned schemes are designed for single receivers, which are not suitable for broadcast communication. For example, to send an identical message to multiple receivers, the sender must encrypt a message for each recipient, resulting in poor performance.

Yu et al. [28] introduced the first multireceiver signcryption scheme based on ID-based PKC, enabling message encryption for n designated receivers. The security of the scheme is based on CDH assumption under the ROM. Later on, several multireceiver certificateless signcryption schemes were proposed. In 2022, Niu et al. [16] proposed a privacy-preserving mutual heterogeneous signcryption scheme based on 5G network slicing, where the sender is in a PKI environment, and the receiver is in a certificateless environment. The proposed scheme is secure against *IND-CCA2* and *EUF-CMA* under the hardness assumptions of CDH and Discrete Logarithm (DL). In addition, numerous multireceiver certificateless signcryption schemes have been introduced in edge computing, smart mobile IoT, and IoT-enabled maritime transportation systems [19,20,25,27]. The above schemes based on large and resource-constrained environment are proven secure in public key settings, however, they may become computationally expensive when dealing with large messages, compared to hybrid settings. On the other hand, hybrid signcryption is generally more efficient than public key signcryption alone because it uses the combination of symmetric key and PKC. A message is encrypted using a symmetric key algorithm, which is faster and more efficient [6,5].

2.2 Certificateless Hybrid Signcryption

Dent et al. [5,6] proposed the first hybrid signcryption scheme with insider and outsider security. Following that, Li et al. [12] proposed the first certificateless hybrid signcryption scheme. Wu et al. [24] proposed a certificateless hybrid signcryption scheme

for IoT. The scheme utilizes PKC to generate a symmetric key and the symmetric key is used to signcrypt the message. While the scheme provides confidentiality, authentication, forward security, and public verification under CDH and DBDH assumptions, it incurs high computational cost due to BP operation. and Yin et al. [26] proposed a certificateless hybrid signcryption scheme for wireless sensor networks. Similarly, Gong et al. [8] presented a lightweight and secure certificateless hybrid signcryption scheme for the Internet of Things (IoT). It ensures data confidentiality, integrity, and authenticity. The scheme utilizes bilinear pairings for initialization and key construction and proves security under CDH and DBDH assumptions. Hongzhen et al. [9] presented a pairing-free certificateless signcryption scheme for Vehicular Ad hoc Networks. Moreover, Zhang et al. [29] introduced a certificateless hybrid signcryption scheme suitable for the IoT. The scheme is constructed to achieve both confidentiality and unforgeability under the hardness assumptions of DL, CDH, DBDH, and BDH.

In 2017, Niu et al. [15] proposed a heterogeneous hybrid signcryption for multi-message and multi-receiver. The scheme proves security against *IND-CCA* and *EUF-CMA* attacks under the ROM based on the hardness assumptions of DBDH and variants of DBDH and CBDH. In 2022, Niu et al. [17] presented a broadcast signcryption scheme based on certificateless cryptography for wireless sensor networks. The scheme aims to ensure the confidentiality and integrity of the data transmitted, while protecting by the privacy of the receiver's identity under ECDH and ECDL assumptions. The scheme uses a trusted third party to outsource the encryption operation and assumes that the trusted third party is always available. However, it may not be realistic in some scenarios, for instance, if the trusted third party is offline, the scheme may not work properly. Moreover, the scheme incurs higher computational costs compared to the AMCLHS scheme (see Table 1).

3 Preliminaries and Assumptions

3.1 Elliptic Curve based Computational Diffie-Hellman (ECCDH) Assumption

The security assumption of ECCDH is according to [18].

Definition 1. *The ECCDH assumption holds given $(P, xP, yP) \in \mathbb{G}$, where $x, y \in \mathbb{Z}_q^*$, it is computationally infeasible for any Probabilistic Polynomial-Time (PPT) algorithm to compute xyP.*

3.2 Elliptic Curve Discrete Logarithm (ECDL) Assumption

The security Definition of ECDL is according to [10].

Definition 2. *Given P and $Q \in \mathbb{G}$, it is hard to find an $x \in \mathbb{Z}_q^*$ for any PPT algorithm with non-negligible probability such that $Q = x \cdot P$.*

3.3 The multi-recipient Key Encapsulation Mechanism (mKEM) and Data Encapsulation Mechanism (DEM)

The notion of mKEM was first proposed by N.P Smart [23] and has a KEM like construction which takes multiple receiver's public keys $\mathsf{pk}_{\mathsf{r}_i}$ where $1 \leq \mathsf{i} \leq t$ and $t < n$ as input and generates a single symmetric session key K and an encapsulation C of K.

Definition 3. *The* mKEM *construction below is according to [23]:*

1. mKEM*: It consists of four algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{mKEM.Encaps}, \mathsf{mKEM.Decaps})$ *defined as follows:*
   - Setup*: On input the security parameter* 1^λ*, the algorithm outputs* PP.
   - KeyGen*: Taking* PP *as input, the algorithm outputs* $(\mathsf{pk}, \mathsf{sk})$ *for each user.*
   - mKEM.Encaps*: On input* PP *and a set of receiver public keys* $\mathsf{pk}_{\mathsf{r}_i}$ *where* $1 \leq \mathsf{i} \leq t$*, this algorithm outputs a symmetric session key* K *and an encapsulation* C_1 *of* K *where* K *is used in* DEM.
   - mKEM.Decaps*: Taking* PP*, receiver's private key* $\mathsf{sk}_{\mathsf{r}_i}$ *corresponding to* $\mathsf{pk}_{\mathsf{r}_i}$*, and an encapsulation* C_1 *as input, this algorithm outputs* K*. The correctness of* mKEM *holds if* $\mathsf{K} = \mathsf{mKEM.Decaps}(\mathsf{PP}, \mathsf{sk}_{\mathsf{r}_i}, \mathsf{C}_1)$.
2. DEM*: It consists of two algorithms* $(\mathsf{Enc}_\mathsf{K}, \mathsf{Dec}_\mathsf{K})$ *[15] defined as follows:*
   - Enc_K*: On input* K *and* m*, this algorithm outputs a ciphertext* C_2.
   - Dec_K*: Taking* K *and* C_2 *as input, this algorithm outputs* m'*. The correctness of* DEM *holds if* $m' = m$.

3.4 KEM-DEM Hybrid Signcryption Scheme

Definition 4. *The construction of* KEM-DEM *hybrid signcryption scheme is given by [5]. It consists of four algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Gen} - \mathsf{Enc}, \mathsf{Dec} - \mathsf{Ver})$ *defined as follows:*

1. Setup*: It takes as input a security parameter* 1^λ *and outputs* PP.
2. KeyGen*: Taking* PP *as input, this algorithm outputs a public/private key pair for sender* $(\mathsf{pk}_\mathsf{s}, \mathsf{sk}_\mathsf{s})$ *and receiver* $(\mathsf{pk}_\mathsf{r}, \mathsf{sk}_\mathsf{r})$.
3. Gen − Enc*: In generation-encryption, the sender runs following algorithms:*
   - Encaps*: Taking as input* PP*, sender's private key* sk_s*, receiver's public keys* pk_r*, and a message* m*, it outputs a symmetric session key* K *and an encapsulation* C_1.
   - Enc_K*: It takes* K *as input and outputs* C_2*. The receiver outputs* $(\mathsf{C}_1, \mathsf{C}_2)$.
4. Dec − Ver*: In decryption-verification, the receiver runs following algorithms:*
   - Decaps*: Taking as input receiver's private key* sk_r *and* C_1*, it outputs* K*. If* $\mathsf{K} = \perp$*, the sender stops. Otherwise, the receiver runs next algorithmic step.*
   - Dec_K*: It takes* C_2 *and* K *as input and retrieves* m*. If* $m = \perp$*, the receivers stops. Otherwise, the receiver runs next* Ver *step.*
   - Ver*: Taking sender's public key* pk_s*,* m*, and* C_1 *as input, it outputs either valid or not. If valid, outputs* m*, else outputs* $\perp$.

4 AMCLHS Framework

4.1 Framework

The framework of the AMCLHS scheme consists of four entities: KGC, a Registration Authority (RA), and n users such as $n = \{\mathsf{PID_s}, \{\mathsf{PID}_1, ..., \mathsf{PID}_{r_i}, ..., \mathsf{PID}_{r_t}\}\}$. Assume, a sender with $\mathsf{PID_s}$ sends an arbitrary length message m to t designated receivers with PID_{r_i} where $1 \leq i \leq t$. The role of each entity is defined below:

- **KGC**: The KGC is responsible for generating public parameters (PP), master secret key (msk) of KGC, master public key (mpk) of KGC, and partial private key (ppk) for each user taking part in communication.
- **RA**: The RA is a semi-trusted authority that first generates its private key $\mathsf{sk_{RA}}$ and public key $\mathsf{pk_{RA}}$. RA is also responsible for user registration, identity verification, and PID generation.
- **Sender**: The sender with identity $\mathsf{PID_s}$ encrypts a m using the set of designated receiver's public key pk_{r_i}, signs with its private key $\mathsf{sk_s}$ and sends the signcrypted ciphertext CT to t designated receivers.
- **Receiver**: The designated receiver with PID_{r_i} and sk_{r_i}, decrypt the CT, and verify the signature using sender's public key $\mathsf{pk_s}$.

4.2 Definition of AMCLHS

The AMCLHS scheme represents a hybrid approach, leveraging both mKEM and DEM components. Before signcrypting the message, each user is assigned a pseudo-identity PID by RA, taking user's real identity $\mathsf{ID_R}$ as input. For signcryption, this framework firstly utilizes mKEM that takes a set of receiver's public keys as input, and generates a symmetric session key K and an encapsulation C_1 of that key. The mKEM also takes a sender's private key to generate the signature S which is encapsulated in C_1 and verified in the unsigncryption phase as given in Definition 5. Following this, the DEM and session key K are jointly used to symmetrically encrypt m, producing a ciphertext C_2. This ciphertext is then represented as a signcrypted ciphertext pair $\mathsf{CT} = (\mathsf{C}_1, \mathsf{C}_2)$. For decryption, the process starts with the decapsulation of C_1 using mKEM and the receiver's private key to retrieve K. After this, the message m is decrypted from C_2 using K. Once the m is decrypted, the receiver verifies the signature S using Ver algorithm by taking sender's public key and C_1 as input. Hence, the AMCLHS scheme introduces an effective and secure mechanism for data signcryption and unsigncryption, employing both symmetric and asymmetric key strategies in a unique hybrid methodology.

Definition 5. *In the AMCLHS scheme, the sender with* $\mathsf{PID_s}$ *sends an arbitrary length* m *to* t *designated receivers denoted with* PID_{r_i} *where* $1 \leq i \leq t$. *The AMCLHS scheme follows the Definitions 3 and 4. The proposed scheme consists of eight polynomial time algorithms.*

1. **Setup**: On input the security parameter 1^λ, the KGC generates PP, msk, and mpk. Next, RA generates $\mathsf{sk_{RA}}$ and $\mathsf{pk_{RA}}$.

