

Title	Anonymous multi-receiver certificateless hybrid signcryption for broadcast communication
Authors	Umrani, Alia;Vangujar, Apurva K.;Palmieri, Paolo
Publication date	2024-02-28
Original Citation	Umrani, A., Vangujar, A. and Palmieri, P. (2024) 'Anonymous multi-receiver certificateless hybrid signcryption for broadcast communication', Proceedings of the 10th International Conference on Information Systems Security and Privacy. Rome, Italy, 26-28 February, pp. 733-744. https://doi.org/10.5220/0012353400003648
Type of publication	Conference item
Link to publisher's version	https://www.scitepress.org/PublicationsDetail.aspx?ID=IJevPrVhldY=&t=1 - 10.5220/0012353400003648
Rights	© 2024, by SCITEPRESS – Science and Technology Publications, Lda. For the purpose of Open Access, the author has applied a CC BY NC ND public copyright licence to any Author Accepted Manuscript version arising from this submission - https://creativecommons.org/licenses/by-nc-nd/4.0/
Download date	2026-10-31 03:20:44
Item downloaded from	https://hdl.handle.net/10468/18218

Anonymous Multi-Receiver Certificateless Hybrid Signcryption for Broadcast Communication

Alia Umrani^{*a}, Apurva K Vangujar^{*b} and Paolo Palmieri^{cd}

Department of Computing & IT, University College Cork, Cork, Ireland

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

Abstract: Confidentiality, authentication, and anonymity are fundamental security requirements in broadcast communication achievable by Digital Signature (DS), encryption, and Pseudo-Identity (PID) techniques. Signcryption, particularly hybrid signcryption, offers both DS and encryption more efficiently than “sign-then-encrypt”, with lower computational and communication costs. This paper proposes an Anonymous Multi-receiver Certificateless Hybrid Signcryption (AMCLHS) scheme 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 key, with the sender’s private key, is used in Data Encapsulation Mechanism (DEM) to signcrypt the message, ensuring confidentiality and authentication. The scheme generates identical ciphertext for multiple recipients while maintaining their anonymity by assigning a PID to each user. Security notions are demonstrated for indistinguishability against chosen-ciphertext attack using the elliptic curve computational diffie-hellman assumption in the random oracle model and existential unforgeability against chosen message attack under elliptic curve diffie-hellman assumption. The AMCLHS scheme operates in a multireceiver certificateless environment, preventing the key escrow problem. Comparative analysis shows that our scheme is computationally efficient, provides optimal communication cost, and simultaneously ensures confidentiality, authentication, anonymity, non-repudiation, and forward security.

1 INTRODUCTION

Confidentiality, authentication, and anonymity are the basic security requirements in broadcast communication (Peng et al., 2020). The current solution to provide for these security requirements are encryption and Digital Signature (DS). However, the traditional “sign-then-encrypt” approach results in high computational costs. Signcryption, on the other hand, allows both the encryption and DS operations to be performed simultaneously, providing both the confidentiality and authentication more efficiently. Signcryption was first proposed by Zhang *et al.* as a novel cryptographic primitive (Zheng, 1997). Malone-Lee proposed the first Identity (ID)-based signcryption

scheme that provides forward security and public verifiability (Malone-Lee, 2002). 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.* proposed a Certificateless Public Key Cryptography (CLPKC) (Al-Riyami and Paterson, 2003). In CLPKC, the Key Generation Center (KGC) only generates a partial private key (ppk) of the user. The user then combines ppk and a secret value to generate the actual private and public key pair. Therefore, the KGC does not have knowledge of the user’s complete private key. Following that, Barbosa and Farshim proposed the first certificateless signcryption scheme (Barbosa and Farshim, 2008). The signcryption methods mentioned above are designed for single receiver, which are not suitable for broadcast communication. When sending the same message to multiple recipients, the user encrypts a message for each individual recipient, increasing computation time and communication lag. To address this, Selvi *et al.* pro-

^a <https://orcid.org/0000-0003-1885-3629>

^b <https://orcid.org/0000-0002-8194-4593>

^c <https://orcid.org/0000-0002-9819-4880>

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

posed the first multireceiver certificateless signcryption scheme (Selvi et al., 2008). 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 (Selvi et al., 2008). (ii) Hybrid signcryption: Hybrid signcryption provides the advantages of combining symmetric key encryption with asymmetric key signature while ensuring integrity, authentication, and non-repudiation (Selvi et al., 2009). 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 (Dent, 2005b; Dent, 2005a) on Hybrid signcryption schemes.

In this paper, we propose a multi-receiver anonymous certificateless hybrid signcryption based on multi-recipient Key Encapsulation Mechanism-Data Encapsulation Mechanism (mKEM-DEM) for broadcast communication. For confidentiality, we prove Indistinguishability against Chosen-Ciphertext Attack (*ind-cca2-I*) for Type-I adversary, and (*ind-cca2-II*) for Type-II adversary using Elliptic Curve based Computational Diffie-Hellman (ECCDH) 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) assumption. Additionally, to ensure anonymity, each user is assigned a Pseudo-Identity (PID) and we further 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, 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 mKEM and DEM. Our main contributions are as follows:

1. We propose an Anonymous Multireceiver Certifi-

cateless Hybrid Signcryption (AMCLHS) scheme based on mKEM-DEM. The AMCLHS scheme uses a combination of PKC and symmetric key to signcrypt a 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 real identities anonymous from each other.
3. The scheme operates in a multireceiver certificateless environment, preventing the key escrow problem. We achieve confidentiality by demonstrating security against *ind-cca2-I* and *ind-cca2-II* and unforgeability by demonstrating *euf-cma-I* and *euf-cma-II*, respectively. The security is proven using ECCDH and ECDL assumptions under the Random Oracle Model (ROM).

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

2 RELATED WORK

2.1 Certificateless Signcryption

Signcryption was first introduced by Zheng *et al.* in 1997 combining the signature and encryption to provide authentication and confidentiality more efficiently than sign-then-encrypt (Zheng, 1997). Several ID-based signcryption schemes have been proposed, however, the key issue is the presence of a key escrow problem. To address this, Barbosa and Farshim proposed the first certificateless signcryption scheme that provides both confidentiality and authentication and is secure under the ROM (Barbosa and Farshim, 2008). Chen *et al.* and Cui *et al.* proposed a certificateless signcryption scheme for the Internet of Medical Things without pairings and the Internet of Vehicles (IoVs), respectively. The schemes provides confidentiality and authentication and proves security under ECDL and Computational Diffie-Hellman (CDH) assumptions (Chen et al., 2023; Cui et al., 2022). Similarly, a certificateless signcryption scheme without ROM was proposed by ZHOU *et al.* that achieves

confidentiality and unforgeability however, does not provide anonymity (ZHOU, 2018). Kasyoka *et al.* proposed a certificateless signcryption for wireless sensor networks (Kasyoka et al., 2021). Additionally, Cui *et al.* presented a pairing-free certificateless signcryption scheme for the IoVs (Cui et al., 2022). Li *et al.* proposed a signcryption scheme for resource-constrained smart terminals in cyber-physical power systems (Li et al., 2022). However, all the aforementioned schemes are designed for single receivers, which are not suitable for broadcast communication. Yu *et al.* introduced the first multireceiver signcryption scheme based on ID-based PKC, enabling message encryption for n designated receivers (Yu et al., 2007). 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.* proposed a privacy-preserving mutual heterogeneous signcryption scheme based on 5G network slicing, where the sender is in a public key infrastructure environment, and the receiver is in a certificateless environment (Niu et al., 2022a). 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, Internet of Things (IoT), and IoT-enabled maritime transportation systems (Peng et al., 2020; Qiu et al., 2019; Yang et al., 2022; Yu et al., 2022). 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 (Dent, 2005b; Dent, 2005a).

