A Multireceiver Certificateless Signcryption (MCLS) Scheme

Alia Umrani* and Paolo Palmieri

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

Abstract. User authentication and message confidentiality are the basic security requirements of high-end applications such as multicast communication and distributed systems. Several efficient signature-then-encrypt cryptographic schemes have been proposed to offer these security requirements with lower computational cost and communication overhead. However, signature-then-encryption techniques take more computation time than signcryption techniques. Signcryption accomplishes both digital signature and public key encryption functions in a single logical step and at a much lower cost than “signature followed by encryption.” Several signcryption schemes based on bilinear pairing operations have been proposed. Similarly, anonymous multi-receiver encryption has recently risen in prominence in multicast communication and distributed settings, where the same messages are sent to several receivers but the identity of each receiver should remain private. Anonymous multi-receiver encryption allows a receiver to obtain the plaintext by decrypting the ciphertext using their own private key, while their identity is kept secret to anyone, including other receivers. Among the Certificateless Multi-receiver Encryption (CLMRE) schemes that have been introduced, Hung et al. proposed an efficient Anonymous Multi-receiver Certificateless Encryption (AMCLE) scheme ensuring confidentiality and anonymity based on bilinear pairings and is secure against IND-CCA and ANON-CCA.

In this paper, we substantially extend Hung et al.’s multireceiver certificateless encryption scheme to a Multireceiver Certificateless Signcryption (MCLS) scheme that provides confidentiality along with authentication. We show that, as compared to Hung et al.’s encryption scheme, our signcryption scheme requires only three additional multiplication operations for signcryption and unsigncryption phases. Whereas, the signcryption cost is linear with the number of designated receivers while the unsigncryption cost remains constant for each designated receiver. We compare the results with other existing single receiver and multireceiver signcryption schemes in terms of number of operations, exemption of key escrow problem, and public key settings. The scheme proposed in this paper is more efficient for single and multireceiver signcryption schemes while providing exemption from the key escrow problem, and working in certificateless public key settings.

* This publication has emanated from research supported in part by a Grant from Science Foundation Ireland under Grant number 18/CRT/6222
Keywords: Authentication · Confidentiality · Certificateless · Public Key Cryptography · Multireceiver · Signcryption

1 Introduction

A message in digital communication must be secure in terms of confidentiality, authentication, and integrity. Encryption-based schemes are generally used for confidentiality, whereas digital signature-based schemes are used for authentication, integrity, and non-repudiation. As a result, digital signatures and public-key encryption are fundamental requirements for achieving security. However, signing and then encrypting a message has a high computational cost. Signcryption, on the other hand, not only signs the message as the traditional approach requires, but also encrypts it in a single step. This ensures that the message is meaningless to anyone but the intended recipient, who can also verify the sender’s identity and the message’s integrity. Signcryption is more attractive than the sign-then-encrypt procedure because it requires less computation time and has a lower message expansion rate. For typical security parameters in high level security applications, signcryption costs 50% less in computation time and 85% less in message expansion than signature followed by encryption [19]. Furthermore, Authenticated Encryption (AE) provides security against both Chosen Ciphertext Attack (CCA) and Chosen Plaintext Attack (CPA), and signcryption provides AE and thus CCA and CPA security.

Zheng et al. [18] proposed the first signcryption scheme, which combines digital signature and public key encryption to provide authentication, non-repudiation, and confidentiality at a lower cost than signing and encrypting operations separately. Malone-Lee [10] proposed the first identity-based signature scheme to provide public verifiability and forward security. Following that, several Identity (ID) - based encryption schemes were proposed. Chen et al. [3] and Chow et al. [4] proposed ID-based signcryption schemes respectively, to demonstrate public verifiability, forward security, ciphertext unlinkability, and anonymity. However, ID-based cryptography has an inherent key escrow problem in which a malicious Key Generation Center (KGC) compromises the entity’s private key. To solve the key escrow problem, Al-Riyami et al. proposed the concept of certificateless Public Key Cryptography (PKC) [1]. In certificateless-PKC, the KGC generates the partial private key for the user, and the full private key pair is the combination of the user’s secret value and the partial private key. The above signcryption schemes are based on a single receiver, which is insufficient for broadcast communication. For example, to send an identical message to multiple receivers, a sender must encrypt the message for each designated receiver, resulting in poor performance. Yu et al. [17] proposed the first multireceiver signcryption scheme based on ID-based PKC in which the message is encrypted for n number of designated receivers. The security is demonstrated through the Random Oracle Model (ROM) and the Computational Diffie-Hellman (CDH) assumption. Later, Hung et al. [6] proposed an efficient anonymous multireceiver certificateless encryption scheme based on bilinear pairing. The scheme proves Indistinguishability
against Chosen Ciphertext Attack (IND-CCA) and Anonymity against Chosen Ciphertext Attack (ANON-CCA). The encryption cost in this scheme is linear with the number of designated receivers, while the decryption cost is constant for each designated receiver.

In this paper, we extend the functionalities of Hung et al.’s encryption scheme into a Multireceiver Certificateless Signcryption (MCLS) scheme. Hung et al.’s encryption scheme demonstrate security against IND-CCA and ANON-CCA to prove confidentiality and anonymity whereas, we demonstrate security against IND-MCLS-CCA and EUF-MCLS-CMA for \( t \) designated receivers that proves confidentiality and authentication. Furthermore, as signcryption focuses on confidentiality and authentication, we omit the ANON-CCA proof, which remains the same for encryption scheme as in Hung et al. [6]. This signcryption scheme requires three additional multiplication operations for signcryption and unsigncryption, with the cost of signcryption being linear with the number of designated receivers and the cost of unsigncryption remaining constant for each designated receiver. In comparison to the other existing signcryption techniques listed at the end of this paper, the MCLS scheme avoids the key escrow problem and works in a multireceiver certificateless public key setting. Specifically, the main contributions are as follows.

- We design a Multireceiver Certificateless Signcryption (MCLS) scheme, that significantly extends the functionalities of the existing Efficient Anonymous Multireceiver Certificateless Encryption (AMCLE) scheme [6].
- We provide a detailed security proof in a ROM under the CDH and Decisional Bilinear Diffie-Hellman Inversion (DBDHI) assumptions which claim that the proposed scheme can achieve authentication by demonstrating Existential Unforgeability security against a Chosen Message Attack (EUF-MCLS-CMA).
- We evaluate the performance of the proposed MCLS scheme and present a comparison with other existing signcryption schemes.

The reminder of this paper is described as follows. Section 2 reviews the research related to the scheme. Section 3 introduces the fundamentals of bilinear pairings as well as mathematical assumptions. Section 4 describes the framework and security model in the MCLS scheme for two types of adversaries. Section 5 introduces the MCLS scheme. In Section 6 we perform a security analysis of the scheme under the assumption of hardness, and in Section 7 we compare it to existing schemes. Section 8 contains the conclusion.