2. **Pseudo Identity**: Taking the Real-Identity $\mathsf{ID_R}$ of each user and $\mathsf{pk_{RA}}$ as input and generates a PID as output.
3. **Partial Private Key**: For each PID, the KGC takes mpk and msk as input, and generates the partial private key (ppk) for each user.
4. **Secret Value**: On input the PID, each user generates a secret value (sv).
5. **Private Key**: Taking ppk and sv as input, each user generates the sk.
6. **Public Key**: On input the sv, each user generates the pk.
7. **Signcryption**: To signcrypt the message m and generate the CT, the sender runs this algorithm in two phases. In Phase 1, the sender runs $\mathsf{mKEM.Encaps}$ and in Phase 2, the sender runs $\mathsf{Enc_K}$ according to the Definition 3. The phases are defined as follows:
   - Phase 1 ($\mathsf{mKEM.Encaps}$): Taking PP, $\mathsf{sk_s}$, a plaintext m and a set $\mathsf{pk_{r_i}}$ for $1 \leq \mathsf{i} \leq t$, this algorithm outputs $\mathsf{C_1}$ and K.
   - Phase 2 ($\mathsf{Enc_K}$): Taking K and m as input, this algorithm outputs $\mathsf{C_2}$ and sets signcrypted ciphertext $\mathsf{CT} = (\mathsf{C_1}, \mathsf{C_2})$.
8. **Unsigncryption**: To unsigncrypt the CT and generate m, the receiver runs this algorithm in three phases. Phase 1 consists of $\mathsf{mKEM.Decaps}$ and Phase 2 consists of $\mathsf{Dec_K}$ according to the Definition 3.
   - Phase 1 ($\mathsf{mKEM.Decaps}$): Taking $\mathsf{sk_{r_i}}$ and $\mathsf{C_1}$ as input, this algorithm outputs K.
   - Phase 2 ($\mathsf{Dec_K}$): Taking K and $\mathsf{C_2}$ as input, this algorithm outputs m'. If $m' \neq m$, the receiver rejects the message. If $m' = m$, the receiver verifies the signature in Phase 3.
   - Phase 3 (Ver): Taking m', $\mathsf{C_1}$, and $\mathsf{pk_s}$ as input, this algorithm verifies the signature S. If it is valid, accepts the m, else returns $\perp$ and aborts.

5 Security Model

We define the notions of *IND-CCA2* and *EUF-CMA* as our security definitions to ensure confidentiality and unforgeability, respectively. We precisely define the security Game-I for *IND-CCA2-I* and *IND-CCA2-II* in Section 5.1, to evaluate the security against Type-I adversary ($\mathcal{A}_{\mathrm{I}}$) and Type-II adversary ($\mathcal{A}_{\mathrm{II}}$), respectively. Moreover, in Section 5.2, we introduce the security Game-II for *EUF-CMA-I* and *EUF-CMA-II* to evaluate the security against $\mathcal{A}_{\mathrm{I}}$ and $\mathcal{A}_{\mathrm{II}}$. The $\mathcal{A}_{\mathrm{I}}$ and $\mathcal{A}_{\mathrm{II}}$ are defined as follows:

1. $\mathcal{A}_{\mathrm{I}}$: $\mathcal{A}_{\mathrm{I}}$ is an honest-but-curious malicious user who cannot access msk but can replace the pk of any ID with the value of his/her own choice. $\mathcal{A}_{\mathrm{I}}$ is not allowed to ask a ppk query q_{ppk} for any of the target identities.
2. $\mathcal{A}_{\mathrm{II}}$: $\mathcal{A}_{\mathrm{II}}$, also known as malicious KGC, cannot make public key replace query q_{pr} for the target ID. $\mathcal{A}_{\mathrm{II}}$ is not allowed to make sv extract queries q_{sv}. If the q_{pr} has been done for the target ID, then the q_{sv} is not allowed for the same ID.

5.1 Game-I

The Game-I is interaction between the Challenger $\mathcal{C}$ and $\mathcal{A}$ in three phases. In each phase, the $\mathcal{A}$ asks a polynomially bounded number of hash and public and private key

queries. Finally, $\mathcal{A}$ provides a target plaintext pair (m_0, m_1) to $\mathcal{C}$. $\mathcal{C}$ picks $\beta \in \{0,1\}^*$ randomly and responds with a challenge CT^*. $\mathcal{A}$ returns $\beta' \in \{0,1\}^*$ and wins the Game-I if $\beta = \beta'$. The details of the security model are provided in Definition 6.

Definition 6. *The IND-CCA2 requires that there exists no PPT Adversary $\mathcal{A}$ which could distinguish ciphertexts. Therefore, the security game that captures confidentiality is based on the ciphertext indistinguishability. The advantage of $\mathcal{A}$ is defined as the probability that $\mathcal{A}$ wins the game.*

1. ***Phase-1***: The $\mathcal{A}$ asks polynomially bounded number of hash queries q_{H_l} where $\{l = 1, 2, 3\}$. The $\mathcal{C}$ keeps a list L_l of q_{H_l} to record the responses.
   - **Setup**: The $\mathcal{C}$ generates $(\mathsf{PP}, \mathsf{msk}, \mathsf{mpk}, \mathsf{sk_{RA}}, \mathsf{pk_{RA}})$ and passes to $\mathcal{A}$. Then $\mathcal{A}$ selects t target $\mathsf{PID_{r_i}}$ where $1 \leq \mathsf{i} \leq t$.

2. ***Phase-2***: The $\mathcal{A}$ proceeds to make a series of queries, subject to the restrictions defined in section 5. The queries include public key retrieve query q_{pk}, partial private key query q_{ppk}, secret value extract query q_{sv}, public key replace query q_{pr}, signcryption query q_{sc}, and unsigncryption query q_{usc}. An initially empty list L_{pk} is maintained by the $\mathcal{C}$ to store the public key and secret value information. The $\mathcal{C}$ responds to each query as follows:
   - q_{pk}: Upon receiving the first such query for PID, the $\mathcal{C}$ searches L_{pk} for pk. If it does not exists, $\mathcal{C}$ runs the secret value algorithm to generate a sv for PID, and then performs the public key algorithm to return the pk to $\mathcal{A}$.
   - q_{ppk}: Given PID as input, the $\mathcal{C}$ checks if $\mathsf{PID} = \mathsf{PID}^*$. If it does, the $\mathcal{C}$ aborts. Otherwise, it fetches the ppk form L_{pk}. If it does not exists in L_{pk} then $\mathcal{C}$ runs partial private key algorithm to return ppk and updates L_{pk}.
   - q_{sv}: Upon receiving q_{sv} for PID, the $\mathcal{C}$ checks L_{pk} for sv. If it does not exists, $\mathcal{C}$ runs q_{pk} and returns sv to $\mathcal{A}$.
   - q_{pr}: Given PID as input, the $\mathcal{C}$ replaces pk with pk' and updates L_{pk}.
   - q_{sc}: On input the message m, $\mathsf{PID_s}$, and $\mathsf{PID_{r_i}}$, the $\mathcal{C}$ checks if $\mathsf{PID_{r_i}} = \mathsf{PID}^*$. If it is not, $\mathcal{C}$ performs normal signcryption operation by taking values from L_{pk}. Otherwise, $\mathcal{C}$ performs the signcryption algorithm to generate CT.
   - q_{usc}: Upon receiving $(\mathsf{CT}, \mathsf{PID_s}, \mathsf{PID_{r_i}})$ as input, the $\mathcal{C}$ checks if $\mathsf{PID_{r_i}} = \mathsf{PID}^*$. If it is not, $\mathcal{C}$ performs normal unsigncryption operation. Otherwise, $\mathcal{C}$ performs the unsigncryption algorithm to answer m.

3. ***Challenge***: The $\mathcal{A}$ outputs a target plaintext pair (m_0, m_1). The $\mathcal{C}$ picks $\beta \in \{0,1\}^*$ at random, sets challenge CT^*, and sends CT^* to $\mathcal{A}$.

4. ***Phase-3***: The $\mathcal{A}$ can make further queries except that the target CT^* is not allowed to appear in the q_{usc}.

5. ***Guess***: Finally, $\mathcal{A}$ responds with its guess $\beta' \in \{0,1\}^*$. If $\beta = \beta'$, $\mathcal{A}$ wins the Game-I. The advantage of $\mathcal{A}_{\mathrm{I}}$ is defined as:

$$Adv_{\mathcal{A}_{\mathrm{I}}}^{IND-CCA2} = \mid \Pr[\beta = \beta'] - 1/2 \mid \tag{1}$$

   The advantage of $\mathcal{A}_{\mathrm{II}}$ is defined as:

$$Adv_{\mathcal{A}_{\mathrm{II}}}^{IND-CCA2} = \mid \Pr[\beta = \beta'] - 1/2 \mid \tag{2}$$

5.2 Game-II

Game-II is the interaction between the Challenger $\mathcal{C}$ and $\mathcal{A}$ in two phases. In each phase, the $\mathcal{A}$ asks a polynomially bounded number of hash and public and private key queries. In the end, $\mathcal{A}$ outputs the forged ciphertext. $\mathcal{A}$ wins if unsigncryption does not return $\perp$. The security is given in the Definition 7 below in detail.

Definition 7. *For EUF-CMA, we define Game-II played between $\mathcal{C}$ and $\mathcal{A}$. An AM-CLHS is Type-I and Type-II EUF-CMA secure if every PPT $\mathcal{A}$ has a negligible advantage in winning the Game-II.*

1. ***Phase-1***: The $\mathcal{A}$ asks polynomially bounded number of hash queries q_{H_l} $\{l = 1, 2, 3\}$. The $\mathcal{C}$ keeps a list L_l of q_{H_l} to record the responses.
   - **Setup**: The $\mathcal{C}$ generates $(\mathsf{PP}, \mathsf{msk}, \mathsf{mpk}, \mathsf{sk}_{\mathsf{RA}}, \mathsf{pk}_{\mathsf{RA}})$ and sends PP to $\mathcal{A}$. $\mathcal{A}$ selects a target $\mathsf{PID}_{\mathsf{s}}^{*}$.

2. ***Phase-2***: The $\mathcal{A}$ first asks number of queries with the restrictions defined in Section 5. The queries include q_{pk}, q_{ppk}, q_{pr}, q_{sv}, q_{sc}, and q_{usc} and are defined in *Phase-2* of Game-I in Definition. 5.1. $\mathcal{C}$ maintains an initially empty list L_{pk} to store the pk and sv information.

3. ***Forgery***: $\mathcal{A}$ outputs the forged ciphertext under a targeted $\mathsf{PID}_{\mathsf{s}}^{*}$. $\mathcal{A}$ wins if unsigncryption does not return $\perp$.

6 Anonymous Multireceiver Certificateless Hybrid Signcryption Scheme (AMCLHS)

In this section, we turn our focus towards the design and construction of the proposed AMCLHS scheme, built upon the integral $\mathsf{mKEM\text{-}DEM}$ framework, in alignment with the guidelines laid out in Definition 5. The core structure of the scheme is shown in Fig. 1.

1. **Setup**: The KGC begins by initializing the system, taking the security parameter λ as input. It chooses a subgroup $\mathbb{G}$ of large prime order $q > 2^{\lambda}$, derived from an elliptic curve E over a finite field $\mathbb{F}_q$. The KGC selects a generator point $P \in \mathbb{G}$ and generates four hash functions. The first hash function is $H_0 : \{0,1\}^{\ell} \rightarrow \mathbb{G}$, where ℓ is a positive integer. The second hash function is $H_1 : \{0,1\}^{\ell} \times \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}$. The third hash function is $H_2 : \mathbb{Z}_q^* \times \mathbb{Z}_q^* \times \mathbb{G} \times \mathbb{G} \rightarrow \{0,1\}^k$, where k denotes the plaintext box length. The fourth hash function is $H_3 : \{0,1\}^* \times \{0,1\}^k \times \{0,1\}^* \times \{0,1\}^k \times \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$.
   Next, the KGC generates the $\mathsf{PP} = \{\mathbb{G}, E, P, q, H_0, H_1, H_2, H_3\}$, randomly selects $x_0 \in \mathbb{Z}_q^*$ as the master secret key msk, and calculates the master public key $\mathsf{mpk} = x_0 \cdot P$. It then publish PP as public and mpk, while keeping msk secret. Subsequently, the RA selects $v \in \mathbb{Z}_q^*$ at random as its secret key $\mathsf{sk}_{\mathsf{RA}}$ and calculates its public key $\mathsf{pk}_{\mathsf{RA}} = v \cdot P$. The RA publicizes $\mathsf{pk}_{\mathsf{RA}}$ and keeps $\mathsf{sk}_{\mathsf{RA}}$ as a secret.
2. **Pseudo-Identity**: This algorithm is run by the each user and RA as follows:

- User: Each user chooses random $\mathsf{ID_R} \in \{0,1\}^\ell$ and computes $R = \alpha \cdot P$ where $\alpha \in \mathbb{Z}_q^*$. Taking $\mathsf{ID_R}$ and α as input, it computes initial $\mathsf{PID_1} = \mathsf{ID_R} \oplus H_0(\alpha.\mathsf{pk_{RA}})$ and sends $(\mathsf{PID_1}, R)$ to RA.
- RA: Taking $(\mathsf{PID_1}, R)$ as input, the RA verifies the $\mathsf{ID_R}$ as $\mathsf{ID_R} = \mathsf{PID_1} \oplus H_0(R.v)$. If it holds, the RA accepts the registration request from users and sends $\mathsf{PID} = \mathsf{ID_R} \oplus H_0(\alpha \cdot \mathsf{pk_{RA}})$.