2.2 Certificateless Hybrid Signcryption

Dent *et al.* proposed the first hybrid signcryption scheme with insider and outsider security (Dent, 2005a; Dent, 2005b). Following that, Li *et al.* proposed the first certificateless hybrid signcryption scheme (Li et al., 2009). Wu *et al.* proposed a certificateless hybrid signcryption scheme for IoT (Wu et al., 2022). The scheme utilizes PKC to generate a symmetric key and is used to signcrypt the message. While the scheme provides confidentiality, authentication, forward security, and public verification under CDH and Decisional Bilinear Diffie-

Hellman (DBDH) assumptions, it incurs high computational cost due to Bilinear Pairing (BP) operation. Yin *et al.* proposed a certificateless hybrid signcryption scheme for wireless sensor network (WSN) (Yin and Liang, 2015). Similarly, Gong *et al.* presented a lightweight and secure certificateless hybrid signcryption scheme for the IoT (Gong et al., 2022). It ensures data confidentiality, integrity, and authenticity. The scheme utilizes BP for initialization and key construction and proves security under CDH and DBDH assumptions. Hongzhen *et al.* presented certificateless signcryption scheme for Vehicular Ad hoc Networks (VANETs) without BP (Hongzhen et al., 2021). Moreover, Zhang *et al.* introduced a certificateless hybrid signcryption scheme suitable for the IoT (Zhang et al., 2022). The scheme is constructed to achieve both confidentiality and unforgeability under DL, CDH, DBDH, and BDH assumptions. In 2017, Niu *et al.* proposed a heterogeneous hybrid signcryption for multi-message and multi-receiver (Niu et al., 2017). 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 Computational BDH. In 2022, Niu *et al.* presented a broadcast signcryption scheme based on certificateless cryptography for WSN (Niu et al., 2022b). The scheme aims to ensure the confidentiality and integrity of the data transmitted, while protecting by the privacy of the receiver's ID 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.

3 PRELIMINARIES AND ASSUMPTIONS

1. **Elliptic Curve Based Computational Diffie-Hellman (ECCDH) Assumption.** The security assumption of ECCDH is according to (Cohen et al., 2005).

Definition 3.1. 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 .

2. **Elliptic Curve Discrete Logarithm (ECDL) Assumption.** The security assumption of ECDL

is adopted from (Cohen et al., 2005).

Definition 3.2. 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 $Q = xP$.

3. **The Multi-Recipient Key Encapsulation Mechanism (mKEM) and Data Encapsulation Mechanism (DEM).** The notion of mKEM was first proposed by N.P Smart (Smart, 2004) and has a KEM like construction which takes multiple receiver's public keys pk_{r_i} where $1 \leq i \leq t$ and $t < n$ as input and generates a single symmetric session key K and an encapsulation C of K .

Definition 3.3. The mKEM construction below is according to the (Smart, 2004):

- (a) mKEM. It consists of four algorithms defined as follows (Setup, KeyGen, mKEM.Encaps, mKEM.Decaps):
- Setup. On input the security parameter 1^λ , the algorithm outputs PP.
 - KeyGen. Taking PP as input, the algorithm outputs (pk, sk) for each user.
 - mKEM.Encaps. On input PP and a set of receiver public keys pk_{r_i} where $1 \leq i \leq t$, 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 sk_{r_i} , and an encapsulation C_1 as input, this algorithm outputs K . The correctness holds if $K = \text{mKEM.Decaps}(PP, sk_{r_i}, C_1)$.
- (b) DEM: It consists of two algorithms ($\text{Enc}_K, \text{Dec}_K$) (Niu et al., 2017) defined as follows:
- Enc_K . On input (K, m) , this algorithm outputs a ciphertext C_2 .
 - Dec_K . Taking (K, C_2) as input, this algorithm outputs m' . The correctness of DEM holds if $m' = m$.

4. KEM-DEM Hybrid Signcryption Scheme.

Definition 3.4. The construction of KEM-DEM hybrid signcryption scheme is given by (Dent, 2005a). It consists of four algorithms (Setup, KeyGen, Gen – Enc, Dec – Ver) defined as follows:

- (a) Setup. It takes as input a security parameter 1^λ and outputs PP.
- (b) KeyGen. Taking PP as input, this algorithm outputs a public and private key pair for sender (pk_s, sk_s) and receiver (pk_r, sk_r) .
- (c) Gen – Enc. In Gen – Enc, the sender runs following algorithms:

- Encaps. On input (PP, sk_s, pk_r, 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 ciphertext $CT = (C_1, C_2)$.
- (d) Dec – Ver. In this phase, the receiver runs following algorithms:
- Decaps. On input (sk_r, C_1) , it outputs K . If $K = \perp$, the sender stops. Otherwise, the receiver runs next algorithmic step.
 - Dec_K . On input (C_2, K) , outputs m . If $m = \perp$, the receiver stops. Else, it runs next step.
 - Ver. Taking (pk_s, m, C_1) as input, it outputs either valid or not. If valid, outputs m , else $\perp$.

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 $n = (PID_s, \{PID_1, \dots, PID_{r_i}, \dots, PID_{r_t}\})$. Assume, a sender with 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 a trusted authority that 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 generates its private key sk_{RA} and public key pk_{RA} . RA is also responsible for user registration, ID verification, and PID assigning.
- **Sender.** The sender with PID_s encrypts a m using the set of designated receiver's public key pk_{r_i} , signs with its private key 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 pk_s .

4.2 Definition of AMCLHS

The AMCLHS scheme represents a hybrid approach, leveraging both mKEM and DEM components. Before signcryption the message, RA verifies user's real identity ID_R , registers, and assigns a PID to each corresponding user. 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 verifies in the unsigncryption phase as given in Def. 4.1. 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 $CT = (C_1, C_2)$. 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 4.1. *In the AMCLHS scheme, the sender with 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 Defs. 3.3 and 3.4. The proposed scheme consists of eight polynomial time algorithms as follows:*

1. **Setup.** On input the security parameter 1^λ , the KGC generates (PP, msk, mpk) . Next, RA generates sk_{RA} and pk_{RA} .
2. **Pseudo-Identity.** Takes Real Identity (ID_R) and pk_{RA} as input and outputs a PID.
3. **Partial Private Key.** For each PID, the KGC takes (mpk, msk) as input, it outputs the partial private key (ppk) .
4. **Secret Value.** On input the PID, each user generates a secret value (sv) .
5. **Private Key.** Taking (ppk, sv) as input, each user generates the sk .
6. **Public Key.** On input the sv , each user outputs the pk .
7. **Signcryption.** To signcrypt the m and generate the CT, the sender runs this algorithm in two phases. In Phase 1, the sender runs mKEM.Encaps and in Phase 2, the sender runs Enc $_K$ according to the Def. 3.3. The phases are defined as follows:
 - Phase 1 (mKEM.Encaps) Taking PP , sk_s , a plaintext m and a set pk_{r_i} for $1 \leq i \leq t$, this algorithm outputs C_1 and K
 - Phase 2 (Enc $_K$) On input (K, m) , this algorithm outputs C_2 and sets signcrypted ciphertext $CT = (C_1, C_2)$.