2 Related Work

Barbosa and Farshim [2] proposed the first certificateless based signcryption scheme that provides confidentiality and authentication while protecting against Type-I and Type-II adversaries and is secure against insider attacks in a ROM. A Type-I adversary is a malicious user who can replace the public key of any
user but cannot access the master key of Trusted Authority (TA). A Type-II adversary is a malicious TA who can access the master key but cannot replace the public key of a user. The scheme is based on the bilinear pairing assumption. To prove the scheme’s security, it employs Gap-Bilinear Diffie Hellman (G-BDH), Decisional Bilinear Diffie-Hellman (DBDH), and Computational Bilinear Diffie-Hellman (CBDH) assumptions and shown to be IND-CPA and sUF-CMA secure. Selvi et al. proposed an efficient and provably secure certificateless multireceiver signcryption scheme [14]. The scheme is based on the Strong (DH) Problem, Collusion Attack Algorithm with K-Traitors (k-CAA), Modified BDHI for K-Values (k-mBDHIP), and Gap-BDH Problem. The scheme employs bilinear pairing operations and compares the efficiency of signcryption and unsigncryption operations to that of identity-based schemes. The scheme proposed by Selvi et al. is not secure against a Type-1 adversary and is improved as enhanced certificateless multireceiver signcryption scheme to prevent against Type-1 adversary [13]. However, Miao et al. proposed a cryptanalysis of a certificateless multireceiver signcryption scheme [11], in which the authors demonstrated that the scheme proposed in [13] is still insecure against a Type-I adversary and presented an attack on Selvi’s enhanced scheme. They demonstrate that the adversary can first replace the sender’s public key and then generate ciphertext on the sender’s behalf. Islam et al. [7] proposed an anonymous and provably secure certificateless multireceiver encryption (AMCLE) scheme which uses an Elliptic Curve Cryptography based technique under the CDH assumption. In this scheme, the encryption cost is quadric with the number of receivers, whereas the decryption cost is linear with the number of receivers, however, its security proof has a drawback that the simulator failed to successfully generate the challenge ciphertext and thus failed in the simulation. To overcome the key escrow problem and provide more efficiency, Hung et al. [6] proposed an Efficient Anonymous Multireceiver Certificateless Encryption (ACMLE) scheme that provides confidentiality and sender’s anonymity. This scheme uses bilinear pairing under the BDDH, Gap-BDH, and CDH assumptions. To prove confidentiality, the scheme defines the IND-CLME-CCA and to achieve anonymity, the authors present ANON-CLME-CCA. The proposed AMCLE scheme provides a constant decryption cost, which means that the required decryption cost of each receiver is independent of the number of receivers as compared to Islam et al.’s scheme. However, the security proof cannot cover all possible attacks due to some restrictions on attackers. Guo et al. [5] proposed an efficient certificateless ring signcryption scheme with conditional privacy preservation. The scheme employs a certificateless cryptographic technique and compares the results to identity-based cryptographic signature schemes. Hung et al.’s scheme provides efficient and anonymous multireceiver certificateless encryption based on bilinear pairing. In this paper, we substantially expand Hung et al.’s scheme and propose an efficient multireceiver certificateless signcryption that not only provides confidentiality but also authentication and non-repudiation.
3 Preliminaries and Assumptions

Here, we briefly review the basic definitions and properties of bilinear pairings and the related security assumptions on which the scheme is based. Let \( G_1 \) and \( G_2 \) be two cyclic additive groups and multiplicative groups respectively, over a prime order \( q \) where \( q \) is a large prime number. A pairing is a map: \( \hat{e} : G_1 \times G_1 \rightarrow G_2 \) which satisfies the bilinearity, computability, and non-degeneracy properties as follows.

- **Bilinearity:** For any \( P, Q \in G_1 \), \( \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} \) where \( a, b \in \mathbb{Z}_q^* \).
- **Computable:** For \( P, Q \in G_1 \), \( \hat{e}(P, Q) \) can be efficiently computed.
- **Non-degenerate:** \( \hat{e}(P, P) \neq 1 \), for some \( P \in G_1 \).

**Definition 1.** Decisional Bilinear Diffie-Hellman (DBDH) Assumption: On input \( P, aP, bP, cP \in G_1 \) and \( R \in G_2 \), the DBDH assumption holds if no PPT adversary \( A \) with non-negligible advantage can decide whether \( R = \hat{e}(P, P)^{abc} \) or not. The advantage of \( A \) is defined as

\[
\text{Adv}_{\text{DBDH}} = \Pr[A(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1] - \Pr[A(P, aP, bP, cP, R) = 1].
\] (1)

**Definition 2.** Computational Diffie-Hellman (CDH) Assumption: On input \( P, aP, bP \in G_1 \), the CDH assumption holds if no PPT adversary \( A \) with non-negligible advantage can compute \( abP \). The advantage of \( A \) is defined as

\[
\text{Adv}_{\text{CDH}} = \Pr[A(P, aP, bP) = abP].
\] (2)

**Definition 3.** GAP-Bilinear Diffie-Hellman (GBDH) Assumption: On input \( P, aP, bP, cP \in G_1 \), the Gap-BDH assumption holds if no PPT adversary \( A \) with non-negligible advantage can compute \( \hat{e}(P, P)^{abc} \) with the help of the DBDH oracle, where \( \text{DBDH}(P, aP, bP, cP, R) = 1 \) if \( \hat{e}(P, P)^{abc} = R \) and 0 otherwise. The advantage of \( A \) is defined as

\[
\text{Adv}_{\text{GBDH}} = \Pr[A(P, aP, bP, cP) = \hat{e}(P, P)^{abc}].
\] (3)

**Definition 4.** Decisional Bilinear Diffie-Hellman Inversion (DBDHI) Problem: On input \( P, aP, bP, cP \in G_1 \) and \( R \in G_2 \), the DBDHI assumption holds if no PPT adversary \( A \) with non-negligible advantage can decide whether \( R = \hat{e}(P, P)^{abc^{-1}} \) or not. The advantage of \( A \) is defined as

\[
\text{Adv}_{\text{DBDHI}} = \Pr[A(P, aP, bP, cP, \hat{e}(P, P)^{abc^{-1}}) = 1] - \Pr[A(P, aP, bP, cP, R) = 1].
\] (4)

4 Framework and Security Model

4.1 Framework

This paper adopts the certificateless signcryption scheme framework from AMCLE [6]. The AMCLE scheme allows a sender to produce the ciphertext of a
message for \( t \) designated receivers. The scheme has two roles; KGC and \( n \) number of users (a sender and \( t \) receivers) where \( t \leq n \).

The scheme consists of seven polynomial-time algorithms.

**Setup** \((1^\lambda)\). On input security parameter \( 1^\lambda \), the KGC runs this algorithm to generate a master secret key \( s \) and public parameters \( PP \). The public parameters \( PP \) are provided as input to other algorithms.

**Partial private key** \((s,ID)\). On input master secret key \( s \), public parameters \( PP \), and user’s identity \( ID \in \{0,1\}^* \), the KGC runs this algorithm to generate a partial private key \( DID \).

**Set secret value** \((ID)\). On input user identity \( ID \) and public parameters \( PP \), the user runs this algorithm and generates a user secret value \( x_{id} \).

**Set private key** \((DID, x_{id})\). Taking partial private key \( DID \) and the secret value \( x_{id} \) as input, the user runs this algorithm to generate a full private key \( SID \) for the identity \( ID \).

**Set public key** \((x_{id})\). Taking the secret value \( x_{id} \) as input, the user with identity \( ID \) runs this algorithm to generate a user public key \( PID \).

**Signcryption** \((m, SID, ((ID_1, PID_1), \ldots, (ID_t, PID_t)))\). On input public parameters \( PP \), a plaintext message \( m \), sender’s private key \( SID \), and receiver’s identity and public key \((ID_1, PID_1), \ldots, (ID_t, PID_t)\) where \( t \leq n \), a sender with identity \( ID \) runs a probabilistic algorithm to generate a ciphertext \( CT \).

**Unsigncryption** \((CT, ID, SID, PID)\). On input public parameters \( PP \), ciphertext \( CT \), sender’s identity \( ID \), designated receiver’s private key \( SID \), and sender’s public key \( PID \), the receiver runs deterministic algorithm to generate a plaintext message \( m \) or “reject”.

### 4.2 Security Model

For confidentiality, we define the Indistinguishability of Multireceiver Certificateless Signcryption against a Chosen Ciphertext Attack (IND-MCLS-CCA) from Hung et al.’s scheme. For authentication, we propose and define Existential Unforgeability against a Chosen Message Attack (EUF-MCLS-CMA). We consider two types of adversaries; Type-I and Type-II. A Type-I adversary is considered a common user who has no knowledge of KGC’s master secret key but can replace the public key of any identity with a value of his/her own choice. A Type-II adversary is considered an insider adversary who has access to the master secret key of KGC but cannot replace the public key of a legitimate user. A Type-II adversary is also known as a malicious KGC. In Definition 5 (Game-I), we define the IND-MCLS-CCA-I for Type-I adversary and the IND-MCLS-CCA-II for Type-II adversary, and in Definition 6 (Game-II), we define the EUF-MCLS-CMA-I for Type-I adversary and the EUF-MCLS-CMA-II for Type-II adversary.