3. **Partial Private Key**: Taking PID, mpk, and msk as input, the KGC computes $Q_{\mathsf{PID}} = H_1(\mathsf{PID}\|\mathsf{mpk})$ and the ppk as $\mathsf{d} = x_0 \cdot Q_{\mathsf{PID}}$.
4. **Secret Value**: Each user with PID chooses $x \in \mathbb{Z}_q^*$ randomly as a sv.
5. **Private Key**: On input (d, x), each user with PID sets $\mathsf{sk} = (\mathsf{d}, x)$.
6. **Public Key**: Taking x as input, each user with PID computes $\mathsf{pk} = x \cdot P$.
7. **Signcryption**: The sender with $\mathsf{PID_s}$ and $\mathsf{sk_s}$ runs following phases to signcrypt a message m and sends CT to receivers with $\mathsf{PID_{r_i}}$ and $\mathsf{pk_{r_i}}$ $1 \leq \mathsf{i} \leq t$:
   - Phase 1 (mKEM-Encaps):
     (a) Randomly chooses $r \in \mathbb{Z}_q^*$ and computes $U = r \cdot P$.
     (b) Taking $\mathsf{pk_{r_i}}$ and $Q_{\mathsf{PID_{r_i}}}$ as input, computes $Z_{1_\mathsf{i}} = \mathsf{d_s}.Q_{\mathsf{PID_{r_i}}}$ and $Z_{2_\mathsf{i}} = x_\mathsf{s}.\mathsf{pk_{r_i}}$.
     (c) Computes $\psi = (Z_{1_\mathsf{i}} \cdot Z_{2_\mathsf{i}})$ and $\mathsf{K} = H_2(\psi)$.
     (d) Computes $f = H_3(m, \psi, \mathsf{PID_s}, \mathsf{PID_{r_i}}, \mathsf{pk_s}, \mathsf{pk_{r_i}})$ and Signature $S_\mathsf{i} = r^{-1}(f + w.\mathsf{d_s}x_\mathsf{s})$ where $w = \mathrm{x}_U \mathrm{mod}(n)$ which is the x-coordinate of U.
     (e) Sets $\mathsf{C_1} = (f, S_\mathsf{i})$ and outputs $(\mathsf{C_1}, \mathsf{K})$.
   - Phase 2 ($\mathsf{Enc_K}$):
     (a) Computes $\mathsf{C_2} = \mathsf{Enc_K}(m)$. Sets $\mathsf{CT} = (\mathsf{C_1}, \mathsf{C_2})$ and sends to t designated receivers.
8. **Unsigncryption**: The designated receiver with $\mathsf{PID_{r_i}}$ takes $\mathsf{sk_{r_i}}$ and $\mathsf{pk_s}$ as input, and runs the following phases to unsigncrypt the CT and generate m:
   - Phase 1 (mKEM-Decaps):
     (a) Taking $x_{\mathsf{r_i}}$ and $\mathsf{d_{r_i}}$ as input, computes $Z_{1_\mathsf{i}} = \mathsf{d_{r_i}}.Q_{\mathsf{PID_s}}$ and $Z_{2_\mathsf{i}} = \mathsf{pk_s}.x_{\mathsf{r_i}}$.
     (b) Computes $\psi = (Z_{1_\mathsf{i}} \cdot Z_{2_\mathsf{i}})$ and $\mathsf{K} = H_2(\psi)$.
     If $\mathsf{K} = \perp$, the receiver aborts otherwise decrypts m as follows:
   - Phase 2 ($\mathsf{Dec_K}$):
     (a) $m' = \mathsf{Dec_K}(\mathsf{C_2})$. If $m' = m$ verifies the signature else rejects.
   - Phase 3 (Ver):
     (a) Taking $\mathsf{C_1}$ and $\mathsf{pk_s}$ as input, computes $f' = H_3(m', \psi, \mathsf{PID_s}, \mathsf{PID_{r_i}}, \mathsf{pk_s}, \mathsf{pk_{r_i}})$.
     (b) If $f' = f$, verifies S_i by checking if $U = r \cdot P$ and $w' = x_U \mathrm{mod} n$.
     If $w' = w$, the receiver will accept the signcrypted m else returns $\perp$ and aborts.

Correctness

1. $\mathsf{ID_R} = \mathsf{PID_1} \oplus H_0(R.v) = \mathsf{ID_R} \oplus H_0(\alpha.\mathsf{pk_{RA}}) \oplus H_0(Rv) = \mathsf{ID_R} \oplus H_0(R.v) \oplus H_0(Rv) = \mathsf{ID_R}$
2. $Z_{1_\mathsf{i}} = \mathsf{d_s}.Q_{\mathsf{PID_{r_i}}} = x_0.Q_{\mathsf{PID_s}}.Q_{\mathsf{PID_{r_i}}} = \mathsf{d_{r_i}}.Q_{\mathsf{PID_s}}$
3. $Z_{2_\mathsf{i}} = x_\mathsf{s}.\mathsf{pk_{r_i}} = x_\mathsf{s}.\mathsf{pk_{r_i}} = x_\mathsf{s}.x_{\mathsf{r_i}}.P = \mathsf{pk_s}.x_{\mathsf{r_i}} = \mathsf{pk_s}.x_{\mathsf{r_i}}$
4. Let $u_1 = f.P$ and $u_2 = w.\mathsf{pk_s}.Z_{1_\mathsf{i}}.Q_{\mathsf{PID_{r_i}}}^{-1}$
   $U_i = S_\mathsf{i}^{-1}(u_1 + u_2) = S_\mathsf{i}^{-1}(f.P + w.\mathsf{pk_s}.Z_{1_\mathsf{i}}.Q_{\mathsf{PID_{r_i}}}^{-1}) = S_\mathsf{i}^{-1}(f.P + w.\mathsf{pk_s}.\mathsf{d_s}.Q_{\mathsf{PID_{r_i}}} Q_{\mathsf{PID_{r_i}}}^{-1}) = S_\mathsf{i}^{-1}(f.P + w.x_\mathsf{s}.P.\mathsf{d_s}) = \frac{(f.P + w.x_\mathsf{s}.P.\mathsf{d_s})}{S_\mathsf{i}} = \frac{P(f + w.x_\mathsf{s}.\mathsf{d_s})}{r^{-1}(f + w.x_\mathsf{s}.\mathsf{d_s})} = \frac{P}{r^{-1}} = r.P.$

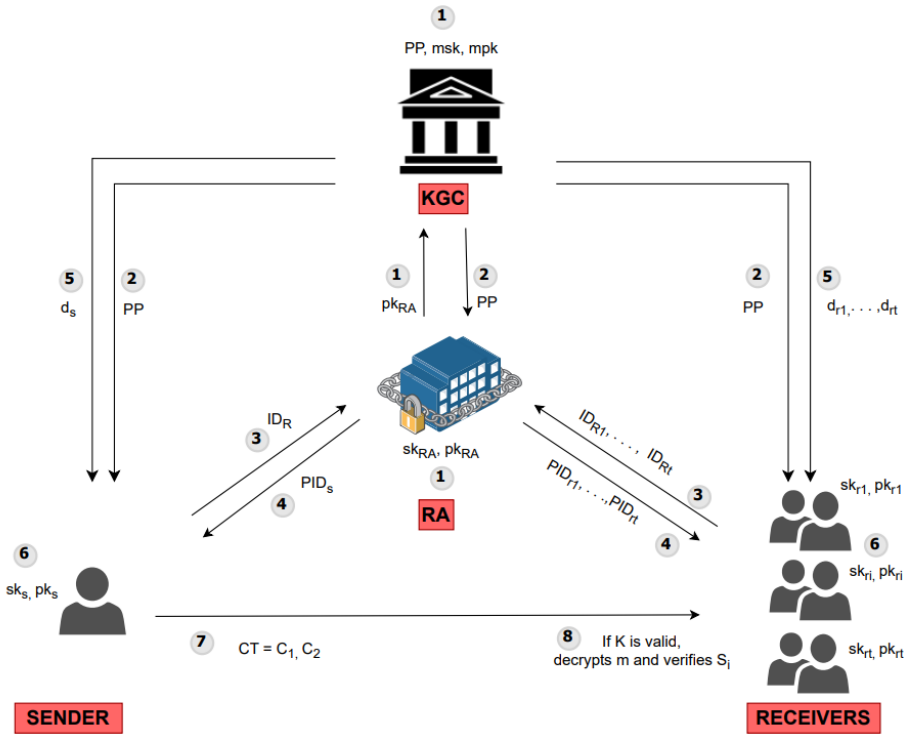

Fig. 1. The mKEM-DEM (AMCLHS) scheme

7 Security Analysis

The security analysis of the proposed AMCLHS scheme is based on the security model defined in Section 5. The message confidentiality is based on Theorems 1 and 2 which demonstrates that the scheme is secure against *IND-CCA2* Type-I and Type-II adversaries in aforementioned Game-I in Definition 5.1. Similarly, unforgeability is based on Theorems 3 and 4 and follows that the scheme is secure against *EUF-CMA* Type-I and Type-II adversaries in the aforementioned Game-II in Definition 5.2.

Confidentiality

Theorem 1. *The proposed scheme is IND-CCA2-I secure under the ROM based on the hardness of the ECCDH assumption. Suppose that the IND-CCA2-I adversary $\mathcal{A}_{\mathrm{I}}$ has a non-negligible advantage ϵ in winning the game then, there is $\mathcal{C}$ that can solve the ECCDH with the non-negligible advantage ϵ'.*

Proof. Given a random instance $(P, xP, yP) \in \mathbb{G}$ of the ECCDH, the $\mathcal{C}$ has to compute xyP as definition given in Section 1 by interacting with the $\mathcal{A}_{\mathrm{I}}$ as follows:

1. ***Phase-1***: A polynomially bounded number of queries q are made by an $\mathcal{A}_{\mathrm{I}}$. The $\mathcal{C}$ keeps a list L_l of q_{H_l} to record the responses.