8. **Unsigncryption.** To unsigncrypt the CT and retrieve m , the receiver runs this algorithm in three phases. Phase 1 consists of mKEM.Decaps, Phase 2 consists of Dec $_K$, and Phase 3 consists of Ver algorithm according to the Def. 3.3.

- Phase 1 (mKEM.Decaps). Taking (sk_{r_i}, C_1) as input, this algorithm outputs K .
- Phase 2 (Dec $_K$). On input (K, C_2) , this algorithm outputs m' . If $m' \neq m$, the receiver rejects the m . If $m' = m$, the receiver verifies the signature in Phase 3.
- Phase 3 (Ver). Taking (m', C_1, pk_s) as input, this algorithm verifies the signature S . If it is valid, accept the m , or else return $\perp$.

5 SECURITY MODEL

We define the security Game-I for *ind-cca2-I* and *ind-cca2-II* in Sec. 5.1, to evaluate the security against Type-I adversary ($\mathcal{A}_I$) and Type-II adversary ($\mathcal{A}_{II}$), respectively. Moreover, in Sec. 5.2, we introduce the security Game-II for *euf-cma-I* and *euf-cma-II* to evaluate the security against $\mathcal{A}_I$ and $\mathcal{A}_{II}$ and are defined as follows:

1. $\mathcal{A}_I$. $\mathcal{A}_I$ is an honest-but-curious user who cannot access msk but can replace the pk of any ID with the value of his/her own choice. $\mathcal{A}_I$ is not allowed to ask a ppk query q_{ppk} for any of the target ID.
2. $\mathcal{A}_{II}$. $\mathcal{A}_{II}$, also known as malicious KGC, cannot make public key replace query q_{pr} for the target ID. $\mathcal{A}_{II}$ 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 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 C . 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 $\beta = \beta'$.

Definition 5.1. *The ind-cca2 requires that there exists no PPT Adversary $\mathcal{A}$ which could distinguish ciphertexts. 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 C keeps a list L_l of q_{H_l} to record the responses.

- **Setup.** C generates $(PP, msk, mpk, sk_{RA}, pk_{RA})$ and passes to $\mathcal{A}$. Then $\mathcal{A}$ selects t target PID_{r_i} where $1 \leq i \leq t$.
- 2. **Phase-2.** The $\mathcal{A}$ proceeds to make a series of queries, subject to the restrictions defined in Sec. 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} , sign-encryption query q_{sc} , and unsign-encryption query q_{usc} . An initially empty list L_{pk} is maintained by the C to store public and secret information. The C responds to each query as follows:
 - q_{pk} . Upon receiving such query for PID , the C searches L_{pk} for pk . If it does not exist, C runs the secret value algorithm to generate a sv for PID , and performs the public key algorithm to return the pk to $\mathcal{A}$.
 - q_{ppk} . Given PID , the C checks if $PID = PID^*$. If it does, the C aborts. Otherwise, it fetches the ppk from L_{pk} . If it does not exist, C runs a partial private key algorithm to return ppk and updates L_{pk} .
 - q_{sv} . Upon q_{sv} , the C checks L_{pk} for sv . If it does not exist, C runs a secret value algorithm and returns sv to $\mathcal{A}$.
 - q_{pr} . Given PID as input, the C replaces pk with pk' and updates L_{pk} .
 - q_{sc} . On input (m, PID_s, PID_{r_i}) , the C checks if $PID_{r_i} = PID^*$. If it is not, C performs normal sign-encryption operation by taking values from L_{pk} . Otherwise, it performs the sign-encryption algorithm to generate CT .
 - q_{usc} . Upon receiving (CT, PID_s, PID_{r_i}) as input, the C checks if $PID_{r_i} = PID^*$. If it is not, C performs normal unsign-encryption operation. Otherwise, C performs the unsign-encryption algorithm to answer m .
- 3. **Challenge.** The $\mathcal{A}$ outputs a target plaintext (m_0, m_1) . The 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 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 $\mathcal{A}_I$ is defined as:

$$Adv_{\mathcal{A}_I}^{IND-CCA2} = |\Pr[\beta = \beta'] - 1/2| \quad (1)$$

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

$$Adv_{\mathcal{A}_{II}}^{IND-CCA2} = |\Pr[\beta = \beta'] - 1/2| \quad (2)$$

5.2 Game-II

Game-II consists of two phases. In each phase, the $\mathcal{A}$ asks a polynomially bounded number of hash, pk and sk queries. In the end, $\mathcal{A}$ outputs the forged ciphertext. $\mathcal{A}$ wins if unsign-encryption does not return $\perp$.

Definition 5.2. An AMCLHS scheme is *uf-cma secure* if every PPT $\mathcal{A}$ has a negligible advantage in winning the Game-II.

1. **Phase-1.** Phase-1 remains same as in Def. 5.1 except that $\mathcal{A}$ selects a target identity as PID_s^* .
2. **Phase-2.** The $\mathcal{A}$ asks a number of queries with the restrictions defined in Sec. 5. The queries include q_{pk} , q_{ppk} , q_{pr} , q_{sv} , q_{sc} , and q_{usc} and are defined in Phase-2 in Def. 5.1. C maintains an initially empty list L_{pk} to store the pk and sk information.
3. **Forgery.** $\mathcal{A}$ outputs the forged CT under a targeted PID_s^* . $\mathcal{A}$ wins if unsign-encryption does not return $\perp$.

6 THE PROPOSED AMCLHS SCHEME

In this section, we focus on the construction of the proposed AMCLHS scheme, built upon the mKEM-DEM framework, according to the Def. 4.1. The structure of the scheme is shown in Fig. 1.

1. **Setup.** The KGC initializes the system by taking the security parameter 1^λ as input. It chooses a group $\mathbb{G}$ of large prime order q , 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 is $H_1 : \{0, 1\}^* \times \mathbb{G} \rightarrow \mathbb{G}$. The third hash function is $H_2 : \mathbb{G} \rightarrow \{0, 1\}^k$, where k denotes the plaintext box length. The fourth is $H_3 : \{0, 1\}^* \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$. The KGC generates $PP = \{\mathbb{G}, E, P, q, H_0, H_1, H_2, H_3\}$, randomly selects $x_0 \in \mathbb{Z}_q^*$ as the master secret key msk , and calculates the master public key $mpk = x_0 P$. It then publishes PP and mpk , while keeping msk secret. Subsequently, the RA selects $v \in \mathbb{Z}_q^*$ at random as its secret key sk_{RA} and calculates its public key $pk_{RA} = vP$. The RA publicizes pk_{RA} and keeps sk_{RA} as a secret.
2. **Pseudo-Identity.** This algorithm is run by each user and RA as follows:
 - User. Each user chooses a random $ID_R \in \{0, 1\}^\ell$ and computes $R = \alpha P$ where $\alpha \in \mathbb{Z}_q^*$. Tak-