**Definition 5.** The Indistinguishability of Multireceiver Certificateless Signcryption against Chosen Ciphertext Attack (IND-MCLS-CCA) requires that there exists no PPT Type-I and Type-II adversaries \( A \) which could distinguish ciphertexts. Therefore, the security game that captures confidentiality is based on the
ciphertext indistinguishability. The advantage of $A$ is defined as the probability that $A$ wins the game.

**Game-I (IND-MCLS-CCA-I, IND-MCLS-CCA-II):** This Game is interaction between the challenger $B$ and Type-I/Type-II adversary as follows.

**Setup.** The challenger $B$ generates the master secret key $s$ and public parameters $PP$. Then $B$ gives $PP$ to adversary $A$.

**Phase-1.** Without loss of generality in a certificateless (CL) setting, the Adversary $A$ outputs $t$ target identities denoted by $ID_i^*$ for $i \in \{1, ..., t\}$ where $t \leq n$. The Adversary $A$ further asks $q_i \{i = 0, ..., 6\}$ hash queries, $q_p$ public key retrieve query, $q_r$ public key replace query, $q_e$ partial private key query, $q_s$ secret value extract query, $q_{sc}$ signcryption query, and $q_{usc}$ unsigncryption query.

The Type-I adversary $A$ has following constraints.

1. Adversary cannot access master secret key $s$.
2. The adversary is not allowed to ask a partial private key query for any of the challenger identities.

The Type-II adversary $A$ has the following constraints.

1. Adversary cannot make public key replace query for the challenge identity.
2. Adversary is not allowed to make secret value extract queries.
3. If the public key replace query has been done for $ID_i^*$, then the secret value extract query for $ID_i^*$ is not allowed.

**Challenge.** The adversary $A$ outputs a target plaintext pair $\{m_0, m_1\}$. The challenger $B$ picks $\beta \in \{0, 1\}$ at random and sets $CT^* = E(PP; ((ID_1, PID_1), ..., (ID_t, PID_t)), m_\beta)$ challenger sends $CT^*$ to adversary $A$.

**Phase-2.** The adversary $A$ can make further queries except that the target ciphertext $CT^*$ is not allowed to appear in the unsigncryption queries.

**Guess.** Finally, $A$ responds with its guess $\beta \in \{0, 1\}^*$. If $\beta = \beta'$, $A$ wins the game. The advantage of Type-I adversary $A$ is defined as

$$Adv^\text{IND-MCLS-CCA-I}_A = |Pr[\beta = \beta'] - 1/2|.$$  \hspace{1cm} (5)

The advantage of Type-II adversary $A$ is defined as

$$Adv^\text{IND-MCLS-CCA-II}_A = |Pr[\beta = \beta'] - 1/2|.$$  \hspace{1cm} (6)

**Definition 6.** For Existential Unforgeability Against Chosen Message Attack (EUF-MCLS-CMA), we define Game-II that is played between a challenger $B$ and an adversary $A$. A certificateless multireceiver signcryption scheme is Type-I and Type-II EUF-CMA if every probabilistic PPT adversary $A$ has a negligible advantage in winning Game-II.

**Game-II (EUF-MCLS-CMA-I, EUF-MCLS-CMA-II):** This Game is interaction between the challenger $B$ and Type-I/Type-II adversary as follows.
Setup. The challenger B generates the master secret key s and public parameters PP. Then B gives PP to adversary A.

Phase-1. The adversary first outputs a target identity denoted by ID_A^∗. The Adversary A further asks q_i \{ i = 0, ..., 6 \} hash queries, q_p public key retrieve query, q_r public key replace query, q_c partial private key query, q_v secret value extract query, q_asc signcryption query, and q_usc unsigncryption query.

The Type-I adversary A has the following constraints.

1. Adversary cannot access master secret key s.
2. The adversary is not allowed to ask a partial private key query for any of the challenger identities.

The Type-II adversary A has the following constraints.

1. Adversary cannot make public key replace query for the challenge identity.
2. Adversary is not allowed to make secret value extract queries.
3. If the public key replace query has been done for ID_A^∗, then the secret value extract query for ID_A^∗ is not allowed.

Forgery. Adversary A outputs the forged signature under a target identity ID_A^∗. A wins if unsigncryption does not return ⊥.

5 Multireceiver Certificateless Signcryption Scheme

In this section, we define the MCLS scheme according to the framework defined in section 4.1. The main scheme is shown in Fig. 1.

**Setup**: Taking the security parameter λ as input, KGC initializes the system. It generates two large cyclic groups G_1 and G_2 of a large prime order q ≥ 2^λ, a bilinear pairing e : G_1 × G_1 → G_2, and selects a generator P of G_1. KGC defines seven hash functions H_0 : \{0,1\}^* → G_1, H_1, H_2 : G_2 × G_1 → \{0,1\}^w, H_3, H_4, H_5 : \{0,1\}^w → \{0,1\}^w, H_6 : \{0,1\}^* × G_1 → \mathbb{Z}_q^∗ for a positive integer w. KGC then chooses a master secret key s ∈ \mathbb{Z}_q^∗ at random, calculates system public key P_{pub} = s.P, chooses a symmetric encryption E_{sk} and a decryption D_{sk} function where sk is a symmetric key. KGC then publishes public parameters PP = (G_1, G_2, e, P, q, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, H_6, E_{sk}, D_{sk}) and keeps the master secret key s.

**Partial Private key**: Taking the user’s identity ID ∈ \{0,1\}^* as input, the KGC computes Q_ID = H_0(ID) and the associated partial private key D_ID = s.Q_ID. The KGC sends D_ID to the user via a secure channel.

**Set Secret Value**: User with identity ID ∈ \{0,1\}^* selects a positive integer x_{id} ∈ \mathbb{Z}_q^∗ as a secret value.

**Set Public Key**: The user with ID ∈ \{0,1\}^* takes x_{id} as input and generates the user public key P_{ID} = x_{id}.P.

**Set Private Key**: The user with ID ∈ \{0,1\}^* takes the partial private key D_ID and secret value x_{id} as input and generates the full private key S_ID = (D_ID, x_{id}).

**Signcryption**: Suppose, a sender generates a ciphertext to transfer a message m
to $t$ designated receivers with public keys $((ID_1, PID_1), ..., (ID_t, PID_t))$ where $t \leq n$. The sender runs the following steps.

- Chooses a value $r \in \mathbb{Z}_q^*$ randomly and computes $U = r.P, F_i = r.PID_i$ for $i = 1, ..., t$, $h_i = H_1(X_a, m)$ where $X_a = F_i \cdot x_{id}$.
- Computes $K_i = \hat{e}(P_{pub}, QID_i)^r, \text{QID}_i = H_0(ID_i)$ and $T_i = H_2(K_i, F_i)$ for $i = 1, ..., t$.
- Picks an ephemeral value $\sigma \in \{0,1\}^w$ randomly and computes $C_i$ for all $i = 1, ..., t$, $C_i = H_3(T_i) \parallel H_4(T_i) + \sigma$.
- Use the ephemeral value $\sigma$ to compute symmetric key $sk = H_5(\sigma)$ and generate $V = Esk(m)$.
- Signs the message $m$ as $Q = (r + h_i)x_{id}$ and sets $Q = \{Q_1, ..., Q_t\}$.
- To ensure data integrity, performs a hash operation $\tilde{\Lambda} = H_6(m, (C_1, ..., C_t), Q, V, U, \sigma)$.
- Set the ciphertext $CT = ((C_1, ..., C_t), Q, V, U, \tilde{\Lambda})$.