- **Setup**: The $\mathcal{C}$ runs the setup algorithm to generate $\mathsf{PP} = \{\mathbb{G}, E, P, q, H_0, H_1, H_2, H_3\}$. The $\mathcal{C}$ sets new value for the $\mathsf{mpk} = \theta \cdot P$ and sends PP and mpk to the $\mathcal{A}_{\mathrm{I}}$. The $\mathcal{A}_{\mathrm{I}}$ selects t target identities denoted by $\mathsf{PID}_{\mathsf{i}}^*$ where $1 \leq \mathsf{i} \leq t$.
- H_1**-Query**: Upon receiving H_1 query from the $\mathcal{A}_{\mathrm{I}}$, $\mathcal{C}$ determines whether the tuple $(Q_{\mathsf{PID}_{\mathsf{i}}}, \mathsf{mpk}, \mathsf{PID}_{\mathsf{i}})$ exists in the list L_1 or not. If it already exists, $\mathcal{C}$ returns $Q_{\mathsf{PID}_{\mathsf{i}}}$ to $\mathcal{A}_{\mathrm{I}}$. Otherwise if $\mathsf{PID}_{\mathsf{i}} \neq \mathsf{PID}_{\mathsf{i}}^*$, $\mathcal{C}$ sets $Q_{\mathsf{PID}_{\mathsf{i}}} = H_1(\mathsf{PID}_{\mathsf{i}} \| \mathsf{mpk})$. If $\mathsf{PID}_{\mathsf{i}} = \mathsf{PID}_{\mathsf{i}}^*$, $\mathcal{C}$ chooses $\gamma \in \mathbb{Z}_q^*$ randomly and computes $Q_{\mathsf{PID}_{\mathsf{i}}} = \gamma \cdot P$ and adds a new tuple $(Q_{\mathsf{PID}_{\mathsf{i}}}, \mathsf{mpk}, \mathsf{PID}_{\mathsf{i}})$ in L_1 and sends $Q_{\mathsf{PID}_{\mathsf{i}}}$ to $\mathcal{A}_{\mathrm{I}}$.
- H_2**-Query**: Upon receiving H_2 query from the $\mathcal{A}_{\mathrm{I}}$, $\mathcal{C}$ determines whether the tuple $(\mathsf{K}, \psi, Z_{1_{\mathsf{i}}}, Z_{2_{\mathsf{i}}})$, exists in the list L_2 or not. If it already exists, $\mathcal{C}$ returns K to $\mathcal{A}_{\mathrm{I}}$. Otherwise, $\mathcal{C}$ chooses $\mathsf{K} \in \{0,1\}^k$ randomly, updates the tuple $(\mathsf{K}, \psi, Z_{1_{\mathsf{i}}}, Z_{2_{\mathsf{i}}})$, and sends K to $\mathcal{A}_{\mathrm{I}}$.
- H_3**-Query**: Upon receiving H_3 query from the $\mathcal{A}_{\mathrm{I}}$, $\mathcal{C}$ determines whether the tuple $H_3(m, \psi, f)$ exists in the list L_3 or not. If it already exists, $\mathcal{C}$ returns f to $\mathcal{A}_{\mathrm{I}}$. Otherwise, it chooses $f \in \mathbb{Z}_q^*$ randomly, updates the tuple $H_3(m, \psi, f)$ and sends f to $\mathcal{A}_{\mathrm{I}}$.

2. ***Phase-2***: The $\mathcal{A}_{\mathrm{I}}$ asks a number queries including q_{pk}, q_{ppk}, q_{pr}, q_{sv}, and q_{usc}. The $\mathcal{C}$ maintains an initially empty list L_{pk} to store the pk and sv information. The $\mathcal{C}$ responds to the queries as follows:
   - q_{pk}: Upon receiving the pk_{i} query for $\mathsf{PID}_{\mathsf{i}}$, $\mathcal{C}$ checks if pk_{i} exists in L_{pk} If it exists, $\mathcal{C}$ returns pk_{i} to $\mathcal{A}_{\mathrm{I}}$. Otherwise, $\mathcal{C}$ chooses $x_{\mathsf{i}} \in \mathbb{Z}_q^*$ and computes $\mathsf{pk}_{\mathsf{i}} = x_{\mathsf{i}}.P$ and adds the tuple $(\mathsf{PID}_{\mathsf{i}}, -, \mathsf{pk}_{\mathsf{i}}, x_{\mathsf{i}})$ in L_{pk} and returns pk_{i} to $\mathcal{A}_{\mathrm{I}}$.
   - q_{ppk}: Upon receiving the query, if $\mathsf{PID}_{\mathsf{i}} = \mathsf{PID}_{\mathsf{i}}^*$, the $\mathcal{C}$ aborts. Otherwise, if it exists in the list L_{pk}, $\mathcal{C}$ sends d_{i} to $\mathcal{A}_{\mathrm{I}}$, if it does not, $\mathcal{C}$ randomly chooses $Q_{\mathsf{PID}_{\mathsf{i}}} = \gamma \cdot P$ from L_1 and return $\mathsf{d}_{\mathsf{i}} = \mathsf{mpk} Q_{\mathsf{PID}_{\mathsf{i}}}$ to $\mathcal{A}_{\mathrm{I}}$. The $\mathcal{C}$ then updates the tuple $(\mathsf{PID}_{\mathsf{i}}, \mathsf{d}_{\mathsf{i}}, \mathsf{pk}_{\mathsf{i}}, x_{\mathsf{i}})$ in L_{pk}.
   - q_{sv}: Upon receiving the query, $\mathcal{C}$ checks if it exists in the list, L_{pk}, if it does, $\mathcal{C}$ sends x_{i} to $\mathcal{A}_{\mathrm{I}}$. If it does not, $\mathcal{C}$ performs the public key retrieve query and return x_{i} to $\mathcal{A}_{\mathrm{I}}$.
   - q_{pr}: Upon receiving the query, the $\mathcal{C}$ replaces the public key pk_{i} with $\mathsf{pk}'_{\mathsf{i}}$ for $\mathsf{PID}_{\mathsf{i}}$ and updates the tuple $(\mathsf{PID}_{\mathsf{i}}, \mathsf{d}_{\mathsf{i}}, \mathsf{pk}'_{\mathsf{i}}, -)$ in the list L_{pk}.
   - q_{sc}: Upon receiving the query with sender's $\mathsf{PID}_{\mathsf{s}}$, receiver's $\mathsf{PID}_{\mathsf{r_i}}$ and m, the $\mathcal{C}$ checks whether $\mathsf{PID}_{\mathsf{r_i}} = \mathsf{PID}_{\mathsf{i}}^*$. The $\mathcal{C}$ performs the normal signcryption operation if $\mathsf{PID}_{\mathsf{r_i}} \neq \mathsf{PID}_{\mathsf{i}}^*$ by taking values from L_{pk}. Otherwise, the $\mathcal{C}$ performs the signcryption as follows:
     - If pk_{i} is replaced, the $\mathcal{A}_{\mathrm{I}}$ will provide another value.
     - Chooses $r \in \mathbb{Z}_q^*$ randomly and computes $U = r \cdot P$.
     - Gets $Q_{\mathsf{PID}_{\mathsf{r_i}}}$ from L_1 and computes $Z_{1_{\mathsf{i}}} = \mathsf{d}_{\mathsf{s}}.Q_{\mathsf{PID}_{\mathsf{r_i}}}$, $Z_{2_{\mathsf{i}}} = x_{\mathsf{s}}.\mathsf{pk}_{\mathsf{r_i}}$, $\psi = (Z_{1_{\mathsf{i}}} \cdot Z_{2_{\mathsf{i}}})$, $\mathsf{K} = H_2(\psi)$, and updates L_2.
     - Computes $f = H_3(m, \psi, \mathsf{PID}_{\mathsf{s}}, \mathsf{PID}_{\mathsf{r_i}}, \mathsf{pk}_{\mathsf{s}}, \mathsf{pk}_{\mathsf{r_i}})$ and updates L_3.
     - Computes S_{i} and sets $\mathsf{C}_1 = (f, S_{\mathsf{i}})$, computes $\mathsf{C}_2 = \mathsf{Enc}_{\mathsf{K}}(m)$, and returns $\mathsf{CT} = (\mathsf{C}_1, \mathsf{C}_2)$ to adversary $\mathcal{A}_{\mathrm{I}}$.
   - q_{usc}: Upon receiving the query with sender's $\mathsf{PID}_{\mathsf{s}}$, receiver's $\mathsf{PID}_{\mathsf{r_i}}$ and a CT, the $\mathcal{C}$ checks whether $\mathsf{PID}_{\mathsf{r_i}} = \mathsf{PID}_{\mathsf{i}}^*$ or not. If $\mathsf{PID}_{\mathsf{r_i}} \neq \mathsf{PID}_{\mathsf{i}}^*$, the $\mathcal{C}$ performs the normal unsigncryption operation. Otherwise, the $\mathcal{C}$ unsigncrypts m as follows:

- If $\mathsf{pk_i}$ is replaced, the $\mathcal{A}_{\mathrm{I}}$ will provide another value.
- Searches the lists L_2 and L_3 for $(\mathsf{K}, \psi, Z_{1_i}, Z_{2_i})$ and $H_3(m, \psi, f)$.
- If the record does not exist, $\mathcal{C}$ returns "failure". If it exists, the $\mathcal{C}$ computes $\mathsf{K} \neq \perp$ and $m = \mathsf{Dec_K}(\mathsf{C}_2)$.
- Checks if $f' = f$, if it holds then checks if $U = r.P$ and $w' = \mathrm{x}_U \bmod(n)$ holds or not. If yes, the $\mathcal{C}$ answers m else, returns $\perp$.

3. ***Challenge***: The $\mathcal{A}_{\mathrm{I}}$ chooses equal length plaintext message pair (m_0, m_1) and sends the target plaintext to the $\mathcal{C}$. The $\mathcal{A}_{\mathrm{I}}$ takes a sender $\mathsf{PID_s}$ and a target $\mathsf{PID}_{\mathsf{r_i}}$. Moreover, the $\mathcal{A}_{\mathrm{I}}$ can not ask for the sk of the target $\mathsf{PID}_{\mathsf{r_i}}$. If $\mathsf{PID}_{\mathsf{r_i}} \neq \mathsf{PID}^*_{\mathsf{i}}$, the returns $\perp$. Otherwise, the $\mathcal{C}$ chooses $\beta \in \{0,1\}^*$ and performs the following steps to generate a challenge CT^*:
   - Chooses $r^* \in \mathbb{Z}_q^*$ and computes $U^* = r^* \cdot P$
   - Computes $Z^*_{1_i} = \mathsf{d_s}.Q_{\mathsf{PID}_{\mathsf{r_i}}}$, $Z^*_{2_i} = x_{\mathsf{s}}.\mathsf{pk}_{\mathsf{r_i}}$, and $\psi^* = (Z^*_{1_i} \cdot Z^*_{2_i})$. Computes $\mathsf{K}^* = H_2(\psi^*)$
   - $f^* = H_3(m, \psi^*, \mathsf{PID_s}, \mathsf{PID}_{\mathsf{r_i}}, \mathsf{pk_s}, \mathsf{pk}_{\mathsf{r_i}})$. Computes $S^*_{\mathsf{i}} = r^{*-1}(f^* + w.d_s x_{\mathsf{s}})$ and $\mathsf{C}^*_1 = (f^*, S^*_i)$.
   - $\mathsf{C}^*_2 = \mathsf{Enc}_{\mathsf{K}^*}(m)$ and computes $\mathsf{CT}^* = (\mathsf{C}^*_1, \mathsf{C}^*_2)$.

4. ***Phase-3***: The adversary $\mathcal{A}_{\mathcal{I}}$ may issue further polynomially bounded queries as in *Phase-1*, however, $\mathcal{A}_{\mathrm{I}}$ cannot send the q_{ppk} of the target $\mathsf{PID}_{\mathsf{r_i}}$, or the unsigncryption query for CT^*.

5. ***Guess***: The $\mathcal{A}_{\mathrm{I}}$ will respond with the guess bit $\beta \in \{0,1\}^*$. Adversary wins the game if $\beta' = \beta$. The $\mathcal{C}$ will win the game by evaluating $\frac{\theta.Z_{1_i} - \mathsf{d_i}.r}{(\mathsf{d_s} - U)} = \theta \cdot \gamma \cdot P$ using $\mathsf{mpk} = \theta \cdot P$, $Q_{\mathsf{PID_i}} = \gamma \cdot P$ which is the solution to the ECCDH.