- ing (ID_R, α) as input, it computes $PID = ID_R \oplus H_0(\alpha pk_{RA})$ and sends (PID, R) to RA.
- **RA.** On input (PID, R) , RA verifies $ID_R = PID \oplus H_0(Rv)$. If it holds, RA accepts the registration request, confirms and assigns $PID = ID_R \oplus H_0(\alpha pk_{RA})$ to each corresponding user ID. Else, RA discards the PID and cancels the registration request.
3. **Partial Private Key.** Taking (PID, mpk, msk) as input, the KGC computes $Q_{PID} = H_1(PID || mpk)$ as a public component. Taking Q_{PID} , the KGC computes ppk as $d = x_0 Q_{PID}$.
 4. **Secret Value.** Each user with PID chooses a random $x \in \mathbb{Z}_q^*$ randomly as a sv.
 5. **Private Key.** On input (d, x) , each user with PID sets $sk = (d, x)$.
 6. **Public Key.** Taking x as input, each user with PID computes $pk = xP$.
 7. **Signcryption.** The sender with PID_s and sk_s runs following phases to signcrypt a m and sends CT to receivers with PID_{r_i} and pk_{r_i} $1 \leq i \leq t$:
 - Phase 1 (mKEM-Encaps).
 - (a) Randomly chooses $r \in \mathbb{Z}_q^*$ and computes $U = rP$.
 - (b) Taking pk_{r_i} and $Q_{PID_{r_i}}$ as input, computes $Z_{1i} = d_s Q_{PID_{r_i}}$ and $Z_{2i} = x_s pk_{r_i}$.
 - (c) Computes $\psi_i = (Z_{1i}, Z_{2i})$ and $K = H_2(\psi_i)$.
 - (d) Computes a hash $f_i = H_3(m, \psi_i, pk_s, pk_{r_i})$ and Signature $S_i = r^{-1}(f_i || w d_s x_s)$ where $w = x_U \bmod q$ which is the x-coordinate of U .
 - (e) Sets $C_{1i} = (f_i, S_i)$ and outputs (C_{1i}, K) .
 - Phase 2 (Enc_K).
 - (a) Computes $C_{2i} = Enc_K(m)$. Sets $CT_i = (C_{1i}, C_{2i})$ and sends to t designated receivers.
 8. **Unsigncryption.** Each designated receiver with PID_{r_i} takes (sk_{r_i}, pk_s) as input for i^{th} receiver and runs the following phases to unsigncrypt the CT_i and generate m :
 - Phase 1 (mKEM-Decaps).
 - (a) Taking (x_{r_i}, d_{r_i}) as input, computes $Z'_{1i} = d_{r_i} Q_{PID_s}$ and $Z'_{2i} = pk_s x_{r_i}$.
 - (b) Computes $\psi'_i = (Z'_{1i}, Z'_{2i})$ and $K = H_2(\psi'_i)$. If $K = \perp$, the receiver aborts else, decrypts m as follows:
 - (a) Phase 2 (Dec_K).
 - (a) Calculates $m' = Dec_K(C_{2i})$. If $m' = m$, verifies the S_i else rejects.
 - (b) Phase 3 (Ver).
 - (a) Inputs (C_{1i}, pk_s) , outputs $f'_i = H_3(m', \psi'_i, pk_s, pk_{r_i})$.

- (b) If $f'_i = f_i$, verifies S_i by checking if $U = rP$ and $w' = x_U \bmod q$. If $w' = w$, the receiver will accept the signcrypted m else returns $\perp$ and aborts.

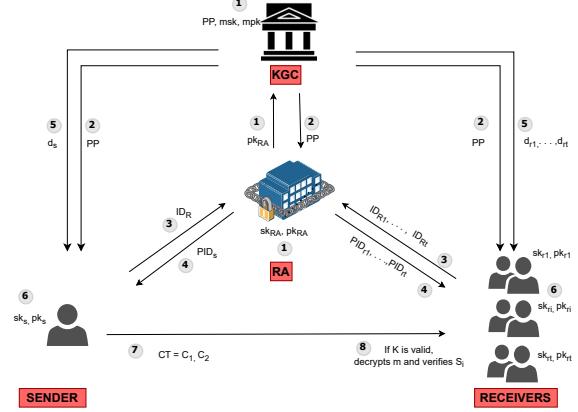


Figure 1: Our Proposed AMCLHS Scheme.

7 SECURITY ANALYSIS

The security analysis of the proposed AMCLHS scheme is based on the security model defined in Sec. 5. The message confidentiality is based on Theorems 7.1 and 7.2 which demonstrates that the scheme is secure against $ind\text{-}cca2$ $\mathcal{A}_I$ and $\mathcal{A}_{II}$ in Def. 5.1. Similarly, unforgeability is based on Theorems 7.3 and 7.4 and follows that the scheme is secure against $euf\text{-}cma$ $\mathcal{A}_I$ and $\mathcal{A}_{II}$ in Def. 5.2.

Confidentiality

Theorem 7.1. Suppose that the $ind\text{-}cca2\text{-I}$ has a non-negligible advantage ϵ in winning the game then, there is 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 assumption, the C has to compute xyP as Def. 3.1 by interacting with the $\mathcal{A}_I$ as follows: $\square$

1. **Phase-1.** A polynomially bounded number of queries q are made by an $\mathcal{A}_I$. The C keeps a list L_I of qH_I to record the responses.
 - **Setup.** The C runs this algorithm to generate $PP = \{\mathbb{G}, E, P, q, H_0, H_1, H_2, H_3\}$. The C sets new value for the $mpk = \theta P$ and sends PP and mpk to the $\mathcal{A}_I$. The $\mathcal{A}_I$ selects t target identities denoted by PID_i^* where $1 \leq i \leq t$.
 - **H_1 -Query.** Upon H_1 query, C determines if the tuple (Q_{PID_i}, mpk, PID_i) exists in L_I or not. If it already exists, C returns Q_{PID_i} to $\mathcal{A}_I$. Else, if $PID_i \neq PID_i^*$, C sets $Q_{PID_i} = H_1(PID_i || mpk)$.