Unsigncryption: The designated receiver with identity $ID$ takes the ciphertext $CT = ((C_1, ..., C_t), Q, V, U, \tilde{\Lambda})$ as input, full private key $SID = (DID, x_{id})$ and runs the following steps.

- Computes $K = \hat{e}(U, DID)$.
- Computes $F = x_{id}.U, T = H_2(K, F)$ and $H_3(T)$.
- Uses $H_3(T)$ to find associated $C_i$ for $i = 1, ..., t$ by the relation $C_i = H_3(T) \parallel W$ where $W = H_4(T) + \sigma$.
- Computes $\sigma' = W + H_4(T)$.
- Sets symmetric key $sk' = H_5(\sigma')$ and computes $m' = Dsk(V)$ and $\tilde{\Lambda}' = H_6(m', \sigma', (C_1, ..., C_t), Q, V, U)$.
- If $\tilde{\Lambda}' = \tilde{\Lambda}$, then checks if $Q.P = F + h_i.PID$, hold or not. If it holds, receiver gets the message $m$, else returns ‘reject’.

Correctness Analysis

1. $K = \hat{e}(P_{pub}, QID)^r = \hat{e}(s.P, QID)^r = \hat{e}(r.P, s.QID) = \hat{e}(U, DID)$.
2. $F = r.PID = r.x_{id}.P = x_{id}.U$.
3. $Q.P = ((r + h_i)x_{id}).P = r.x_{id}.P + h_i.x_{id}.P = r.PID + h_i.PID = F + h_i.PID$

6 Security Analysis

Here, we illustrate that the MCLS scheme fulfills both confidentiality and unforgeability. For confidentiality, Theorems 4 and 5 below demonstrate that the scheme is secure against IND-MCLS-CCA Type-I and Type-II adversaries in the aforementioned Game-I in Definition 5. For unforgeability, Theorems 6 and 7 below demonstrate that the scheme is secure against EUF-MCLS-CMA Type-I and Type-II adversaries in the aforementioned Game-II in Definition 6.
**Theorem 1.** The MCLS scheme is provably secure against an IND-MCLS-CCA Type-I adversary. Assume that an IND-MCLS-CCA Type-I adversary $A$ with a non-negligible advantage $\epsilon$ can break the MCLS scheme with running time $\tau$, in ROM after $q_i \{i = 0, \ldots, 6\}$ hash queries, $q_p$ public key retrieve query, $q_r$ public key replace query, $q_e$ partial private key query, $q_s$ secret value extract query, $q_{sc}$ signcryption query, and $q_{usc}$ unsigncryption query. Then, there exists an algorithm that can solve the Gap-BDH problem with a non-negligible advantage $\epsilon'$ with running time $\tau'$ which are defined at the end of the Proof.

*Proof.* To solve the mathematical difficult problem Gap-BDH, the challenger $B$ is given an instance $(P, aP, bP, cP)$ where $P, aP, bP, cP \in G_1$ with unknown $a, b, c \in \mathbb{Z}_q^*$. Let $R = \hat{e}(P, P)^{abc}$ be the solution of the G-BDH problem. The challenger $B$ in Game-I (Definition 5) who would like to compute $R$ by interacting with adversary $A$ as follows.

**Setup:** $B$ runs the initialized algorithm and generates public parameters $PP = \{G_1, G_2, \hat{e}, P, q, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, H_6, E_{sk}, D_{sk}\}$ with $P_{pub} = aP$. The challenger $B$ sends $PP$ to adversary $A$.

**Phase-1:** Without loss of generality in a CL setting, the adversary first selects $t$ target identities of receivers denoted by $ID_i^*\{i \in \{1, \ldots, t\}\}$ where $t \leq n$ (in IND-MCLS-CCA-I, the target identity is the receiver’s identity denoted by $ID_i^*\{i \in \{1, \ldots, t\}\}$). The adversary $A$ makes number of queries including $q_i \{i = 0, \ldots, 6\}$ hash queries, $q_p$ query, $q_r$ query, $q_e$ query, $q_s$ query, $q_{sc}$ query, and...
$q_{asc}$ query. The challenger $B$ sets empty lists $PK^{list}$ to record public key values and maintains seven empty lists ($L_0, ..., L_6$) to record the responses of $q_i$ queries. The challenger responds to adversary’s queries by the following ways.

$H_0$ query: If there exists $(ID, u, QID)$ in $L_0$, $B$ returns $QID$ to $A$. Otherwise, $B$ performs the following steps.

- Picks a value $u \in Z_q^*$ at random. If $ID = ID_i^*$ for some $i \in \{1, ..., t\}$, sets $QID = u.bP$, otherwise sets $QID = u.P$. Stores $(ID, u, QID)$ in $L_0$ and responds with $QID$.

$H_1$ query: If there exists a record $(m, X_a, h)$ in $L_1$, $B$ returns $h$ to $A$. Otherwise, $B$ chooses a random string $h \in \{0, 1\}^w$ and returns to $A$. It stores $(m, X_a, h)$ in $L_1$.

$H_2$ query: If there exists $(K, F, T)$ in the list $L_2$, $B$ returns $T$ to $A$. Otherwise, $B$ picks a string $T \in \{0, 1\}^w$ at random, stores $(K, F, T)$ in $L_2$, and responds with $T$.

$H_3$ query: If there exists $(T, x)$ in the list $L_3$, $B$ returns $x$ to $A$. Otherwise, $B$ picks a string $x \in \{0, 1\}^w$ at random, stores $(T, x)$ in $L_3$, and responds with $x$.

$H_4$ query: If there exists $(T, y)$ in the list $L_4$, $B$ returns $y$ to $A$. Otherwise, $B$ picks a random string $y \in \{0, 1\}^w$, stores $(T, y)$ in $L_4$, and $B$ returns $y$ to $A$.

$H_5$ query: If there exists $(k, w)$ in the list $L_5$, $B$ returns $w$ to $A$. Otherwise, $B$ randomly picks a string $w \in \{0, 1\}^w$, stores $(k, w)$ in $L_5$, and responds with $w$.

$H_6$ query: If there exists $(m, \sigma, (C_1, ..., C_t), U, V, Q, \wedge)$ in the list $L_6$, $B$ returns $\wedge$ to $A$. Otherwise, $B$ picks a random value $\wedge \in Z_q^*$, stores the tuple $(m, \sigma, (C_1, ..., C_t), U, V, Q, \wedge)$ in $L_6$, and then returns $\wedge$ to $A$.

Public key retrieve query ($q_c$): If there exists $(ID, PID, x_{id})$ in $PK^{list}$, $B$ returns $PID$ to $A$. Otherwise, $B$ randomly picks $x_{id} \in Z_q^*$, sets $PID = x_{id}.P$ and stores $(ID, PID, x_{id})$ in $PK^{list}$ and provide $PID$ to $A$.

Public key replace query ($q_r$): $B$ replaces the associated tuple $(ID, PID, x_{id})$ in $PK^{list}$ with the new tuple $(ID, PID', \bot)$. Since, the secret value $x_{id}$ for $PID'$ is unknown, $B$ will set $\bot$ as $x_{id}$.

Partial private key query ($q_p$): If $ID = ID_i^*$, for some $i \in \{1, ..., t\}$, $B$ returns $\bot$ because, $ID_i^*$ is a target identity and a Type-I adversary is not allowed to ask for a partial private key query for the target identity. Otherwise, if $(ID, u, QID)$ exists in $L_0$, $B$ computes and returns $DID = u.P_{pub}$ to $A$. Otherwise, $B$ randomly picks a value $u \in Z_q^*$, sets $QID = u.P$, and $DID = u.P_{pub}$, and stores $(ID, u, QID)$ in $L_0$. $B$ returns $DID$ to $A$.