In the end, the $\mathcal{C}$ is able to find the solution to the ECCDH $\theta \cdot \gamma \cdot P$. Next, we evaluate the advantage of $\mathcal{C}$ winning the Game-I *(IND-CCA-I)* by calculating the probability of aborting the game during occurrence of the following events:

1. In partial private key query, the game aborts for $\mathsf{PID_i} = \mathsf{PID}^*_{\mathsf{i}}$. The probability is $\Pr(E_{q_{\mathsf{ppk}}}) = 1/q_{\mathsf{ppk}}$.
2. In unsigncryption query, the game aborts due to invalid m. The probability is $\Pr(E_{q_{\mathsf{usc}}}) = q_{\mathsf{usc}}/2^k$.
3. In the challenge phase, $\mathcal{C}$ aborts the game if the adversary queries against the identity $\mathsf{PID}_{\mathsf{r_i}} \neq \mathsf{PID}^*_{\mathsf{i}}$. The probability is $\Pr(E_{q_{H_1}}) = (1 - 1/qH_1)$.

Moreover, the $\mathcal{C}$ fetches the list L_1 to retrieve $Q_{\mathsf{PID_i}}$ and L_2 to retrieve Z_{1_i} and evaluates $\frac{\theta.Z_{1_i} - \mathsf{d_i}.r}{(\mathsf{d_s} - U)} = \theta \cdot \gamma \cdot P$ with probability $(1/q_{H_1} + 1/q_{H_2})$. Therefore, the probability of the $\mathcal{C}$ winning the game with advantage ϵ' is:

$$\epsilon' \geq \epsilon \left(\frac{1}{qH_1} + \frac{1}{qH_2}\right)\left(\frac{1}{qH_1}\right)\left(1 - \frac{1}{q_{\mathsf{ppk}}}\right)\left(1 - \frac{q_{\mathsf{usc}}}{2^k}\right) \tag{3}$$

Theorem 2. *The proposed scheme is IND-CCA2-II secure under the ROM based on the hardness of the ECCDH assumption. Suppose that the IND-CCA2-II adversary $\mathcal{A}_{\mathrm{II}}$ has a non-negligible advantage ϵ in winning the game then, there is a $\mathcal{C}$ that can solve the ECCDH with the non-negligible advantage ϵ'.*

Proof. Given a random instance $(P, xP, yP) \in \mathbb{G}$ of the ECCDH, the $\mathcal{C}$ has to compute xyP as definition given in Section 1 by interacting with the $\mathcal{A}_{\text{II}}$ as follows:

1. ***Phase-1***: A polynomially bounded number of queries q are made by an $\mathcal{A}_{\text{II}}$. The Challenger $\mathcal{C}$ keeps a list L_l of q_{H_l} to record the responses.
   - **Setup**: The $\mathcal{C}$ runs the setup algorithm to generate $\mathsf{PP} = \{\mathbb{G}, P, q, H_0, H_1, H_2, H_3\}$. The $\mathcal{C}$ sets new $\mathsf{mpk} = \theta \cdot P$ and sends PP and mpk to the $\mathcal{A}_{\text{II}}$. The $\mathcal{A}_{\text{II}}$ selects the target $\mathsf{PID}_{\mathsf{i}}^*$ $1 \leq \mathsf{i} \leq t$.
   - H_1**-Query**: Upon receiving H_1 query from the $\mathcal{A}_{\text{II}}$, the $\mathcal{C}$ determines whether the tuple $(Q_{\mathsf{PID}_\mathsf{i}}, \mathsf{mpk}, \mathsf{PID}_\mathsf{i})$ exists in the list L_1 or not. If it already exists, $\mathcal{C}$ returns $Q_{\mathsf{PID}_\mathsf{i}}$ to $\mathcal{A}_{\text{II}}$. Otherwise, if $\mathsf{PID}_\mathsf{i} \neq \mathsf{PID}_\mathsf{i}^*$, $\mathcal{C}$ sets $Q_{\mathsf{PID}_\mathsf{i}} = H_1(\mathsf{PID}_\mathsf{i} \| \mathsf{mpk})$. If $\mathsf{PID}_\mathsf{i} = \mathsf{PID}_\mathsf{i}^*$, $\mathcal{C}$ chooses $\gamma \in \mathbb{Z}_q^*$ randomly and computes $Q_{\mathsf{PID}_\mathsf{i}} = \gamma \cdot P$ and adds a new tuple $(Q_{\mathsf{PID}_\mathsf{i}}, \mathsf{mpk}, \mathsf{PID}_\mathsf{i})$ in L_1. The $\mathcal{C}$ sends $Q_{\mathsf{PID}_\mathsf{i}}$ to $\mathcal{A}_{\text{II}}$.
   - H_2, H_3**-Query**: Upon receiving H_2 and H_3 queries from the $\mathcal{A}_{\text{II}}$, the $\mathcal{C}$ determines whether the tuple $(\mathsf{K}, \psi, Z_{1_\mathsf{i}}, Z_{2_\mathsf{i}})$, and $H_3(m, \psi, f)$ exists in the list L_2 and L_3 or not. If it already exists, $\mathcal{C}$ returns K and f to $\mathcal{A}_{\text{II}}$. Otherwise, the $\mathcal{C}$ chooses $\mathsf{K} \in \{0,1\}^k$ and $f \in \mathbb{Z}_q^*$ randomly and updates the tuple $(\mathsf{K}, \psi, Z_{1_\mathsf{i}}, Z_{2_\mathsf{i}})$, and $H_3(m, \psi, f)$. The $\mathcal{C}$ sends ψ and f to $\mathcal{A}_{\text{II}}$.

2. ***Phase-2***: The adversary $\mathcal{A}_{\text{II}}$ asks a number of queries in an manner, including q_{pk}, q_{sv}, and q_{usc}. The $\mathcal{C}$ maintains an initially empty list L_{pk} to store the public key and secret value information. The $\mathcal{C}$ responds to the queries as follows:
   - q_{pk}: Upon receiving the pk_i query for PID_i, the $\mathcal{C}$ checks if pk_i exists in the L_{pk} as $(\mathsf{PID}_\mathsf{i}, \mathsf{d}_\mathsf{i}, \mathsf{pk}_\mathsf{i}, x_\mathsf{i})$. If it exists, $\mathcal{C}$ returns pk_i to $\mathcal{C}$. Otherwise, $\mathcal{C}$ chooses $x_\mathsf{i} \in \mathbb{Z}_q^*$ and computes $\mathsf{pk}_\mathsf{i} = x_\mathsf{i} \cdot P$ and adds the tuple $(\mathsf{PID}_\mathsf{i}, -, \mathsf{pk}_\mathsf{i}, x_\mathsf{i})$ in L_{pk} and returns pk_i to $\mathcal{A}_{\text{II}}$.
   - q_{sv}: Upon receiving the query for PID_i, the $\mathcal{C}$ checks if $\mathsf{PID}_\mathsf{i} = \mathsf{PID}_\mathsf{i}^*$. If it holds, the $\mathcal{C}$ aborts because in this case, the PID_i is a target identity. Otherwise, it checks if x_i already exists in the L_{pk} as $(\mathsf{PID}_\mathsf{i}, \mathsf{d}_\mathsf{i}, \mathsf{pk}_\mathsf{i}, x_\mathsf{i})$. If it exists, the $\mathcal{C}$ returns x_i to $\mathcal{A}_{\text{II}}$. Otherwise, $\mathcal{C}$ runs q_{pk} and computes $\mathsf{pk}_\mathsf{i} = x_\mathsf{i} \cdot P$ and adds the tuple $(\mathsf{PID}_\mathsf{i}, \mathsf{d}_\mathsf{i}, \mathsf{pk}_\mathsf{i}, x_\mathsf{i})$ in L_{pk} and returns x_i to $\mathcal{A}_{\text{II}}$.
   - q_{sc}: Upon receiving the query with sender's PID_s, target $\mathsf{PID}_{\mathsf{r}_\mathsf{i}}$, and m, the $\mathcal{C}$ checks whether $\mathsf{PID}_{\mathsf{r}_\mathsf{i}} = \mathsf{PID}_\mathsf{i}^*$ or not. The $\mathcal{C}$ performs the normal signcryption operation if $\mathsf{PID}_{\mathsf{r}_\mathsf{i}} \neq \mathsf{PID}_\mathsf{i}^*$ by taking values from L_{pk}. Otherwise, if $\mathsf{PID}_{\mathsf{r}_\mathsf{i}} = \mathsf{PID}_\mathsf{i}^*$, the $\mathcal{C}$ performs the signcryption as follows:
     - Chooses $r \in \mathbb{Z}_q^*$ and computes $U = r \cdot P$.
     - Gets $Q_{\mathsf{PID}_{\mathsf{r}_\mathsf{i}}}$ from L_1 and computes $Z_{1_\mathsf{i}} = \mathsf{d}_\mathsf{s}.Q_{\mathsf{PID}_{\mathsf{r}_\mathsf{i}}}$, $Z_{2_\mathsf{i}} = x_\mathsf{s}.\mathsf{pk}_{\mathsf{r}_\mathsf{i}}$, and $\psi = (Z_{1_\mathsf{i}} \cdot Z_{2_\mathsf{i}})$ and $\mathsf{K} = H_2(\psi)$.
     - Computes $f = H_3(m, \psi, \mathsf{PID}_\mathsf{s}, \mathsf{PID}_{\mathsf{r}_\mathsf{i}}, \mathsf{pk}_\mathsf{s}, \mathsf{pk}_{\mathsf{r}_\mathsf{i}})$ and updates L_3.
     - Computes S_i and computes $\mathsf{C}_1 = (f, S_i)$, $\mathsf{C}_2 = \mathsf{Enc}_\mathsf{K}(m)$ and returns $\mathsf{CT} = (\mathsf{C}_1, \mathsf{C}_2)$ to adversary $\mathcal{A}_{\text{II}}$.
   - q_{usc}: Upon receiving the query with sender's PID_s, receiver's $\mathsf{PID}_{\mathsf{r}_\mathsf{i}}$, and a CT, the $\mathcal{C}$ checks whether $\mathsf{PID}_{\mathsf{r}_\mathsf{i}} = \mathsf{PID}_\mathsf{i}^*$ or not. The $\mathcal{C}$ performs the normal unsigncryption operation if $\mathsf{PID}_{\mathsf{r}_\mathsf{i}} \neq \mathsf{PID}_\mathsf{i}^*$. Otherwise, the $\mathcal{C}$ unsigncrypts m as follows:
     - The $\mathcal{C}$ searches the lists L_2 and L_3 for $(\mathsf{K}, \psi, Z_{1_\mathsf{i}}, Z_{2_\mathsf{i}})$, and (m, ψ, f).

- If the record does not exist, $\mathcal{C}$ returns "failure". If it exists, the $\mathcal{C}$ computes $\mathsf{K} \neq \perp$ and $m' = \mathsf{Dec}_{\mathsf{K}}(\psi)$.
- Checks if $f' = f$, if it holds then checks if $U = r \cdot P$ and $w' = \mathrm{x}_U \bmod(n)$ holds or not. If yes, the $\mathcal{C}$ answers m else, returns $\perp$.