- If $PID_i = PID_i^*$, C chooses $\gamma \in \mathbb{Z}_q^*$, computes $Q_{PID_i} = \gamma P$, adds in L_1 and sends Q_{PID_i} to $\mathcal{A}_1$.
- **H_2, H_3 -Query.** Upon H_2 and H_3 queries, the C determines if the tuple $(K, \psi_i, Z_{1_i}, Z_{2_i})$, (m, ψ_i, f_i) exists in L_2 and L_3 . If it does, C returns K and f_i to $\mathcal{A}_1$. Else, the C chooses $K \in \{0, 1\}^k$ and $f_i \in \mathbb{Z}_q^*$ randomly, updates both tuples. The C sends ψ_i and f_i to $\mathcal{A}_1$.
2. **Phase-2.** The $\mathcal{A}_1$ asks queries including q_{pk} , q_{ppk} , q_{pr} , q_{sv} , and q_{usc} . The C maintains an initially empty list L_{pk} to store the public and secret information. The C responds to the queries as follows:
- q_{pk} . Upon pk_i query for PID_i , C checks if pk_i exists in L_{pk} . If it exists, C returns pk_i to $\mathcal{A}_1$. Otherwise, C chooses $x_i \in \mathbb{Z}_q^*$ and computes $pk_i = x_i P$, adds the tuple $(PID_i, -, pk_i, x_i)$ in L_{pk} and returns pk_i to $\mathcal{A}_1$.
 - q_{ppk} . Upon q_{ppk} , if $PID_i = PID_i^*$, the C aborts. Otherwise, if it exists in L_{pk} , C sends d_i to $\mathcal{A}_1$, if it does not, C randomly chooses $Q_{PID_i} = \gamma P$ from L_1 and return $d_i = \text{mpk} Q_{PID_i}$ to $\mathcal{A}_1$. The C then updates the tuple (PID_i, d_i, pk_i, x_i) in L_{pk} .
 - q_{sv} . Upon receiving q_{sv} , C checks if it exists in L_{pk} , if it does, C sends x_i to $\mathcal{A}_1$. If not, C performs the q_{pk} and return x_i to $\mathcal{A}_1$.
 - q_{pr} . Upon q_{pr} , C replaces the pk_i with pk'_i for PID_i and updates the tuple $(PID_i, d_i, pk'_i, -)$.
 - q_{sc} . Upon receiving the query with PID_s , PID_{r_i} and m , the C checks if $PID_{r_i} = PID_i^*$. If not, the C performs the normal signcryption operation by taking values from L_{pk} . Otherwise, the C performs the signcryption as follows:
 - If pk_i is replaced, $\mathcal{A}_1$ provides another value.
 - Chooses $r \in \mathbb{Z}_q^*$ and computes $U = rP$.
 - Gets $Q_{PID_{r_i}}$ from L_1 and computes $Z_{1_i} = d_s Q_{PID_{r_i}}$, $Z_{2_i} = x_s pk_{r_i}$, $\psi_i = (Z_{1_i} Z_{2_i})$, $K = H_2(\psi_i)$, and updates L_2 .
 - Computes $f_i = H_3(m, \psi_i, pk_s, pk_{r_i})$ and updates L_3 .
 - Computes S_i , $C_{1_i} = (f_i, S_i)$, $C_{2_i} = \text{Enc}_K(m)$ and returns $CT_i = (C_{1_i}, C_{2_i})$ to $\mathcal{A}_1$.
 - q_{usc} . Upon q_{usc} with PID_s , PID_{r_i} and a CT , the C checks whether $PID_{r_i} = PID_i^*$. If not, the C performs the normal unsigncryption operation. Otherwise, the C unsigncrypts m as follows:
 - If pk_i is replaced, $\mathcal{A}_1$ provides another value.
 - Searches the lists L_2 and L_3 for $(K, \psi'_i, Z'_{1_i}, Z'_{2_i})$ and $H_3(m, \psi'_i, f'_i)$.
 - If the record does not exist, C returns "failure". If it exists, the C computes $K \neq \perp$ and $m' = \text{Dec}_K(C_{2_i})$.
 - Checks if $f'_i = f_i$, if it holds then checks if $U = rP$ and $w' = x_U \bmod q$ holds or not. If yes, the C answers m else, returns $\perp$.
3. **Challenge.** The $\mathcal{A}_1$ chooses equal length plaintext message pair (m_0, m_1) and sends the target plaintext to the C . The $\mathcal{A}_1$ takes a sender PID_s and a target PID_{r_i} . Moreover, the $\mathcal{A}_1$ can not ask for the sk of the target PID_{r_i} . If $PID_{r_i} \neq PID_i^*$, the C returns $\perp$. Otherwise, it chooses $\beta \in \{0, 1\}^*$ and performs the following steps to generate a CT_i^* :
- Chooses $r^* \in \mathbb{Z}_q^*$ and computes $U^* = r^* P$
 - Computes $Z_{1_i}^* = d_s Q_{PID_{r_i}}$, $Z_{2_i}^* = x_s pk_{r_i}$, and $\psi_i^* = (Z_{1_i}^* Z_{2_i}^*)$. Computes $K^* = H_2(\psi_i^*)$.
 - $f_i^* = H_3(m, \psi_i^*, pk_s, pk_{r_i})$. Computes $S_i^* = r^{*-1} (f_i^* || w d_s x_s)$ and $C_{1_i}^* = (f_i^*, S_i^*)$.
 - $C_{2_i}^* = \text{Enc}_{K^*}(m)$ and computes $CT_i^* = (C_{1_i}^*, C_{2_i}^*)$.
4. **Phase-3.** $\mathcal{A}_1$ may issue further polynomially bounded queries as in *Phase-1*, however, $\mathcal{A}_1$ can not send the q_{ppk} of the target PID_{r_i} , or the unsigncryption query for CT_i^* .
5. **Guess.** The $\mathcal{A}_1$ will respond with the guess bit $\beta \in \{0, 1\}^*$. $\mathcal{A}_1$ wins the game if $\beta' = \beta$. The C will win the game by evaluating $\frac{\theta Z_{1_i} - d_i r}{(d_s - U)} = \theta \gamma P$ using $\text{mpk} = \theta P$, $Q_{PID_i} = \gamma P$ which is the solution to the ECCDH assumption.
- Next, we evaluate the advantage of C winning the Game-I (*ind-cca2-I*) by calculating the probability of aborting the game during the following events:
1. In q_{ppk} , the game aborts for $PID_i = PID_i^*$. The probability is $\Pr(E_{q_{ppk}}) = 1/q_{ppk}$.
 2. In q_{usc} , the game aborts due to invalid m . The probability is $\Pr(E_{q_{usc}}) = q_{usc}/2^k$.
 3. In the challenge phase, C aborts if the adversary queries against the identity $PID_{r_i} \neq PID_i^*$. The probability is $\Pr(E_{q_{H_1}}) = (1 - 1/q_{H_1})$.
- Moreover, the C fetches L_1 to retrieve Q_{PID_i} and L_2 to retrieve Z_{1_i} and evaluates $\frac{\theta Z_{1_i} - d_i r}{(d_s - U)} = \theta \gamma P$ with probability $(1/q_{H_1} + 1/q_{H_2})$. Therefore, the probability of the C winning the game with advantage ϵ' is:
- $$\epsilon' \geq \epsilon \left(\frac{1}{q_{H_1}} + \frac{1}{q_{H_2}} \right) \left(\frac{1}{q_{H_1}} \right) \left(1 - \frac{1}{q_{ppk}} \right) \left(1 - \frac{q_{usc}}{2^k} \right) \quad (3)$$
- Theorem 7.2.** Suppose that the *ind-cca2-II* has a non-negligible advantage ϵ in winning the game then, there is a C that can solve the ECCDH assumption with the non-negligible advantage ϵ' .

Proof. Given a random instance $(P, xP, yP) \in \mathbb{G}$ of the ECCDH assumption, the C has to compute xyP as Def. 3.1 by interacting with $\mathcal{A}_{II}$ as follows: $\square$