Secret value extract query ($q_s$): If $(ID, PID, x_{id})$ exists in $PK^{list}$, $B$ returns $x_{id}$ to the adversary $A$. Otherwise, $B$ randomly picks $x_{id} \in Z_q^*$, sets $PID = x_{id}.P$, stores $(ID, PID, x_{id})$ in $PK^{list}$ and returns $x_{id}$ to $A$.

Signcryption query ($q_{asc}$): When $B$ receives a signcryption query with a message $m$, sender $ID$, and receiver’s $ID_i$ for $i \in \{1, ..., t\}$ and public key in a tuple $((ID_1, PID_1), ..., (ID_t, PID_t))$. $B$ checks whether $ID = ID_i^*$ or not. If $ID \neq ID_i^*$, $B$ performs normal signcryption as this scheme. Otherwise, $B$ obtains the tuple $(DID, x_{id}, PID)$ via $q_p$, $q_c$ queries, and $q_s$ and generates a ciphertext $CT$ as follows.

- Pick a value $r \in Z_q^*$ randomly and compute $U = r.P$. 

11
\( C_1 = r \cdot PID_i, \ h_i = H_1(X_a,m), \ X_a = F_i \cdot x_{id} \) and add in \( L_1 \).

- Computes \( K_i = \hat{e}(U,DID), \ T_i = H_2(K_i,F_i) \) and adds in \( L_2 \).

- Picks an ephemeral value \( \sigma \in \{0,1\}^w \) at random and computes \( C_i = H_3(T_i) \parallel H_4(T_i) + \sigma \) and adds in \( L_3 \) and \( L_4 \).

- Use \( \sigma \) to compute symmetric key \( sk = H_5(\sigma) \) and generate \( V = E_{sk}(m) \).

- Compute \( Q = (r + h_i)x_{id} \) and compute \( \tilde{\Lambda} = H_6(m, \sigma, (C_1, \ldots, C_t), Q, V, U) \) and update \( L_6 \).

- Set \( C_T = (\langle C_1, \ldots, C_t \rangle, V, U, Q, \tilde{\Lambda}) \).

**Unsigncryption query \( q_{usc} \):** If \( ID \neq ID_i^* \) for \( i \in \{1, \ldots, t\} \), B can obtain its full private key \((DID, x_{id})\) via the \( q_e \) and \( q_s \) queries, unsigncrypt \( CT \) and return \( m \) to \( A \). Otherwise, B performs the following procedure.

- If \( \langle m, \sigma, (C_1, \ldots, C_t), Q, V, U, \tilde{\Lambda} \rangle \) is not in \( L_6 \), B terminates. Otherwise, B obtains \( \langle m, \sigma \rangle \) for possible utilization further in the following.

- B obtains \( QID \) from \( L_0 \) by issuing \( H_0 \) query.

For \( k = 1, \ldots, t \), B runs the following steps.

- Pick the leftmost \( w \) bits of \( C_k \) and denote it by \( x_k \).
- Pick the rightmost \( w \) bits of \( C_k \), denote it by \( w_k \).
- Compute \( y_k = w_k + \sigma \).
- Find a common \( T_k \) such that both the tuples \((T_k, x_k)\) and \((T_k, y_k)\) lie in the \( L_3 \) and \( L_4 \), respectively. If no such \( T_k \) exists, return 'abort'.
- Search \((K_k, F_k, T_k)\) associated with \( T_k \) from \( L_2 \). If not found, return 'abort'.
- Compute the value of \( h \) from \( L_1 \).
- Record the output of the query \( \text{DBDH} (P, QID, P_{pub}, U, K_k) \) to the \( \text{DBDH} \) oracle.
- If \( \text{DBDH} (P, PID, P_{pub}, U, K_k) = 1 \), for some \( k \), B compute \( sk' = H_4(\sigma) \) and \( m' = D'_{sk}(V) \).
- If \( m' = m \), B returns \( m \) to the adversary \( A \). In all other cases, B terminates.

**Challenge:** \( A \) gives a target plaintext pair \((m_0, m_1)\) to B. B randomly chooses \( \beta \in \{0,1\} \) and runs the following steps.

- Set \( U^* = cP \).
- Choose \( r^* \in Z_q^* \).
- \( F_i^* = r^* \cdot PID_i, \ X_i^* = F_i^* \cdot x_{id}, \ h_i^* = H_1(X_a, m_\beta) \).
- Pick a string \( \sigma \in \{0,1\}^w \) at random.
- For \( i = 1, \ldots, t \), randomly pick \( x_i^* \in \{0,1\}^w \) and \( y_i^* \in \{0,1\}^w \), and compute \( C_i^* = x_i^* \parallel (y_i^* \oplus \sigma^*) \).
- Compute \( sk = H_4(\sigma^*), \ V^* = E_{sk}(m_\beta), \) compute \( Q^* = (r^* + h_i^*)x_{id} \) and set \( \tilde{\Lambda}^* = (\langle C_1^*, \ldots, C_t^* \rangle, V^*, U^*, Q^* \) \).
- Finally, B returns \( CT^* = (\langle C_1^*, \ldots, C_t^* \rangle, V^*, U^*, \tilde{\Lambda}^*, Q^* \) \).

**Phase-2:** The adversary \( A \) may ask further queries as in Phase-2 but \( CT^* \) is not allowed to appear in the unsigncryption query.

**Guess:** The adversary \( A \) responds with its guess \( \beta' \in \{0,1\} \). If \( \beta = \beta' \), \( A \) wins.
the game.

The challenger will win the Game by obtaining \( R = \hat{e}(P, P)^{abc} \) which is solution to the DBDH problem. The challenger solves the DBDH problem by obtaining the list \( L_2 \) for \((K, F)\). Since, \( QID_i = u_i b P, \ P_{pub} = a P, \) and \( U^* = c P, \) B can obtain \( \hat{e}(P, P)^{abc} \) by evaluating \( K_{ui}^{ui-1}. \) Next, we evaluate the advantage of challenger B winning the Game-I (IND-MCLS-CCA-I) by calculating the probability of occurrence of the following events.

1. In the unsigncryption query if \( \langle C_1, \ldots, C_t \rangle, V, U, V_Q \) cannot be found in \( L_6 \), B returns 'failure' and terminates. The probability is \( 1/q_{H_6}. \)
2. In the partial private key query, the game terminates if \( ID \neq ID^*_i \) for some \( i \in \{1, \ldots, t\} \). The probability is \( 1/q_e. \)
3. In the unsigncryption query, the game aborts due to invalid message \( m' \neq m. \) The probability is \( q_{uns}/q. \)

Further, the challenger B will obtain the list \( L_2 \) for the some \((K, F)\) with the probability \( 1/q_{H_2}. \) Hence, if an IND-CLMS-CCA-I adversary \( A \) can break MCLS scheme with a non-negligible advantage \( \epsilon \), then the Gap-BDH problem can be solved with a non-negligible advantage \( \epsilon' \)

\[
\epsilon' \geq \epsilon \left( \frac{1}{q_{H_6}} \right) \left( \frac{1}{q_{H_2}} \right) \left( 1 - \frac{1}{q_e} \right) \left( 1 - \frac{q_{uns}}{q} \right)
\]

\( \tau' \) is the required computation time while answering the queries in the aforementioned simulation Game-I. It turns out that \( \tau' = \tau + O(q_o + q_p + q_e)\tau_1 + O(q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + q_r + q_s + q_{usc}) \) where \( \tau_1 \) is the time to perform a scalar multiplication in \( G_1 \) and \( t \) is the number of target identities.

**Theorem 2.** The scheme is provably secure against an IND-MCLS-CCA Type-II adversary with a non-negligible advantage \( \epsilon \) can break the MCLS scheme with running time \( \tau \) in ROM after \( q_i \{i = 0, \ldots, 6\} \) hash queries including \( q_p \) queries to public key retrieve query, \( q_s \) public key replace query, \( q_s \) secret value extract query, and \( q_{usc} \) unsigncryption query respectively. Then, there is an algorithm B that can solve the CDH problem with a non-negligible advantage \( \epsilon' \) with running time \( \tau' \) defined as the end of the Proof.