3. ***Challenge***: The $\mathcal{A}_{\mathrm{II}}$ chooses target plaintext m_0, m_1 and sends the target plaintext to the $\mathcal{C}$. The $\mathcal{A}_{\mathrm{II}}$ takes a sender $\mathsf{PID_s}$ and a target $\mathsf{PID}_{\mathsf{r_i}}$. Moreover, the $\mathcal{A}_{\mathrm{II}}$ can not ask for the sk of the receiver $\mathsf{PID}_{\mathsf{r_i}}$. If $\mathsf{PID}_{\mathsf{r_i}} \neq \mathsf{PID}^*_{\mathsf{i}}$, the returns $\perp$. Otherwise, the $\mathcal{C}$ chooses $\beta \in \{0,1\}^*$ and performs the following steps to generate a challenge CT^*:
   - Chooses $r^* \in \mathbb{Z}_q^*$ and computes $U^* = r^* \cdot P$.
   - Computes $Z_{1_{\mathsf{i}}} = \mathsf{d_s}.Q_{\mathsf{PID}_{\mathsf{r_i}}}$, $Z_{2_{\mathsf{i}}} = x_{\mathsf{s}}.\mathsf{pk}_{\mathsf{r_i}}$, and $\psi^* = (Z^*_{1_{\mathsf{i}}} \cdot Z^*_{2_{\mathsf{i}}})$. Computes $\mathsf{K}^* = H_2(\psi^*)$.
   - $f^* = H_3(m, \psi^*, \mathsf{PID_s}, \mathsf{PID}_{\mathsf{r_i}}, \mathsf{pk_s}, \mathsf{pk}_{\mathsf{r_i}})$. Computes $S^*_{\mathsf{i}} = r^{*-1}(f^* + w.\mathsf{d_s}x_{\mathsf{s}})$ and $\mathsf{C}^*_1 = (f^*, S^*_i)$
   - $\mathsf{C}^*_2 = \mathsf{Enc}^*_{\mathsf{K}}(m)$ and $\mathsf{CT}^* = (\mathsf{C}^*_1, \mathsf{C}^*_2)$.

4. ***Phase-3***: The adversary $\mathcal{A}_{\mathrm{II}}$ may issue further polynomially bounded queries as in *Phase-1* however, $\mathcal{A}_{\mathrm{II}}$ cannot send the q_{sv} for the target $\mathsf{PID}^*_{\mathsf{r_i}}$ and the unsigncryption query for CT^*.

5. ***Guess***: The adversary $\mathcal{A}_{\mathrm{II}}$ will respond with the guess bit $\beta \in \{0,1\}^*$. Adversary wins the game if $\beta' = \beta$. The $\mathcal{C}$ will win the game by obtaining $\theta \cdot \gamma \cdot P$ which is the solution to the ECCDH assumption. The $\mathcal{C}$ obtains it by evaluating $\frac{\theta.Z_{1_i} - \mathsf{d_i}.r}{(\mathsf{d_s} - U)} = \theta \cdot \gamma \cdot P$ since $\mathsf{mpk} = \theta \cdot P$, $Q_{\mathsf{PID_i}} = \gamma \cdot P$.

In the end, the $\mathcal{C}$ is able to find $\theta \cdot \gamma \cdot P$ which is the solution to the ECCDH assumption. Next, we will analyse the advantage of the $\mathcal{C}$ in winning the game. The $\mathcal{C}$ advantage is based on the occurrence of the events in which the game aborts. The $\mathcal{C}$ aborts the game under the following conditions:

- The secret value query where the game aborts for $\mathsf{PID_i} = \mathsf{PID}^*_{\mathsf{i}}$. The probability is $\Pr(E_{q_{\mathsf{sv}}}) = 1/q_{\mathsf{sv}}$.
- An unsigncryption query where the game aborts due to invalid m. The probability is $\Pr(E_{q_{\mathsf{usc}}}) = q_{\mathsf{usc}}/2^k$.
- In the challenge phase, the adversary queries for $\mathsf{PID}^*_{\mathsf{r_i}} \neq \mathsf{PID}^*_{\mathsf{i}}$. The probability is $\Pr(E_{q_{H_1}}) = (1 - 1/qH_1)$.

Moreover, the $\mathcal{C}$ fetches the list L_1 to retrieve $Q_{\mathsf{PID_i}}$ and L_2 to retrieve $Z_{1_{\mathsf{i}}}$ and evaluates $\theta \cdot \gamma \cdot P$ with probability $(1/q_{H_1} + 1/q_{H_2})$. Therefore, the probability of the $\mathcal{C}$ winning the game with advantage ϵ' is:

$$\epsilon' \geq \epsilon \left(\frac{1}{qH_1} + \frac{1}{qH_2}\right)\left(\frac{1}{qH_1}\right)\left(1 - \frac{1}{q_{\mathsf{sv}}}\right)\left(1 - \frac{q_{\mathsf{usc}}}{2^k}\right) \tag{4}$$

Unforgeability

Theorem 3. *The proposed scheme is EUF-CMA-I secure under the ROM based on the hardness of the ECDL assumption. Suppose that the EUF-CMA-I adversary $\mathcal{A}_{\mathrm{I}}$ has a non-negligible advantage ϵ in winning the game then, there is $\mathcal{C}$ that can solve the ECDL with the non-negligible advantage ϵ'.*

Proof. Given two random instances of the ECDL $(Q, P) \in \mathbb{G}$ where $Q = \phi \cdot P$. The $\mathcal{C}$ has to find ϕ by interacting with the $\mathcal{A}_{\mathrm{I}}$.

1. ***Phase-1***: A polynomially bounded number of queries q are made by an $\mathcal{A}_{\mathrm{I}}$. The Challenger $\mathcal{C}$ keeps a list L_l of q_{H_l} to record the responses.
   - **Setup**: The $\mathcal{C}$ runs the setup algorithm to generate $\mathsf{PP} = \{\mathbb{G}, E, P, q, H_0, H_1, H_2, H_3\}$. The $\mathcal{C}$ sets $\mathsf{mpk} = \theta \cdot P$ and sends PP and mpk to the $\mathcal{A}_{\mathrm{I}}$. The $\mathcal{A}_{\mathrm{I}}$ selects a target identity $\mathsf{PID}_{\mathsf{s}}^*$.

2. ***Phase-2***: The $\mathcal{A}_{\mathrm{I}}$ asks a number of queries including q_{pk}, q_{ppk}, q_{pr}, q_{sv}, and q_{sc}. The $\mathcal{C}$ maintains an initially empty list L_{pk} to store the pk and sv information. $\mathcal{C}$ responds to all queries as in *Phase-2* of Theorem 1, except the responds to q_{ppk} as follows:
   - q_{ppk}: Upon receiving the query, if $\mathsf{PID} = \mathsf{PID}_{\mathsf{s}}^*$, the $\mathcal{C}$ aborts. Otherwise, if it exists in the list L_{pk}, the $\mathcal{C}$ sends $\mathsf{d_i}$ to $\mathcal{A}_{\mathrm{I}}$, if it does not, the $\mathcal{C}$ randomly chooses $\phi \in \mathbb{Z}_q^*$ and computes $\mathsf{d_i} = \phi Q_{\mathsf{PID_i}}$. The $\mathcal{C}$ returns $\mathsf{d_i} = \phi Q_{\mathsf{PID_i}}$ to $\mathcal{A}_{\mathrm{I}}$ and updates the tuple $(\mathsf{PID_i}, \mathsf{d_i}, \mathsf{pk_i}, x_{\mathsf{i}})$ in L_{pk}.

3. ***Forgery***: Taking the target sender's $\mathsf{PID}_{\mathsf{s}}^*$ and designated receiver's $\mathsf{PID}_{\mathsf{r_i}}$, the adversary outputs a forged $\mathsf{CT}^* = (\mathsf{C}_1^*, \mathsf{C}_2^*)$ on m^* where $\mathsf{C}_1^* = (f^*, S_i^*)$ and $\mathsf{C}_2^* = \mathsf{Enc}_{\mathsf{K}}^*(m)$ which is the valid signcrypted ciphertext and is not the result of signcryption oracle.
   - Case-1 ($\mathsf{PID} \neq \mathsf{PID}_{\mathsf{s}}^*$): The $\mathcal{C}$ returns $\perp$.
   - Case-2 ($\mathsf{PID} = \mathsf{PID}_{\mathsf{s}}^*$): The $\mathcal{C}$ extracts the list L_{pk} for the record $(\mathsf{PID}_{\mathsf{i}}^*, \mathsf{d}_{\mathsf{i}}^*, \mathsf{pk}_{\mathsf{i}}^*, x_{\mathsf{i}}^*)$ and L_3 for the record (m^*, ψ^*, f^*).

   According to Forking Lemma, $\mathcal{C}$ replays the $\mathcal{A}_{\mathrm{I}}$ with the same random tape but distinct attributes from H_1 and H_3. It implies that, $h_1^* = H_1(\mathsf{mpk}, \mathsf{PID}_{\mathsf{i}}^*)$ and $h_1'^* = H_1(\mathsf{mpk}, \mathsf{PID}_{\mathsf{i}}^*)$, and $h_1^* \neq h_1'^*$ i.e. $Q_{\mathsf{PID_s}}^* \neq Q_{\mathsf{PID_s}}'^*$. Similarly, $h_3^* = H_3(m^*, \psi^*, \mathsf{PID}_{\mathsf{s}}^*, \mathsf{PID}_{\mathsf{r_i}}^*, \mathsf{pk}_{\mathsf{s}}^*, \mathsf{pk}_{\mathsf{r_i}}^*)$, $h_3'^* = H_3(m^*, \psi^*, \mathsf{PID}_{\mathsf{s}}^*, \mathsf{PID}_{\mathsf{r_i}}^*, \mathsf{pk}_{\mathsf{s}}^*, \mathsf{pk}_{\mathsf{r_i}}^*)$ and $h_3^* \neq h_3'^*$ i.e. $f^* \neq f'^*$. Finally, the $\mathcal{A}_{\mathrm{I}}$ outputs another forged $\mathsf{CT}'^* = (\mathsf{C}_1^*, \mathsf{C}_2^*)$ on the same m^* where $\mathsf{C}_1'^* = (f'^*, S_i'^*)$ and $\mathsf{C}_2^* = \mathsf{Enc}_{\mathsf{K}}^*(m)$. Finally, $\mathcal{C}$ will have two valid signatures:

$$S_{\mathsf{i}}^* = r^{*-1}(f^* + w.\mathsf{d}_{\mathsf{s}}^*.x_{\mathsf{s}}) \tag{5}$$

$$S_{\mathsf{i}}'^* = r'^{*-1}(f'^* + w.\mathsf{d}_{\mathsf{s}}'^*.x_{\mathsf{s}}) \tag{6}$$

   where $r^* = r'^*$ and $\mathsf{d}_{\mathsf{s}}^* = \mathsf{d}_{\mathsf{s}}'^*$. From the Equations 8 and 9 above, $\mathcal{C}$ can extract ϕ as follows:

$$\phi = r^{*-1}(f'^* - f^*) + (S_{\mathsf{i}}^* - S_{\mathsf{i}}'^*)(r^{*-1}(w.x_{\mathsf{s}}(Q_{\mathsf{PID_s}}^* - Q_{\mathsf{PID_s}}'^*))^{-1}$$

   Given that, the $\mathcal{C}$ solves the ECDL assumption $Q = \phi$ with the advantage ϵ':

$$\epsilon' \geq \epsilon \left(\frac{1}{qH_1} + \frac{1}{qH_2}\right)\left(\frac{1}{qH_1}\right)\left(1 - \frac{1}{q_{\mathsf{ppk}}}\right)\left(1 - \frac{q_{\mathsf{usc}}}{2^k}\right) \tag{7}$$

Theorem 4. *The proposed scheme is EUF-CMA-II secure under the ROM based on the hardness of the ECDL assumption. Suppose that the EUF-CMA-II adversary $\mathcal{A}_{\mathrm{II}}$ has a non-negligible advantage ϵ in winning the game then, there is $\mathcal{C}$ that can solve the ECDL assumption with the non-negligible advantage ϵ'.*

Proof. Given two random instances of the ECDL $(Q, P) \in \mathbb{G}$ where $Q = \pi \cdot P$ where $\pi \in \mathbb{Z}_q^*$. The $\mathcal{C}$ has to find π by interacting with the $\mathcal{A}_{\text{II}}$ such that $Q = \pi \cdot P$.