1. **Phase-1.** A polynomially bounded number of queries q are made by an $\mathcal{A}_{II}$.
 - **Setup.** The C generates PP and mpk as in Theorem 7.1 and sends PP and mpk to the $\mathcal{A}_{II}$.
 - **H_1 -Query.** The response of H_1 query to $\mathcal{A}_{II}$ is same as in Theorem 7.1.
 - **H_2, H_3 -Query.** This query remains the same as in Theorem 7.1 and the C sends ψ_i and f_i to $\mathcal{A}_{II}$.
2. **Phase-2.** $\mathcal{A}_{II}$ asks a number of queries including q_{pk}, q_{sv} . The C responds as follows:
 - q_{pk} . Upon pk_i for PID_i , the C checks if pk_i exists in the L_{pk} as (PID_i, d_i, pk_i, x_i) . If it exists, C returns pk_i to C . Otherwise, C chooses $x_i \in \mathbb{Z}_q^*$, $pk_i = x_i P$, adds the tuple $(PID_i, -, pk_i, x_i)$ in L_{pk} and returns pk_i to $\mathcal{A}_{II}$.
 - q_{sv} . Upon q_{sv} , C checks if $PID_i = PID_i^*$. If it holds, the C aborts because, in this case, the PID_i is a target ID. Otherwise, it checks if x_i already exists in the L_{pk} . If it exists, the C returns x_i to $\mathcal{A}_{II}$. Otherwise, C runs q_{pk} , computes $pk_i = x_i P$, adds the tuple (PID_i, d_i, pk_i, x_i) in L_{pk} and returns x_i to $\mathcal{A}_{II}$.
 - q_{sc} . The q_{sc} query from $\mathcal{A}_{II}$ and the response from C will be similar to the Theorem 7.1.
 - q_{usc} . The q_{usc} query from $\mathcal{A}_{II}$ and the response from C will be same as in Theorem 7.1.
3. **Challenge.** The $\mathcal{A}_{II}$ chooses target plaintext m_0, m_1 and sends to the C . $\mathcal{A}_{II}$ takes a PID_s and a target PID_{r_i} . Moreover, the $\mathcal{A}_{II}$ can not ask for the sk of the receiver PID_{r_i} . If $PID_{r_i} \neq PID_{r_i}^*$, the C returns $\perp$. Otherwise, it chooses $\beta \in \{0, 1\}^*$ and performs the following steps to generate a challenge CT^* :
 - Chooses $r^* \in \mathbb{Z}_q^*$ and computes $U^* = r^* P$.
 - Computes $Z_{1_i} = d_s Q_{PID_{r_i}}, Z_{2_i} = x_s pk_{r_i}$, and $\psi_i^* = (Z_{1_i}^*, Z_{2_i}^*)$. Computes $K^* = H_2(\psi_i^*)$.
 - Computes $f_i^* = H_3(m, \psi_i^*, pk_s, pk_{r_i})$. Computes $S_i^* = r^{*-1}(f_i^* || wd_s x_s)$ and $C_{1_i}^* = (f_i^*, S_i^*)$.
 - $C_{2_i}^* = \text{Enc}_K^*(m)$ and $CT_i^* = (C_{1_i}^*, C_{2_i}^*)$.
4. **Phase-3.** The $\mathcal{A}_{II}$ may issue further queries as in Phase-1 however, $\mathcal{A}_{II}$ cannot send the q_{sv} for the target $PID_{r_i}^*$ and the q_{uns} for CT_i^* .
5. **Guess.** The $\mathcal{A}_{II}$ will respond with the guess bit $\beta' \in \{0, 1\}^*$. The adversary wins the game if $\beta' = \beta$. The C will win the game by obtaining

$\theta \gamma P$, which is the solution to the ECCDH assumption. The C obtains it by evaluating $\frac{\theta Z_{1_i} - d_i r}{(d_s - U)} = \theta \gamma P$ since $mpk = \theta P, Q_{PID_i} = \gamma P$.

Next, we will analyse the advantage of the C in winning the game. The C advantage is based on the occurrence of the events in which the game aborts. The C aborts under the following events:

- The q_{sv} where the game aborts for $PID_i = PID_i^*$. The probability is $\Pr(E_{q_{sv}}) = 1/q_{sv}$.
- The q_{usc} where the game aborts due to invalid m . The probability is $\Pr(E_{q_{usc}}) = q_{usc}/2^k$.
- In the challenge phase, $\mathcal{A}_{II}$ queries for $PID_{r_i}^* \neq PID_{r_i}^*$. The probability is $\Pr(E_{q_{H_1}}) = (1 - 1/q_{H_1})$.

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

$$\epsilon' \geq \epsilon \left(\frac{1}{q_{H_1}} + \frac{1}{q_{H_2}} \right) \left(\frac{1}{q_{H_1}} \right) \left(1 - \frac{1}{q_{sv}} \right) \left(1 - \frac{q_{usc}}{2^k} \right) \quad (4)$$

Unforgeability

Theorem 7.3. Suppose that the euf-cma-I adversary $\mathcal{A}_I$ has a non-negligible advantage ϵ in winning the game then, there is C that can solve the ECDL with the non-negligible advantage ϵ' .

Proof. Given a generator point $P \in \mathbb{G}$ and a new generator $Q = \phi P$ in the same group, the C has to find ϕ by interacting with $\mathcal{A}_I$. $\square$

1. **Phase-1.** $\mathcal{A}_I$ makes same queries as in the Phase-1 of Theorem 7.1.
 - **Setup.** The C runs setup algorithm to generate $PP = \{\mathbb{G}, E, P, q, H_0, H_1, H_2, H_3\}$. The C sets $mpk = \theta P$ and sends PP and mpk to the $\mathcal{A}_I$. The $\mathcal{A}_I$ selects a target identity as PID_s^* .
2. **Phase-2.** The queries and responses are the same as in Phase-2 of Theorem 7.1 except the response to q_{ppk} is as follows:
 - q_{ppk} . Upon q_{ppk} , if $PID = PID_s^*$, the C aborts. Otherwise, if it exists in L_{pk} , the C sends d_i to $\mathcal{A}_I$, if it does not, the C randomly chooses $\phi \in \mathbb{Z}_q^*$ and computes $d_i = \phi Q_{PID_i}$. The C returns $d_i = \phi Q_{PID_i}$ to $\mathcal{A}_I$ and updates L_{pk} .
3. **Forgery.** Taking the targets PID_s^* and PID_{r_i} , $\mathcal{A}_I$ outputs a forged $CT_i^* = (C_{1_i}^*, C_{2_i}^*)$ on m^* where $C_{1_i}^* = (f_i^*, S_i^*)$ and $C_{2_i}^* = \text{Enc}_K^*(m)$ which is the valid signcrypted ciphertext and is not the result of signcryption oracle.

- Case-1 ($PID \neq PID_s^*$): The $\mathcal{C}$ returns $\perp$.
- Case-2 ($PID = PID_s^*$): The $\mathcal{C}$ extracts the L_{pk} for the record $(PID_i^*, d_i^*, pk_i^*, x_i^*)$ and L_3 for the record (m^*, ψ_i^*, f_i^*) .

According to Forking Lemma, $\mathcal{C}$ replays the $\mathcal{A}_I$ with the same random tape but distinct attributes from H_1 and H_3 . It implies that, $h_1^* = H_1(\text{mpk}, PID_i^*)$ and $h_1'^* = H_1(\text{mpk}, PID_i^*)$, and $h_1^* \neq h_1'^*$ i.e. $Q_{PID_s}^* \neq Q_{PID_s}'^*$. Similarly, $h_3^* = H_3(m^*, \psi_i^*, pk_s^*, pk_{r_i}^*)$, $h_3'^* = H_3(m^*, \psi_i^*, pk_s^*, pk_{r_i}^*)$ and $h_3^* \neq h_3'^*$ i.e. $f_i^* \neq f_i'^*$. Finally, the $\mathcal{A}_I$ outputs another forged $CT_i'^* = (C_{1_i}^*, C_{2_i}^*)$ on the same m^* where $C_{1_i}^* = (f_i'^*, S_i'^*)$ and $C_{2_i}^* = \text{Enc}_K^*(m)$. Finally, $\mathcal{C}$ will have two valid signatures:

$$S_i^* = r^{*-1}(f_i^* \| wd_s^* x_s) \quad (5)$$

$$S_i'^* = r'^{-1}(f_i'^* \| wd_s'^* x_s) \quad (6)$$

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

$$\phi = r^{*-1}(f_i'^* - f_i^*) + (S_i^* - S_i'^*)(r^{*-1}(wd_s^* Q_{PID_s}^* - Q_{PID_s}'^*))^{-1}$$

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

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

Theorem 7.4. Suppose that the euf-cma-II adversary $\mathcal{A}_{II}$ 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 a $P \in \mathbb{G}$ and a new generator $Q = \pi P$ in the same group where $\pi \in \mathbb{Z}_q^*$. The $\mathcal{C}$ has to find π by interacting with the $\mathcal{A}_{II}$ such that $Q = \pi P$. $\square$

1. **Phase-1.** The queries are similar to Theorem 7.2.