**Proof.** Assume that an algorithm B is given a random instance \((P, aP, bP)\) of the CDH problem, where \( P, aP, bP \in G_1 \), within unknown \( a, b \in Z_q^*. \) Let \( J = abP \) be the solution of the CDH problem. The algorithm B plays the challenger of Game-I (Definition 5) who would like to compute \( J \) by interacting with the adversary \( A \) as follows.

**Setup:** B runs the initialized algorithm and generates public parameters \( PP = \{G_1, G_2, \hat{e}, P, q, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, E_{sk}, D_{sk} \} \) with \( P_{pub} = aP. \) The challenger B sends \( PP \) to adversary \( A. \)

**Phase-1:** Without loss of generality in CL setting, the adversary first selects \( t \) target identities of denoted by \( ID^*_i \) for \( i \in \{1, \ldots, t\} \) where \( t \leq n \) (in IND-MCLS-CCA-II, the target identity is the receiver’s identity denoted by \( ID^*_i \) for
The adversary $A$ makes number of queries including a $H_0$ query, public key replace query, secret value extract query, and unsigncryption query. The challenger $B$ sets empty lists $PK^{list}$ to record public key values. Challenger $B$ responds to all the queries as follows.

$H_0$ query: If there exists $(ID, u, QID)$ in $L_0$, $B$ returns $QID$ to $A$. Otherwise, $B$ picks a value $u \in Z_q^*$ randomly and computes $QID = uP$. Then, $B$ Stores $(ID, u, QID)$ in $L_0$ and responds with $QID$.

Public key retrieve query ($q_p$): If there exists $(ID, P ID, x_{id})$ in the $PK^{list}$, $B$ returns $PID$ to $A$. Otherwise, $B$ performs the following step.

- Picks a value $x_{id} \in Z_q^*$ at random.
- If $ID = ID_i^*$ for some $i \in \{1, ..., t\}$, set $PID = x_{id}.a.P$ Otherwise, set $PID = x_{id}.P$, store $(ID, PID, x_{id})$ in the $PK^{list}$ and return $PID$ to $A$.

Public key replace query ($q_r$): If $ID = ID_i^*$ for some $i \in \{1, ..., t\}$, $B$ reports failure and terminates because, $ID_i^*$ is a target identity and Type-II adversary is not allowed to ask public key replace query for the target identity. Otherwise, $B$ replaces the associated tuple $(ID, PID, x_{id})$ in the $PK^{list}$ with a new tuple $(ID, PID', \bot)$.

Secret value extract query ($q_s$): If $ID = ID_i^*$ for some $i \in \{1, ..., t\}$, $B$ returns $\bot$ because, in this case, $ID_i^*$ is a target identity. If $(ID, PID, x_{id})$ exists in the $PK^{list}$, $B$ returns $x_{id}$ to the adversary $A$. Otherwise, $B$ randomly picks $x_{id} \in Z_q^*$ sets $PID = x_{id}.P$, stores $(ID, PID, x_{id})$ in $PK^{list}$, and returns $x_{id}$ to $A$.

Unsigncryption query ($q_{usc}$): If $ID \neq ID_i^*$, $B$ can obtain its full private key $(PID, x_{id})$ via the $q_s$ and $q_r$ queries, unsigncrypt ciphertext and return $m$ to $A$. Otherwise, $B$ perform the following procedure.

If $(m, \sigma, (C_1, ..., C_t), Q, V, U, A)$ is not in $L_6$, $B$ terminates. Otherwise, $B$ obtains $(m, \sigma)$ for possible utilization further in the following.

- $B$ obtains $QID$ associated with $ID$ from the list $PK^{list}$ or by issuing public key retrieve query.

For $k = 1, ..., t$, $B$ runs the following steps.

- Pick the leftmost $w$ bits of $C_k$ and denote it by $x_k$.
- Pick the rightmost $w$ bits of $C_k$, denote it by $w_k$.
- Compute $y_k = w_k + \sigma$.
- Find a common $T_k$ such that both the tuples $(T_k, x_k)$ and $(T_k, y_k)$ lie in the $L_3$ and $L_4$, respectively. If no such $T_k$ exists, return 'abort'.
- Search the tuple $(K_k, F_k, T_k)$ associate with $T_k$ from $L_2$. If not found, return 'abort'.
- If $\hat{e}(P, F_k) = \hat{e}(U, PID)$, record the value $k$.
- If $\hat{e}(P, F_k) = \hat{e}(U, PID)$ for some $k \in \{1, ..., t\}$, $B$ computes $sk' = H_4(\sigma)$ and $m' = D_{sk'}(V)$. If $m' = m$, $B$ returns $m$ to the adversary $A$. In all the other cases, $B$ terminates.

Challenge: $A$ gives a target plaintext pair $(m_0, m_1)$ to $B$. Then, $B$ randomly chooses $\beta \in \{0, 1\}$ and runs the following steps.
In the following, we assess the required computation time $\tau$ running time $\tau$ of an adversary with a non-negligible advantage.

**Theorem 3.** The scheme is provably secure against EUF-MCLS-CMA Type-I adversary with a non-negligible advantage $\epsilon$ can break the MCLS scheme with running time $\tau$ in ROM after $q_i \{i = 0, \ldots, 6\}$ hash queries including $q_p$ queries.
to public key retrieve query, $q_r$, public key replace query, $q_s$, secret value extract query, and $q_{usc}$ unsigncryption query respectively. Then, there is an algorithm $B$ that can solve the DBDHI problem with a non-negligible advantage $\epsilon'$ with running time $\tau'$ defined at the end of the Proof.

Proof. Assume that an algorithm $B$ is given a random instance $(P, aP, bP, cP)$ of the DBDHI problem, where $P, aP, bP, cP \in G_1$ with unknown $a, b, c \in \mathbb{Z}_q^*$. Let $R = \hat{c}(P, P)^{ab-c}$ be the solution of DBDHI problem. The algorithm $B$ plays the challenger in Game-II (Definition 6) who would like to compute $R$ by interacting with the adversary $A$ as follows.

Setup: $B$ runs the initialized algorithm and generates public parameters $PP = \{G_1, G_2, \hat{e}, P, q, pub, H_0, H_1, H_2, H_3, H_4, H_5, H_6, E_{sk}, D_{sk}\}$ with $P_{pub} = aP$. The challenger $B$ sends $PP$ to adversary $A$.

**Phase-1:** The adversary first selects a target identity denoted by $ID_A^*$ (in EUF-MCLS-CMA-I, the target identity is the sender’s identity denoted by $ID_A^*$). The adversary $A$ makes number of queries including a $H_0$ query, partial private key query, and signcryption query. The challenger $B$ responds to these queries as follows.

- **$H_0(ID)$ query:** If there exists $(ID, u, QID)$ in the list $L_0$, $B$ returns $QID$ to $A$. Otherwise, $B$ performs the following steps.
  - Picks a value $u \in \mathbb{Z}_q^*$ at random.
  - If $ID = ID_A^*$, sets $QID = ub^{-1}P$. Otherwise, sets $QID = uP$, stores $(ID, u, QID)$ in $L_0$ and responds with $QID$ to adversary $A$.

- **Partial private key query ($q_p$):** If $ID = ID_A^*$, $B$ returns $\bot$ because, $ID_A^*$ is a target identity and a Type-I adversary is not allowed to ask for a partial private key query for the target identity. Otherwise, if $(ID, u, QID)$ exists in $L_0$, $B$ computes and returns $DID = u.P_{pub}$ to $A$. Otherwise, $B$ randomly picks a value $u \in \mathbb{Z}_q^*$, sets $QID = u.P$, and $DID = u.P_{pub}$, and stores $(ID, u, QID)$ in $L_0$. $B$ returns $DID$ to $A$.