1. ***Phase-1***: It queries are similar to Theorem 2, respectively. The $\mathcal{C}$ keeps a list L_l of q_{H_l} to record the responses.
   - **Setup**: The $\mathcal{C}$ runs the setup algorithm to generate $\mathsf{PP} = \{\mathbb{G}, E, P, q, H_0, H_1, H_2, H_3\}$. The $\mathcal{C}$ sets $\mathsf{mpk} = \theta \cdot P$ and sends PP and mpk to the$\mathcal{A}_{\text{II}}$.

2. ***Phase-2***: The adversary $\mathcal{A}_{\text{II}}$ asks a number of queries including q_{pk}, q_{ppk}, q_{pr}, q_{sv}, and q_{sc}. The $\mathcal{C}$ maintains an initially empty list L_{pk} to store the pk and sv values. $\mathcal{C}$ responds to all queries as in Phase-2 of Theorem 2, except the responds to secret value extract query q_{sv} as follows:
   - q_{sv}: Upon receiving the query for PID, the $\mathcal{C}$ checks if $\mathsf{PID} = \mathsf{PID}_{\mathsf{s}}^*$. If it holds, the $\mathcal{C}$ aborts because in this case, the PID is a target identity. Otherwise, it checks if x_{i} exists in the list L_{pk} $(\mathsf{PID}_{\mathsf{i}}, \mathsf{d}_{\mathsf{i}}, \mathsf{pk}_{\mathsf{i}}, x_{\mathsf{i}})$. If it exists, the $\mathcal{C}$ returns x_{i} to $\mathcal{A}_{\text{II}}$. Otherwise, $\mathcal{C}$ computes $\mathsf{pk}_{\mathsf{i}} = \pi \cdot P$ where $x_{\mathsf{i}} = \pi \in \mathbb{Z}_q^*$ and adds the tuple $(\mathsf{PID}_{\mathsf{i}}, \mathsf{d}_{\mathsf{i}}, \mathsf{pk}_{\mathsf{i}}, x_{\mathsf{i}})$ in L_{pk} and returns x_{i} to $\mathcal{A}_{\mathcal{II}}$.

3. ***Forgery***: Taking the target sender $\mathsf{PID}_{\mathsf{s}}^*$ and designated receiver's $\mathsf{PID}_{\mathsf{r_i}}$, the adversary outputs a forged $\mathsf{CT}^* = (\mathsf{C}_1^*, \mathsf{C}_2^*)$ on m^* where $\mathsf{C}_1^* = (f^*, S_i^*)$ and $\mathsf{C}_2^* = \mathsf{Enc}_{\mathsf{K}}^*(m)$ which is the valid signcrypted ciphertext and is not the result of signcryption oracle.
   - Case-1 ($\mathsf{PID} \neq \mathsf{PID}_{\mathsf{s}}^*$): The $\mathcal{C}$ returns $\perp$.
   - Case-2 ($\mathsf{PID} = \mathsf{PID}_{\mathsf{s}}^*$): The $\mathcal{C}$ extracts the list L_{pk} for the record $(\mathsf{PID}_{\mathsf{i}}^*, \mathsf{d}_{\mathsf{i}}^*, \mathsf{pk}_{\mathsf{i}}^*, x_{\mathsf{i}}^*)$ and L_3 for the record (m^*, ψ^*, f^*).

   According to the Forking Lemma, the $\mathcal{C}$ replays the $\mathcal{A}_{\text{II}}$ with the same random tape but distinct attributes from H_1 and H_3. It implies that, $h_1^* = H_1(\mathsf{mpk}, \mathsf{PID}_{\mathsf{i}}^*)$ i.e. , $h_1'^* = H_1(\mathsf{mpk}, \mathsf{PID}_{\mathsf{i}}^*)$ and $h_1^* \neq h_1'^*$ i.e. $Q_{\mathsf{PID}_{\mathsf{s}}}^* \neq Q_{\mathsf{PID}_{\mathsf{s}}}'^*$. Similarly, $h_3^* = H_3(m^*, \psi^*, \mathsf{PID}_{\mathsf{s}}^*, \mathsf{PID}_{\mathsf{r_i}}^*, \mathsf{pk}_{\mathsf{s}}^*, \mathsf{pk}_{\mathsf{r_i}}^*)$, $h_3'^* = H_3(m^*, \psi^*, \mathsf{PID}_{\mathsf{s}}^*, \mathsf{PID}_{\mathsf{r_i}}^*, \mathsf{pk}_{\mathsf{s}}^*, \mathsf{pk}_{\mathsf{r_i}}^*)$, and $h_3^* \neq h_3'^*$ i.e. $f^* \neq f'^*$. In the end, the $\mathcal{A}_{\text{II}}$ outputs another forged $\mathsf{CT}'^* = (\mathsf{C}_1^*, \mathsf{C}_2^*)$ on the same m^* where $\mathsf{C}_1'^* = (f'^*, S_i'^*)$ and $\mathsf{C}_2^* = \mathsf{Enc}_{\mathsf{K}}^*(m)$. Finally, $\mathcal{C}$ will have two valid signatures:

$$S_{\mathsf{i}}^* = r^{*-1}(f^* + w.\mathsf{d}_{\mathsf{s}}^* x_{\mathsf{s}}^*) \tag{8}$$

$$S_{\mathsf{i}}'^* = r'^{*-1}(f'^* + w.\mathsf{d}_{\mathsf{s}}^* x_{\mathsf{s}}'^*) \tag{9}$$

where $r^* = r'^*$ and $x_{\mathsf{s}}^* = x_{\mathsf{s}}'^*$. From the Eq. 8 and 9 above, the $\mathcal{C}$ can extract π as follows:

$$\pi = r^{*-1}(f'^* - f^*) + (S_{\mathsf{i}}^* - S_{\mathsf{i}}'^*)(r^{*-1}(w.\mathsf{mpk}.(Q_{\mathsf{PID}_{\mathsf{s}}}^* - Q_{\mathsf{PID}_{\mathsf{s}}'}^*))^{-1}$$

Given that, the $\mathcal{C}$ solves the ECDL assumption $Q = \pi \cdot P$ with the advantage ϵ':

$$\epsilon' \geq \epsilon \left(\frac{1}{qH_1} + \frac{1}{qH_2}\right)\left(\frac{1}{qH_1}\right)\left(1 - \frac{1}{q_{\mathsf{sv}}}\right)\left(1 - \frac{q_{\mathsf{usc}}}{2^k}\right) \tag{10}$$

Anonymity: In the proposed scheme, each user uses the PID to communicate with each other instead of the ID_{R}. Therefore, the users will be able to validate the identity but cannot detect or modify the ID_{R}. Each user create a PID from the ID_{R} as

$\mathsf{PID} = \mathsf{ID_R} \oplus (\alpha.\mathsf{pk_{RA}})$ where α is chosen randomly and $R = \alpha \cdot P$. In order to obtain the $\mathsf{ID_R}$, the attacker need to calculate the $\mathsf{ID_R} = \mathsf{PID} \oplus (R \cdot v)$ However, it is based on ECDL hard assumption therefore, the attacker will not be able to compute $\mathsf{ID_R}$. Moreover, to validate the $\mathsf{ID_R}$ of the user, RA will verify the $\mathsf{ID_R}$ using its private key as $\mathsf{ID_R} = H_0(R \cdot v)$. Since, only RA knows its private key, no else could generate the $\mathsf{ID_R}$. The scheme also provide the conditional anonymity i.e. in case of a dispute, the RA will expose the $\mathsf{ID_R}$ of the user.

Non-repudiation: Non-repudiation refers to the concept which ensures that a user cannot later deny sending a message by adding some of its unique information to the message. In communication, non-repudiation is typically achieved through the use of signature, in which the sender signs message with its $\mathsf{sk_s}$ and the message is verified using $\mathsf{pk_s}$. By signing message with their $\mathsf{sk_s}$, the sender proves that they sent the message and cannot later deny it since, only the sender knows its $\mathsf{sk_s}$. Similarly, in our scheme, message is signed by the sender with its $\mathsf{sk_s}$ as $S_\mathsf{i} = r^{-1}(f + w.\mathsf{d_s}x_\mathsf{s})$. The message is verified by the receiver using $\mathsf{pk_s}$ as $R_\mathsf{i} = S_\mathsf{i}^{-1}(f.P + w.\mathsf{pk_s}.Z_{1_\mathsf{i}}.Q^{-1}_{\mathsf{PID}_{\mathsf{r_i}}})$. Since, the sender signs message with its $\mathsf{sk_s}$ that only sender knows, it cannot deny sending a message. Hence, our scheme achieves non-repudiation.

Forward Security: Forward security is a property that ensures the security of a message even if the sk of the user is compromised. The $\mathcal{A}$ cannot extract previously exchanged messages during communication and the messages remain secure. It is typically achieved by using key agreement protocol which generate a new key for each session. Even if the key for one session is compromised, other sessions cannot be exploited by the $\mathcal{A}$. In our scheme, the symmetric session key K and its encapsulation $\mathsf{C_1}$ is generated using the $\mathsf{sk_s}$, receiver's public key $\mathsf{pk_{r_i}}$, along with the random parameter $r \in \mathbb{Z}_q^*$ which is different for each session. In this case, even if the $\mathsf{sk_{r_i}}$ is exploited, which is used for the decapsulation of $\mathsf{C_1}$ to generate K, the $\mathcal{A}$ cannot extract the m, since r is always randomly chosen for each session. Therefore, our scheme ensure that m remains secure during communication.

8 Performance Analysis

We compare the computational cost, communication cost, and security requirements of the proposed AMCLHS scheme with existing multireceiver signcryption schemes. The computational overhead of multireceiver schemes is compared with [15,17,19] as shown in Table 1. Among the multireceiver signcryption schemes, Niu et al. [15] have highest computational overhead, utilizing a total of $(2n+4)P+1M+(2n+2)E$ operations, with $(2n+4)P$ pairing operations, which are considered as the most expensive and time consuming. Peng at al. [19] require $(2n+2)M$ operations for signcryption and same number of operations in the unsigncryption phase. Niu et al. [17] require a total of $(4n+4)M$ operations for signcryption and unsigncryption. Contrasting with existing solutions, our proposed scheme delivers high efficiency with only $(2n+5)M$ total operations. It uniquely pairs a linear signcryption cost with a constant unsigncryption cost per receiver, regardless of scale. This optimal combination results in a predictable, scalable system, setting a new performance standard in VANETs. Given its scalability

and robustness, our scheme emerges as a compelling choice for larger, more complex VANETs, providing a significant upgrade over existing schemes.

The Table 2 shows the communication cost in terms of size of the ciphertext generated by each scheme [15,19,15] for signcryption and unsigncryption. The Table 2 indicates that the proposed AMCLHS scheme has the optimal communication cost among the four schemes, as it only requires $n|m| + |\mathbb{Z}_q^*| + |\mathbb{G}| + |\mathsf{K}|$ bits to signcrypt a message. Moreover, our scheme has linear communication cost in signcryption while, unsigncryption cost remains constant.

