IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 August 2025, accepted 7 September 2025, date of publication 10 September 2025,
date of current version 18 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3608453

RESEARCH ARTICLE

Cryptanalysis and Modification of Yang et al.'s Certificateless Multi-User Matchmaking Encryption Scheme

JISEUNG KIM[1] **AND HYUNG TAE LEE**[2]
[1]Department of Computer Science and Artificial Intelligence, Jeonbuk National University, Jeonju 54896, Republic of Korea
[2]School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Republic of Korea

Corresponding author: Hyung Tae Lee (hyungtaelee@cau.ac.kr)

This work was supported in part by the Institute of Information and communications Technology Planning and evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00399491, 50%). The work of Hyung Tae Lee was also supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2021-NR058311 (25%), NRF-2022R1A4A5034130 (25%)).

ABSTRACT Matchmaking encryption, which is a cryptographic primitive that enables two-sided access control for both senders and receivers, has attracted significant attention from cryptography community due to its broad range of applications. Recently, Yang et al. proposed a certificateless multi-user matchmaking encryption scheme that was claimed to achieve IND-CCA2 security (IEEE Transactions on Information Forensics and Security, 2024). In this paper, we propose a plaintext recovery attack against their scheme under an adaptive chosen ciphertext attack model. We further identify critical flaws in both the scheme's design and its accompanying security proof. To address these issues, we propose a revised construction that mitigates the identified vulnerabilities and restores the intended security guarantees. Our modification incurs only minimal overhead; hence, the computational and communication efficiency of the modified scheme remains almost equivalent to that of the original design.

INDEX TERMS Certificateless matchmaking encryption, IND-CCA2 security, adaptive chosen ciphertext attack.

I. INTRODUCTION

Matchmaking encryption (ME) is a cryptographic paradigm that enforces two-sided access control, enabling both senders and receivers to specify their own access policies. This distinctive feature makes ME particularly well-suited for a wide range of applications, including secure social networking, anonymous bulletin boards, and privacy-preserving data sharing. Since its initial proposal by Ateniese et al., who introduced the concept along with concrete instantiations [1], the research community has actively explored various extensions and refinements of ME [2], [3], [4], [5], [6], [7], [8], aiming to enhance its functionality and strengthen its security guarantees. Notably, a number of recent works [5], [6], [7], [8] have concentrated on achieving indistinguishability under adaptive chosen ciphertext attacks (IND-CCA2), a stringent security notion for modern encryption scheme. Among these, the scheme proposed by Yang et al. [8] stands out as a lightweight certificateless multi-user matchmaking encryption (LC-MUME) scheme, tailored for resource-constrained environments such as mobile devices. Their construction was claimed to achieve IND-CCA2 security in the random oracle model.

In this paper, we present an adaptive chosen ciphertext attack (CCA2 attack) against LC-MUME scheme proposed by Yang et al. [8]. That is, we refute their claim that the scheme achieves IND-CCA2 security by constructing a concrete attack algorithm that enables plaintext recovery. The core idea of our attack leverages the malleability inherent in the ciphertext structure. Specifically, a ciphertext

The associate editor coordinating the review of this manuscript and approving it for publication was S. K. Hafizul Islam.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

corresponding to a plaintext m in Yang et al.'s scheme is represented as a tuple $(C_1, C_2, C_3, C_4, C_5)$, where

$$C_3 = \mathcal{H}(C_1, C_2, d_1, d_2) \oplus m,$$
$$C_4 = \overline{\mathcal{H}}(C_1, C_2, C_3, C_5)$$

with d_1 and d_2 being random values chosen during encryption, and $\mathcal{H}, \overline{\mathcal{H}}$ denoting appropriate cryptographic hash functions. Given any chosen value $\overline{m}$, an adversary can manufacture a new valid ciphertext $(C_1, C_2, \overline{C}_3, \overline{C}_4, C_5)$ as follows:

$$\overline{C}_3 = C_3 \oplus \overline{m} = \mathcal{H}(C_1, C_2, d_1, d_2) \oplus (m \oplus \overline{m}) \text{ and}$$
$$\overline{C}_4 = \overline{\mathcal{H}}(C_1, C_2, \overline{C}_3, C_5).$$

By querying this manipulated ciphertext to the decryption oracle, the adversary obtains $m \oplus \overline{m}$, from which the original plaintext m can be easily recovered using the known value $\overline{m}$. We emphasize that access to the decryption oracle–except on the challenge ciphertext–is permitted in the IND-CCA2 security model, thus validating the feasibility of our attack.