- **Signcryption query ($q_{sc}$):** When $B$ receives a signcrypt query with a message $m$, sender $ID$, and receiver’s identity $ID_i$ for $i \in \{1, ..., t\}$ and public key in a tuple $(ID_1, PID_1), ..., (ID_t, PID_t))$. It checks if $ID = ID_A^*$. If $ID \neq ID_A^*$, the challenger $B$ runs the normal signcryption algorithm. Otherwise, $B$ obtains the tuple $(DID, x_{id}, PID)$ via $q_{cr}$, $q_s$, and $q_{sc}$ queries and generates a ciphertext $CT$ via following procedure.
  - Pick a value $r \in \mathbb{Z}_q^*$ randomly and compute $U = rP$.
  - Compute $F_i = r.PID_i$, $h_i = H_1(X_a, m)$ where $X_a = F_i.x_{id}$ and adds in $L_1$.
  - Compute $K_i = \hat{c}(U, DID)$, $T_i = H_2(K_i, F_i)$ and adds in $L_2$.
  - Picks an ephemeral value $\sigma \in \{0, 1\}^w$ at random and compute $C_i = H_3(T_i) \parallel H_4(T_i) + \sigma$ and adds in $L_3$ and $L_4$.
  - Use $\sigma$ to compute symmetric key $sk = H_5(\sigma)$ and generate $V = E_{sk}(m)$.
  - Compute $Q = (r + h_i)x_{id}$ and compute $\bigwedge = H_6(m, \sigma, \langle C_1, ..., C_t\rangle, Q, V, U)$ and update $L_6$.
  - Set $CT = \langle\langle C_1, ..., C_t\rangle, V, U, Q, \bigwedge\rangle$.
$\textbf{Forgery}$: After the query phase completes, $A$ outputs the challenge identity $ID_A^t$, a receiver’s identity $ID_i$ for $i \in \{1, ..., t\}$, a message $m$, and a challenge ciphertext $CT^* = (C_1^*, ..., C_t^*, V^*, U^*, \wedge^*, Q^*)$. However, it cannot ask for the unsigncryption query for the challenge $CT^*$ with the private key of any target identity.

If the game does not abort, the challenger fetches the list $L_2$ for $(K, F)$ to obtain $R = \hat{\ell}(P, P)^{ab^{-1}c}$ which is the solution to the DBDHI problem. Since, $QID = ub^{-1}P$, $P_{pub} = aP$, and $U^* = cP$, $B$ can obtain $R = \hat{\ell}(P, P)^{ab^{-1}c}$ by evaluating $K_{\hat{\ell}}^{ab^{-1}}$. Hence, if the Game-II (EUF-MCLS-CMA-I) adversary $A$ can break the MCLS scheme with a non-negligible advantage $\epsilon$, the DBDHI problem can be solved with a non-negligible advantage $\epsilon'$

$$\epsilon' \geq \epsilon \left( \frac{1}{qH_6} \right) \left( \frac{1}{qH_2} \right) \left( 1 - \frac{1}{q_e} \right) \left( 1 - \frac{q_{uns}}{q} \right)$$

(9)

$\tau'$ is the required computation time while answering the queries in the aforementioned simulation game. It turns out that $\tau' = \tau + O(q_0 + q_p + q_e)\tau_1 + O(q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + q_7 + q_8 + q_{uns})$ where $\tau_1$ is the time to perform a scalar multiplication in $G_1$ and $t$ is the number of target identities.

**Theorem 4.** The scheme is provably secure against EUF-MCLS-CMA Type-II adversary with a non-negligible advantage $\epsilon$ can break the MCLS scheme with running time $\tau$ in ROM after $q_i \{i = 0, ..., 6\}$ hash queries including $q_p$ queries to public key retrieve query, $q_s$ public key replace query, $q_1$ secret value extract query, and $q_{uns}$ unsigncryption query respectively. Then, there is an algorithm $B$ that can solve the CDH problem with a non-negligible advantage $\epsilon'$ with running time $\tau'$ defined at the end of the Proof.

**Proof.** Assume that an algorithm $B$ is given a random instance $(P, aP, bP)$ of the CDH problem, where $P, aP, bP \in G_1$, within unknown $a, b \in Z_q^*$. Let $J = abP$ be the solution of the CDH problem. The algorithm $B$ plays the challenger of Game-II (Definition [ ] who would like to compute $J$ by interacting with the adversary $A$ as follows.

**Setup:** $B$ runs the initialized algorithm and generates public parameters $PP = \{G_1, G_2, \hat{\ell}, P, q, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, H_6, E_{sk}, D_{sk}\}$ with $P_{pub} = aP$. The challenger $B$ sends $PP$ to adversary $A$.

**Phase-1:** The adversary first selects a target identity denoted by $ID_A^t$ (in EUF-MCLS-CMA-II, the target identity is the sender’s identity denoted by $ID_A$). The adversary $A$ makes number of queries including a $H_0$ query, partial private key query, and signcryption query. The challenger $B$ responds to these queries as follows.

**Public key retrieve query ($qp$):** If there exists $(ID, PID, x_{id})$ in the list $PK^{list}$, $B$ returns $PID$ to $A$. Otherwise, $B$ performs the following steps.

- Pick $x_{id} \in Z_q^*$ at random.
- If $ID = ID_A^t$, set $PID = x_{id}aP$; otherwise, set $PID = x_{id}P$.
- Store $(ID, PID, x_{id})$ in the $PK^{list}$ and returns $PID$ to $A$. 

17
Signcryption query \((q_{sc})\): When B receives a signcrypt query with a message \(m\), sender \(ID\), and receiver’s identity \(ID_i\) for \(i \in \{1, ..., t\}\) and public key in a tuple \(((ID_1, PID_1), ..., (ID_t, PID_t))\). It checks if \(ID = ID_i^\ast\). If \(ID \neq ID_i^\ast\), formal signcryption algorithm runs. Otherwise, B obtains the tuple \((DID, x_{id}, PID)\) via \(q_p, q_e, \text{ and } q_s\) and generates a ciphertext \(CT\) via following procedure.

- Pick a value \(r \in Z_q^\ast\) randomly and compute \(U = rP\).
- Compute \(F_i = r.PID_i, h_i = H_1(X_a, m)\) where \(X_a, x_{id}\) and adds in \(L_1\).
- Compute \(K_i = \hat{e}(U, PID)\), \(T_i = H_2(K_i, F_i)\) and adds in \(L_2\).
- Picks an ephemeral value \(\sigma \in \{0, 1\}^w\) at random and compute \(C_i = H_3(T_i) \parallel H_4(T_i) + \sigma \) and adds in \(L_3\) and \(L_4\).
- Use \(\sigma\) to compute symmetric key \(sk = H_5(\sigma)\) and generate \(V = E_{sk}(m)\).
- Compute \(Q = (r + h_i)x_{id}\) and compute \(\Lambda = H_6(m, \sigma, (C_1, ..., C_t), Q, V, U)\) and update \(L_6\).
- Set \(CT = ((C_1, ..., C_t), V, U, Q, \Lambda)\).

Forgery: After the query phase completes, A outputs the challenge sender identity \(ID_i^\ast\), a receiver’s identity \(ID_i\) for \(i \in \{1, ..., t\}\), a message \(m\), and a challenge ciphertext \(CT^\ast = ((C_1^\ast, ..., C_t^\ast), V^\ast, U^\ast, \Lambda^\ast, Q^\ast)\). However, it cannot ask for the unsigncryption query for the challenge \(CT^\ast\) with the private key of any target identity.