- **Setup.** The $\mathcal{C}$ generates (PP, mpk) as in Theorem 7.3 and sends to the $\mathcal{A}_{II}$.

2. **Phase-2.** The queries and responses are same as in Phase-2 of Theorem 7.2, except the response to q_{sv} is as follows:

- q_{sv} . Upon q_{sv} , the $\mathcal{C}$ checks if $PID = PID_s^*$. If it holds, $\mathcal{C}$ aborts because, in this case, the PID is a target ID. Otherwise, it checks if x_i exists in L_{pk} . If it exists, the $\mathcal{C}$ returns x_i to $\mathcal{A}_{II}$. Otherwise, computes $pk_i = \pi P$ where $x_i = \pi \in \mathbb{Z}_q^*$ and adds in L_{pk} and returns x_i to $\mathcal{A}_{II}$.

3. **Forgery.** Taking the target PID_s^* and PID_{r_i} , $\mathcal{A}_{II}$ outputs a forged $CT_i^* = (C_{1_i}^*, C_{2_i}^*)$ on m^* where $C_{1_i}^* = (f_i^*, S_i^*)$ and $C_{2_i}^* = \text{Enc}_K^*(m)$ which is the valid signcrypted ciphertext and is not the result of signcryption oracle.

- Case-1 ($PID \neq PID_s^*$). The $\mathcal{C}$ returns $\perp$.
- Case-2 ($PID = PID_s^*$). The $\mathcal{C}$ extracts the L_{pk} for the record $(PID_i^*, d_i^*, pk_i^*, x_i^*)$ and L_3 for the record (m^*, ψ_i^*, f_i^*) .

According to the Forking Lemma, the $\mathcal{C}$ replays the $\mathcal{A}_{II}$ with the same random tape but distinct attributes from H_1 and H_3 . It implies that, $h_1^* = H_1(\text{mpk}, PID_i^*)$ i.e. $h_1'^* = H_1(\text{mpk}, PID_i^*)$ and $h_1^* \neq h_1'^*$ i.e. $Q_{PID_s}^* \neq Q_{PID_s}'^*$. Similarly, $h_3^* = H_3(m^*, \psi_i^*, pk_s^*, pk_{r_i}^*)$, $h_3'^* = H_3(m^*, \psi_i^*, pk_s^*, pk_{r_i}^*)$, and $h_3^* \neq h_3'^*$ i.e. $f_i^* \neq f_i'^*$. In the end, the $\mathcal{A}_{II}$ outputs another forged $CT_i'^* = (C_{1_i}^*, C_{2_i}^*)$ on the same m^* where $C_{1_i}^* = (f_i'^*, S_i'^*)$ and $C_{2_i}^* = \text{Enc}_K^*(m)$. Finally, $\mathcal{C}$ will have two valid signatures:

$$S_i^* = r^{*-1}(f_i^* \| wd_s^* x_s) \quad (8)$$

$$S_i'^* = r'^{-1}(f_i'^* \| wd_s'^* x_s) \quad (9)$$

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

$$\pi = r^{*-1}(f_i'^* - f_i^*) + (S_i^* - S_i'^*)(r^{*-1}(wmpk(Q_{PID_s}^* - Q_{PID_s}'^*)))^{-1}$$

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

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

Anonymity. Each user utilizes the PID to communicate instead of the ID_R where the sender sends same m to multiple receivers while ID_R of the receiver remains private from each other. The PID is assigned by RA after verifying each user's ID_R using its private key v . If ID_R is not verified, then the corresponding PID will be discarded. Additionally, since only RA knows its private key, no else could falsely verify the ID_R . In case of a dispute, RA can expose the ID_R .

Non-Repudiation. In our scheme, message m is signed by the sender with its sk_s as $S_i = r^{-1}(f_i \| wd_s x_s)$. The receiver verifies m using pk_s as $R_i = S_i^{-1}(f_i P \| wmpk Z_{1_i} Q_{PID_{r_i}}^{-1})$. Since the sender signs m with its sk_s that only the sender knows, it cannot deny sending m , thus proving non-repudiation.

Forward Security. In the proposed AMCLHS scheme, the symmetric session key K and its encapsulation C_{1_i} is generated using the (sk, pk) using a randomly generated secret value $x \in \mathbb{Z}_q^*$ and a ppk . In this case, even if the sk is exploited, the $\mathcal{A}$ cannot extract the past sessions since the x is randomly generated and updated for each session.

8 PERFORMANCE ANALYSIS

We compare the computational cost, communication cost, and security requirements of the AMCLHS scheme with existing multireceiver signcryption schemes. M shows point multiplication operation, E shows exponentiation in $\mathbb{Z}_q^*$, and n represents the number of users. The computational overhead is compared with (Niu et al., 2017; Niu et al., 2022b; Peng et al., 2020) as shown in Table 1. The overhead for signcryption is calculated for multiple recipients as outlined in our scheme, whereas the overhead for unsigncryption is determined on a per-receiver basis. Among the multireceiver signcryption schemes, Niu *et al.* have the highest computational overhead, utilizing a total of $(2n+4)BP + 1M + (2n+2)E$ operations (Niu et al., 2017). Peng *et al.* require total $(2n+5)M$ operations (Peng et al., 2020) and Niu *et al.* require a total of $(4n+6)M$ operations (Niu et al., 2022b). 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 and setting a new performance standard. Given its scalability and robustness, our scheme emerges as a compelling choice for larger, more complex broadcast communication scenarios, providing a significant upgrade over existing schemes.

Table 1: Comparison of Computational Overhead.

Schemes	Signcryption	Unsigncryption	Total
Niu <i>et al.</i> (Niu et al., 2017)	$(2n)BP + 1M + (2n)E$	$4BP + 2E$	$(2n+4)BP + 1M + (2n+2)E$
Peng <i>et al.</i> (Peng et al., 2020)	$(2n+1)M$	$4M$	$(2n+5)M$
Niu <i>et al.</i> (Niu et al., 2022b)	$(2n+4)M$	$(2n+2)M$	$(4n+6)M$
Our scheme	$(2n+2)M$	$3M$	$(2n+5)M$

Table 2: Comparison of Communication Cost.

Schemes	Ciphertext Length	Complexity of Communication	
		Signcryption	Unsigncryption
Niu <i>et al.</i> (Niu et al., 2017)	$n m + G + 2n G $	$O(n^2)$	$O(n)$
Peng <i>et al.</i> (Peng et al., 2020)	$n m + (n+2) \mathbb{Z}_q^* $	$O(n^2)$	$O(n)$
Niu <i>et al.</i> (Niu et al., 2022b)	$n(m+2) + 2 G + 2 \mathbb{Z}_q^* $	$O(n)$	$O(n)$
Our scheme	$n m + \mathbb{Z}_q^* + G + K $	$O(n)$	$O(1)$

The Table 2 shows the communication cost in terms of the size of the ciphertext generated by each scheme (Niu et al., 2017; Niu et al., 2022b; Peng et al., 2020). The proposed AMCLHS scheme has the optimal communication cost among the four schemes, as it only requires $n|m| + |\mathbb{Z}_q^*| + |G| + |K|$ bits to signcrypt a message. Moreover, our scheme has linear communication cost in signcryption while the un-

Table 3: Comparison based on Security Requirements.