Table 1. Computational Overhead Comparison with Multireceiver Schemes

Schemes	Signcryption	Unsigncryption	Total
Niu et al. [15]	$2nP + 1M + 2nE$	$4P + 2E$	$(2n+4)P + 1M + (2n+2)E$
Peng et al. [19]	$(2n+1)M$	$4M$	$(2n+5)M$
Niu et al. (2022) [17]	$(2n+2)M$	$(2n+2)M$	$(4n+4)M$
Our scheme	$(2n+2)M$	$3M$	$(2n+5)M$

Legend: (P) Bilinear Pairing operation, (M) Point multiplication operation, and (E) Exponentiation operation in $\mathbb{Z}_q^*$

Table 2. Communication Cost

Schemes	Ciphertext Length	Complexity of Communication	
		Signcryption	Unsigncryption
Niu et al. [15]	$n\|m\| + \|\mathbb{G}\| + 2n\|\mathbb{G}\|$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Peng et al. [19]	$n\|m\| + (n+2)\|\mathbb{Z}_q^*\|$	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
Niu et al. [17]	$n\|(m+2)\| + 2\|\mathbb{G}\| + 2\|\mathbb{Z}_q^*\|$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Our scheme	$n\|m\| + \|\mathbb{Z}_q^*\| + \|\mathbb{G}\| + \|\mathsf{K}\|$	$\mathcal{O}(n)$	$\mathcal{O}(1)$

Legend: n is the number of users, $|m|$ is the plaintext Length, $|\mathbb{Z}_q^*|$ is the length of an element in finite field $\mathbb{Z}_q^*$, $|\mathbb{G}|$ is the length of an element in $\mathbb{G}$, and $|\mathsf{K}|$ is the size of the symmetric key.

In Table 3, we present a comparative analysis of the security requirements between our scheme and existing multireceiver hybrid signcryption schemes [15,17,19]. The comparison parameters include confidentiality, unforgeability, anonymity, non-repudiation, and forward security.

Our proposed scheme successfully achieves all security requirements as shown in Table 3, offering superior efficiency with lower computational costs, setting it apart from the others.

9 Conclusion

Our paper introduces a novel mKEM-DEM based AMCLHS scheme for broadcast communication. The proposed scheme is based on the multi-recipient Key Encapsula-

Table 3. Security requirements

Schemes	Confidentiality	Unforgeability	Anonymity	Non-repudiation	Forward Security
Niu et al. [15]	✓	✓	✓	✓	✓
Niu et al. [17]	✓	✓	✓	×	×
Peng et al. [19]	✓	✓	✓	×	×
Our scheme	✓	✓	✓	✓	✓

tion Mechanism (mKEM) and Data Encapsulation Mechanism (DEM) that generates a symmetric key using the public and private key pair of the users. The message is then signcrypted with the previously generated symmetric key and the private key of the sender. We provide a detailed security analysis using ECCDH and ECDL hard assumptions and demonstrate that the scheme is secure against *IND-CCA2* and *EUF-CMA* attacks for Type-I and Type-II adversaries. Moreover, in this scheme, each user is assigned a PID to ensure user anonymity. Lastly, we compare our scheme with existing single receiver and multireceiver certificateless hybrid signcryption schemes in terms of computation cost, communication cost, and security requirements. We show that the proposed scheme has less communication cost and is computationally more efficient, with the signcryption cost linear with the number of designated receivers while the unsigncryption cost remains constant and simultaneously achieves confidentiality, unforgeability, anonymity, non-repudiation, and forward security.

References

1. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C. (ed.) Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 - December 4, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2894, pp. 452–473. Springer (2003), https://doi.org/10.1007/978-3-540-40061-5_29
2. Barbosa, M., Farshim, P.: Certificateless signcryption. In: Abe, M., Gligor, V.D. (eds.) Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2008, Tokyo, Japan, March 18-20, 2008. pp. 369–372. ACM (2008), https://doi.org/10.1145/1368310.1368364
3. Chen, X., He, D., Khan, M.K., Luo, M., Peng, C.: A secure certificateless signcryption scheme without pairing for internet of medical things. IEEE Internet Things J. **10**(10), 9136–9147 (2023). https://doi.org/10.1109/JIOT.2022.3233180, https://doi.org/10.1109/JIOT.2022.3233180
4. Cui, B., Lu, W., He, W.: A new certificateless signcryption scheme for securing internet of vehicles in the 5g era. Security and Communication Networks **2022** (2022)
5. Dent, A.W.: Hybrid signcryption schemes with insider security. In: Boyd, C., Nieto, J.M.G. (eds.) Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3574, pp. 253–266. Springer (2005). https://doi.org/10.1007/11506157_22
6. Dent, A.W.: Hybrid signcryption schemes with outsider security. In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) Information Security, 8th International Conference, ISC 2005, Singapore, September 20-23, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3650, pp. 203–217. Springer (2005). https://doi.org/10.1007/11556992_15

7. Fu, M., Gu, X., Dai, W., Lin, J., Wang, H.: Secure multi-receiver communications: Models, proofs, and implementation. In: Wen, S., Zomaya, A.Y., Yang, L.T. (eds.) Algorithms and Architectures for Parallel Processing - 19th International Conference, ICA3PP 2019, Melbourne, VIC, Australia, December 9-11, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11944, pp. 689–709. Springer (2019). https://doi.org/10.1007/978-3-030-38991-8_45
8. Gong, B., Wu, Y., Wang, Q., Ren, Y., Guo, C.: A secure and lightweight certificateless hybrid signcryption scheme for internet of things. Future Gener. Comput. Syst. **127**, 23–30 (2022). https://doi.org/10.1016/j.future.2021.08.027
9. Hongzhen, D., Qiaoyan, W., Shanshan, Z., Mingchu, G.: A pairing-free certificateless signcryption scheme for vehicular ad hoc networks. Chinese Journal of Electronics **30**(5), 947–955 (2021)
10. Imghoure, A., El-Yahyaoui, A., Omary, F.: Ecdsa-based certificateless conditional privacy-preserving authentication scheme in vehicular ad hoc network. Veh. Commun. **37**, 100504 (2022). https://doi.org/10.1016/j.vehcom.2022.100504, https://doi.org/10.1016/j.vehcom.2022.100504
11. Kasyoka, P.N., Kimwele, M.W., Mbandu, A.S.: Efficient certificateless signcryption scheme for wireless sensor networks in ubiquitous healthcare systems. Wirel. Pers. Commun. **118**(4), 3349–3366 (2021). https://doi.org/10.1007/s11277-021-08183-y
12. Li, F., Shirase, M., Takagi, T.: Certificateless hybrid signcryption. In: Bao, F., Li, H., Wang, G. (eds.) Information Security Practice and Experience, 5th International Conference, ISPEC 2009, Xi'an, China, April 13-15, 2009, Proceedings. Lecture Notes in Computer Science, vol. 5451, pp. 112–123. Springer (2009). https://doi.org/10.1007/978-3-642-00843-6_11
13. Li, X., Jiang, C., Du, D., Wang, S., Fei, M., Wu, L.: A novel efficient signcryption scheme for resource-constrained smart terminals in cyber-physical power systems. CoRR **abs/2212.04198** (2022). https://doi.org/10.48550/arXiv.2212.04198
14. Malone-Lee, J.: Identity-based signcryption. IACR Cryptol. ePrint Arch. p. 98 (2002), http://eprint.iacr.org/2002/098
15. Niu, S., Niu, L., Yang, X., Wang, C., Jia, X.: Heterogeneous hybrid signcryption for multi-message and multi-receiver. PloS one **12**(9), e0184407 (2017)
16. Niu, S., Shao, H., Hu, Y., Zhou, S., Wang, C.: Privacy-preserving mutual heterogeneous signcryption schemes based on 5g network slicing. IEEE Internet Things J. **9**(19), 19086–19100 (2022). https://doi.org/10.1109/JIOT.2022.3163607
17. Niu, S., Zhou, S., Fang, L., Hu, Y., Wang, C.: Broadcast signcryption scheme based on certificateless in wireless sensor network. Comput. Networks **211**, 108995 (2022). https://doi.org/10.1016/j.comnet.2022.108995, https://doi.org/10.1016/j.comnet.2022.108995
18. Parai, K., Islam, S.H.: Iot-rrhm: Provably secure iot-based real-time remote healthcare monitoring framework. J. Syst. Archit. **138**, 102859 (2023). https://doi.org/10.1016/j.sysarc.2023.102859, https://doi.org/10.1016/j.sysarc.2023.102859
19. Peng, C., Chen, J., Obaidat, M.S., Vijayakumar, P., He, D.: Efficient and provably secure multireceiver signcryption scheme for multicast communication in edge computing. IEEE Internet Things J. **7**(7), 6056–6068 (2020). https://doi.org/10.1109/JIOT.2019.2949708
20. Qiu, J., Fan, K., Zhang, K., Pan, Q., Li, H., Yang, Y.: An efficient multi-message and multi-receiver signcryption scheme for heterogeneous smart mobile iot. IEEE Access **7**, 180205–180217 (2019). https://doi.org/10.1109/ACCESS.2019.2958089

21. Selvi, S.S.D., Vivek, S.S., Rangan, C.P.: Certificateless KEM and hybrid signcryption schemes revisited. IACR Cryptol. ePrint Arch. p. 462 (2009), http://eprint.iacr.org/2009/462
22. Selvi, S.S.D., Vivek, S.S., Shukla, D., Rangan, C.P.: Efficient and provably secure certificateless multi-receiver signcryption. In: Baek, J., Bao, F., Chen, K., Lai, X. (eds.) Provable Security, Second International Conference, ProvSec 2008, Shanghai, China, October 30 - November 1, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5324, pp. 52–67. Springer (2008). https://doi.org/10.1007/978-3-540-88733-1_4
23. Smart, N.P.: Efficient key encapsulation to multiple parties. In: Blundo, C., Cimato, S. (eds.) Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3352, pp. 208–219. Springer (2004). https://doi.org/10.1007/978-3-540-30598-9_15
24. Wu, Y., Gong, B., Zhang, Y., et al.: An improved efficient certificateless hybrid signcryption scheme for internet of things. Wireless Communications and Mobile Computing **2022** (2022)
25. Yang, Y., He, D., Vijayakumar, P., Gupta, B.B., Xie, Q.: An efficient identity-based aggregate signcryption scheme with blockchain for iot-enabled maritime transportation system. IEEE Trans. Green Commun. Netw. **6**(3), 1520–1531 (2022). https://doi.org/10.1109/TGCN.2022.3163596
26. Yin, A., Liang, H.: Certificateless hybrid signcryption scheme for secure communication of wireless sensor networks. Wirel. Pers. Commun. **80**(3), 1049–1062 (2015). https://doi.org/10.1007/s11277-014-2070-y
27. Yu, X., Zhao, W., Tang, D.: Efficient and provably secure multi-receiver signcryption scheme using implicit certificate in edge computing. J. Syst. Archit. p. 102457 (2022). https://doi.org/10.1016/j.sysarc.2022.102457
28. Yu, Y., Yang, B., Huang, X., Zhang, M.: Efficient identity-based signcryption scheme for multiple receivers. In: Xiao, B., Yang, L.T., Ma, J., Müller-Schloer, C., Hua, Y. (eds.) Autonomic and Trusted Computing, 4th International Conference, ATC 2007, Hong Kong, China, July 11-13, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4610, pp. 13–21. Springer (2007), https://doi.org/10.1007/978-3-540-73547-2_4
29. Zhang, W., Zhang, Y., Guo, C., An, Q., Guo, Y., Liu, X., Zhang, S., Huang, J.: Certificateless hybrid signcryption by a novel protocol applied to internet of things. Computational Intelligence and Neuroscience **2022** (2022)
30. Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) $<<$ cost(signature) + cost(encryption). In: Jr., B.S.K. (ed.) Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1294, pp. 165–179. Springer (1997). https://doi.org/10.1007/BFb0052234
31. ZHOU, C.: Certificateless signcryption scheme without random oracles. Chinese Journal of Electronics **27**(5), 1002–1008 (2018)