If the game does not abort, the challenger fetches the list \(L_2\) for \((K, F)\) such that \(\hat{e}(P, F) = \hat{e}(U^\ast, PID)\). The challenger B can find such a \(F\) by verifying the equality \(\hat{e}(P, F) = \hat{e}(U^\ast, PID)\). Since, \(U^\ast = bP, PID = x_{id}(aP)\), B can obtains \(J = abP\) by evaluating \(F_i^2X_i^{-1}\) which is the solution to the CDH problem. Hence, if the Game-II (EUF-MCLS-CMA-II) adversary A can break the proposed MCLS scheme with a non-negligible advantage \(\epsilon\), the CDH problem can be solved with a non-negligible advantage \(\epsilon’\)

\[
\epsilon’ \geq \epsilon \left( \frac{1}{qH_0} \right) \left( \frac{1}{qH_2} \right) \left( 1 - \frac{1}{q_r} \right) \left( 1 - \frac{1}{q_s} \right) \left( 1 - \frac{q_{uns}}{q} \right) . \tag{10}
\]

In the following, we assess the required computation time \(\tau’\) while answering queries in the aforementioned simulation game. It turns out that \(\tau’ = \tau + O(q_0 + q_p)\tau_1 + O(q_{uns})\tau_2 + O(q_1 + q_2 + q_3 + q_4 + q_5 + q_6 + q_r + q_s)\), where \(\tau_1\) is the time to perform a scalar multiplication in \(G_1\), \(\tau_2\) is the time to perform a pairing operation and \(t\) is the number of target identities.

7 Performance Comparison and Discussion

Here, we compare the proposed MCLS scheme with the existing single receiver and multireceiver encryption and signcryption schemes which are mainly based on bilinear pairing operation \([15, 12, 8, 9, 16, 6]\). The notations are defined in Table \(1\) For single receiver signcryption schemes, Table \(2\) compares the computational cost of signcryption/unsigncryption, public key settings, and exemption from the key escrow problem with \([9]\) and \([8]\). Li et al. \([9]\) require
2T_p + 9T_m + T_e total operations whereas, the proposed MCLS scheme requires total 2T_p + 6T_m + T_e operations for signcryption/unsigncryption and works in CLPKC settings. Karati et al. \[8\] has equal computational cost as MCLS scheme however, it works in IDPKC and does not provide exemption of the escrow problem. The computational cost of multireceiver signcryption / unsigncryption, public key settings, and exemption from the key escrow problem are compared with \[12\], \[15\], and \[16\]. Wang et al. \[15\] require total (n + 1)T_e + (n + 2)T_m + 2T_p operations for signcryption and unsigncryption however, works in IDPKC-PKC settings and are not exempt from the key escrow problem. Niu et al. \[12\] require (n + 3)T_m + 2(n + 2)T_p + 2nT_e + T_{pa} total operations for multiple recipients and the scheme works in IDPKC for senders and CLPKC for receivers. Therefore, the sending entities are not exempt from the key escrow problem. Furthermore, Yang et al. \[16\] require total 3nT_m + 5nT_{pm} + nT_p + nT_e signcryption / unsigncryption operations whereas, it works in IDPKC and is not exempt from the key escrow.

### Table 1. Notations

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_p</td>
<td>The time of executing a bilinear pairing operation $\hat{e} : G_1 \times G_1 \rightarrow G_2$.</td>
</tr>
<tr>
<td>T_m</td>
<td>The time of executing a scalar multiplication operation in $G_1$.</td>
</tr>
<tr>
<td>T_e</td>
<td>The time of executing an exponentiation in $G_2$ or an exponentiation operation in $\mathbb{Z}_q^*$.</td>
</tr>
<tr>
<td>T_{pm}</td>
<td>The time of executing an multiplication operation in $G$.</td>
</tr>
<tr>
<td>n</td>
<td>The number of receivers.</td>
</tr>
<tr>
<td>IDPKC</td>
<td>Identity-based Public Key Cryptography.</td>
</tr>
<tr>
<td>CLPKC</td>
<td>Certificateless Public Key Cryptography.</td>
</tr>
<tr>
<td>Enc</td>
<td>Encryption.</td>
</tr>
<tr>
<td>Dec</td>
<td>Decryption.</td>
</tr>
</tbody>
</table>

### Table 2. Comparison between MCLS scheme with the existing single and multi-receiver encryption and signcryption schemes based on bilinear pairings.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Single receiver</th>
<th>Multireceiver</th>
<th>Public key settings</th>
<th>Exemption of key escrow</th>
</tr>
</thead>
<tbody>
<tr>
<td>LI et al. (2019) [9]</td>
<td>$T_p + 4T_m + T_e$</td>
<td>$T_p + 5T_m$</td>
<td>Dec/Unsigncrypt</td>
<td>IDPKC - CLPKC</td>
</tr>
<tr>
<td>Karati et al. (2018) [8]</td>
<td>$4T_e$</td>
<td>$2T_p + 2T_e + T_i$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Niu et al. (2017) [12]</td>
<td>-</td>
<td>-</td>
<td>$(n + 2)T_m + 2nT_p + 2nT_e$</td>
<td>$4T_p + T_m + T_{pa}$</td>
</tr>
<tr>
<td>Wang et al. (2017) [15]</td>
<td>-</td>
<td>-</td>
<td>$(n + 1)T_e + nT_m$</td>
<td>$2T_p + 2T_m$</td>
</tr>
<tr>
<td>Yang et al. (2022) [16]</td>
<td>-</td>
<td>-</td>
<td>$2nT_m + 2nT_{pm} + nT_p$</td>
<td>$4T_m + 3nT_{pm} + nT_p$</td>
</tr>
<tr>
<td>Hung et al. (2017) [6]</td>
<td>-</td>
<td>-</td>
<td>$nT_p + nT_e + (n + 1)T_m$</td>
<td>$T_p + T_m$</td>
</tr>
<tr>
<td>Our scheme</td>
<td>$T_p + T_e + 2T_m$</td>
<td>$T_p + 2T_m$</td>
<td>$nT_p + nT_e + (2n + 2)T_m$</td>
<td>$T_p + 2T_m$</td>
</tr>
</tbody>
</table>
From the multireceiver signcryption schemes the MCLS scheme requires total \((n + 1)T_p + (2n + 4)T_m + nT_e\) operations, works in certificateless settings, and exempts from the key escrow. The signcryption cost in the MCLS scheme is linear with the number of designated receivers, while the unsigncryption cost is constant for each receiver. Further, Hung et al.’s scheme requires \(nT_p + nT_e + (n + 1)T_m\) operations for encryption phase and \(T_p + T_m\) operations for decryption. Therefore, Hung et al.’s scheme require total \((n+1)T_p+nT_e+(n+2)T_m\) operations for encryption and decryption. As compared to Hung et al.’s scheme, the MCLS require only three additional multiplication operations for signcryption and unsigncryption resulting in total \((n + 1)T_p + (2n + 4)T_m + nT_e\) operations, whereas, the signcryption cost is linear with the number of designated receivers and the unsigncryption cost is constant with the number of designated receivers. Finally, while the proposed scheme significantly extends Hung et al.’s efficient anonymous multireceiver certificateless encryption scheme into a multireceiver certificateless signcryption scheme (secure against a chosen message attack), this comes at a relatively small cost as shown in Table 2.

8 Conclusion

This paper presents a Multireceiver Certificateless Signcryption (MCLS) scheme to fulfill both confidentiality and authentication requirements, building on Hung et al.’s encryption scheme. In the proposed scheme, the message is encrypted and signed with sender’s private key, ensuring the message reliability and authenticity. While traditional multireceiver signcryption schemes require each receiver to verify the message, which increases computational cost, in this scheme, the required unsigncryption cost of each receiver is constant and independent of the number of receivers. We formally demonstrate the semantic security of the scheme against the IND-CCA (from Hung et al.’s scheme) and EUF-CMA (proposed) attacks in the random oracle model using the Gap-BDH, CDH, and DBDHI assumptions, respectively. Finally, we compare the proposed MCLS scheme’s performance and functionality to existing single receiver and multi-receiver signcryption approaches. In comparison to other existing single receiver and multireceiver signcryption techniques, the MCLS scheme is more efficient, while both avoiding the key escrow problem and work in multireceiver certificateless public key setting.

References