Schemes	Confidentiality	Unforgeability	Anonymity	Non-repudiation	Forward Security
Niu <i>et al.</i> (Niu et al., 2017)	✓	✓	✓	✓	✓
Niu <i>et al.</i> (Niu et al., 2022b)	✓	✓	✓	×	×
Peng <i>et al.</i> (Peng et al., 2020)	✓	✓	✓	×	×
Our scheme	✓	✓	✓	✓	✓

signcryption cost remains constant. In Table 3, we present a comparative analysis of the security requirements between our scheme and existing multireceiver hybrid signcryption schemes (Niu et al., 2017; Niu et al., 2022b; Peng et al., 2020). The comparison parameters are 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 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 EC-CDH and ECDL 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

- Al-Riyami, S. S. and Paterson, K. G. (2003). Certificateless public key cryptography. In *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*, volume 2894 of LNCS, pages 452–473. Springer.
- Barbosa, M. and Farshim, P. (2008). Certificateless signcryption. In *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2008, Tokyo, Japan, March 18-20, 2008*, pages 369–372. ACM.
- Chen, X., He, D., Khan, M. K., Luo, M., and Peng, C. (2023). A secure certificateless signcryption scheme

- without pairing for internet of medical things. *IEEE Internet Things J.*, 10(10):9136–9147.
- Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., and Vercauteren, F. (2005). *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press.
- Cui, B., Lu, W., and He, W. (2022). A new certificateless signcryption scheme for securing internet of vehicles in the 5g era. *Security and Communication Networks*.
- Dent, A. W. (2005a). Hybrid signcryption schemes with insider security. In Boyd, C. and Nieto, J. M. G., editors, *Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings*, volume 3574 of *LNCS*, pages 253–266. Springer.
- Dent, A. W. (2005b). Hybrid signcryption schemes with outsider security. In Zhou, J., López, J., Deng, R. H., and Bao, F., editors, *Information Security, 8th International Conference, ISC 2005, Singapore, September 20-23, 2005, Proceedings*, volume 3650 of *LNCS*, pages 203–217. Springer.
- Gong, B., Wu, Y., Wang, Q., Ren, Y., and Guo, C. (2022). A secure and lightweight certificateless hybrid signcryption scheme for internet of things. *Future Gener. Comput. Syst.*, 127:23–30.
- Hongzhen, D., Qiaoyan, W., Shanshan, Z., and Mingchu, G. (2021). A pairing-free certificateless signcryption scheme for vehicular ad hoc networks. *Chinese Journal of Electronics*, 30(5):947–955.
- Kasyoka, P. N., Kimwele, M. W., and Mbandu, A. S. (2021). Efficient certificateless signcryption scheme for wireless sensor networks in ubiquitous healthcare systems. *Wirel. Pers. Commun.*, 118(4):3349–3366.
- Li, F., Shirase, M., and Takagi, T. (2009). Certificateless hybrid signcryption. In Bao, F., Li, H., and Wang, G., editors, *Information Security Practice and Experience, 5th International Conference, ISPEC, Xi'an, China, April 13-15, 2009, Proceedings*, volume 5451 of *LNCS*, pages 112–123. Springer.
- Li, X., Jiang, C., Du, D., Wang, S., Fei, M., and Wu, L. (2022). A novel efficient signcryption scheme for resource-constrained smart terminals in cyber-physical power systems. *CoRR*, abs/2212.04198.
- Malone-Lee, J. (2002). Identity-based signcryption. *IACR Cryptol. ePrint Arch.*, page 98.
- Niu, S., Niu, L., Yang, X., Wang, C., and Jia, X. (2017). Heterogeneous hybrid signcryption for multi-message and multi-receiver. *PloS one*, 12(9):e0184407.
- Niu, S., Shao, H., Hu, Y., Zhou, S., and Wang, C. (2022a). Privacy-preserving mutual heterogeneous signcryption schemes based on 5g network slicing. *IEEE Internet Things J.*, 9(19):19086–19100.
- Niu, S., Zhou, S., Fang, L., Hu, Y., and Wang, C. (2022b). Broadcast signcryption scheme based on certificateless in wireless sensor network. *Comput. Networks*, 211:108995.
- Peng, C., Chen, J., Obaidat, M. S., Vijayakumar, P., and He, D. (2020). Efficient and provably secure multireceiver signcryption scheme for multicast communication in edge computing. *IEEE Internet Things J.*, 7(7):6056–6068.
- Qiu, J., Fan, K., Zhang, K., Pan, Q., Li, H., and Yang, Y. (2019). An efficient multi-message and multi-receiver signcryption scheme for heterogeneous smart mobile iot. *IEEE Access*, 7:180205–180217.
- Selvi, S. S. D., Vivek, S. S., and Rangan, C. P. (2009). Certificateless KEM and hybrid signcryption schemes revisited. *IACR Cryptol. ePrint Arch.*, page 462.
- Selvi, S. S. D., Vivek, S. S., Shukla, D., and Rangan, C. P. (2008). Efficient and provably secure certificateless multi-receiver signcryption. In *Provable Security, Second International Conference, ProvSec 2008, Shanghai, China, October 30 - November 1, 2008. Proceedings*, volume 5324 of *LNCS*, pages 52–67. Springer.
- Smart, N. P. (2004). Efficient key encapsulation to multiple parties. In Blundo, C. and Cimato, S., editors, *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, volume 3352 of *LNCS*, pages 208–219. Springer.
- Wu, Y., Gong, B., Zhang, Y., et al. (2022). An improved efficient certificateless hybrid signcryption scheme for internet of things. *Wireless Communications and Mobile Computing*, 2022.
- Yang, Y., He, D., Vijayakumar, P., Gupta, B. B., and Xie, Q. (2022). An efficient identity-based aggregate signcryption scheme with blockchain for iot-enabled maritime transportation system. *IEEE Trans. Green Commun. Netw.*, 6(3):1520–1531.
- Yin, A. and Liang, H. (2015). Certificateless hybrid signcryption scheme for secure communication of wireless sensor networks. *Wirel. Pers. Commun.*, 80(3):1049–1062.
- Yu, X., Zhao, W., and Tang, D. (2022). Efficient and provably secure multi-receiver signcryption scheme using implicit certificate in edge computing. *J. Syst. Archit.*, 126:102457.
- Yu, Y., Yang, B., Huang, X., and Zhang, M. (2007). Efficient identity-based signcryption scheme for multiple receivers. In *Autonomic and Trusted Computing, 4th International Conference, ATC, Hong Kong, China, July 11-13, 2007, Proceedings*, volume 4610 of *LNCS*, pages 13–21. Springer.
- Zhang, W., Zhang, Y., Guo, C., An, Q., Guo, Y., Liu, X., Zhang, S., Huang, J., et al. (2022). Certificateless hybrid signcryption by a novel protocol applied to internet of things. *Computational Intelligence and Neuroscience*.
- Zheng, Y. (1997). Digital signcryption or how to achieve $\text{cost}(\text{signature}) + \text{cost}(\text{encryption})$. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *LNCS*, pages 165–179. Springer.
- Zhou, C. (2018). Certificateless signcryption scheme without random oracles. *Chinese Journal of Electronics*, 27:1002–1008.