Following our attack, we identify specific flaws in both the design and the security proof of the LC-MUME scheme. To address these issues, we propose a modified construction that mitigates the identified vulnerabilities. Our enhanced scheme generates a ciphertext of the form $(C_1, C_2, C_3', C_4', C_5)$ where

$$C_3' = \mathcal{H}'(C_1, C_2, d_1, d_2) \oplus (m \| r) \text{ and}$$
$$C_4' = \overline{\mathcal{H}}'(C_1, C_2, C_3', C_5, m, r)$$

and r is a random scalar used to compute $C_1 = rP$, with P being the generator to the underlying group $\mathbb{G}$. Here, $\mathcal{H}'$ and $\overline{\mathcal{H}}'$ denote appropriate cryptographic hash functions. Our modification draws inspiration from a construction technique used to achieve IND-CCA2 security in the context of public-key encryption with equality test [9]. To successfully adapt our attack to the modified scheme, an adversary would need to correctly compute C_4' and recover the secret value r embedded in the challenge ciphertext. However, due to the computational hardness of extracting r from $C_1 = rP$, or from the components C_3' and C_4', such an attack becomes infeasible–thereby rendering our original attack ineffective against the revised construction. Finally, we note that our modification incurs only minimal overhead, as it merely increases the output length of C_3' and the input length of $\overline{\mathcal{H}}'$ involved in the computation of C_4'. Consequently, the computational and communication efficiency of the modified scheme remains nearly identical to that of the original construction.

Organization of the Paper: Section II presents a review of Yang et al.'s LC-MUME scheme [8]. In Section III, we describe our attack algorithm in detail and then analyze the flaws in both the design and security proof in the original paper. Subsequently, we propose a plausible solution to mitigate the identified vulnerability in Section IV.

II. REVIEW OF YANG ET AL.'S LIGHTWEIGHT CERTIFICATELESS MULTI-USER MATCHMAKING ENCRYPTION

In this section, we review the LC-MUME scheme proposed by Yang et al. [8]. We begin by introducing the notations that will be used throughout the paper. For any bit string a, we denote its first ℓ bits by $[a]_\ell$, and its last ℓ bits by $[a]^\ell$. Given two bit strings a and b, their concatenation is represented by $a\|b$.

Yang et al.'s LC-MUME scheme is composed of five algorithms–**Setup**, **SKGen**, **RKGen**, **Enc**, and **Dec**–which are outlined as follows:

- **Setup**: Given a security parameter 1^λ, the key generation center (KGC) performs the following steps:
  1) Generate an additive cyclic group $\mathbb{G}$ of order q, defined over an elliptic curve $\mathbb{E}(\mathbb{F}_p)$, with a generator P, where p and q are sufficiently large primes.
  2) Choose a random element $\alpha \in \mathbb{Z}_q^*$ to serve as the master secret key msk.
  3) Compute the master public key $\mathcal{P}' = \alpha P$.
  4) Select the following five cryptographic (collision-resistant) hash functions:
     - $\mathcal{H} : \mathbb{G} \to \mathbb{Z}_q^*$
     - $\mathcal{H}_1 : \{0,1\}^* \times \mathbb{G} \to \mathbb{Z}_q^*$
     - $\mathcal{H}_2 : \mathbb{G}^2 \times (\mathbb{Z}_q^*)^2 \to \{0,1\}^l$
     - $\mathcal{H}_3 : \mathbb{G}^2 \times \{0,1\}^l \times (\mathbb{Z}_q^*)^{2n} \to \{0,1\}^{\text{len}}$
     - $\mathcal{H}_4 : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G} \to \mathbb{Z}_q^*$

     where l and len are appropriate positive integers.
  5) Output the public parameter $\mathsf{pp} = (q, P, \mathbb{G}, \mathcal{H}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4, \mathcal{P}')$ and keep the master secret key $\text{msk} = \alpha$ private.

  Note that the KGC maintains the user list for managing users who wish to join the system.
- **SKGen**: This protocol is executed between the KGC and a sender with identity id_0. The KGC takes as input the public parameter pp, the master secret key msk, and sender's identity id_0, while the sender takes pp and id_0 as input.
  1) The KGC selects a random integer $r_0 \in \mathbb{Z}_q^*$ and computes partial keys
  $$\mathcal{R}_0 = r_0 P,$$
  $$d_0 = r_0 + \alpha \mathcal{H}_1(\text{id}_0, \mathcal{R}_0) \pmod q.$$
  The KGC adds the sender id_0 to the user list and transmits the tuple $(\mathcal{R}_0, d_0)$ to the sender via a secure channel.
  2) Upon receiving $(\mathcal{R}_0, d_0)$, the sender proceeds as follows:
     a) Check if $d_0 P = \mathcal{R}_0 + \mathcal{H}_1(\text{id}_0, \mathcal{R}_0)\mathcal{P}'$. Abort if the check fails.

b) If the check passes, choose a random integer $x_0 \in \mathbb{Z}_q^*$ and compute $\mathcal{X}_0 = x_0 P$.
c) Set the encryption private key $\mathrm{ek}_{\mathrm{id}_0} = (\mathrm{ek}^1_{\mathrm{id}_0}, \mathrm{ek}^2_{\mathrm{id}_0}) = (x_0, d_0)$, and output the corresponding public key $\mathrm{epk}_{\mathrm{id}_0} = (\mathrm{epk}^1_{\mathrm{id}_0}, \mathrm{epk}^2_{\mathrm{id}_0}) = (\mathcal{X}_0, \mathcal{R}_0)$.

- **RKGen**: This protocol is executed between the KGC and a receiver with identity id_i. The KGC takes as input the public parameter pp, the master secret key msk, and receiver's identity id_i, while the receiver takes pp and id_i as input.
  1) The KGC selects a random integer $r_i \in \mathbb{Z}_q^*$ and computes partial keys

$$\mathcal{R}_i = r_i P,$$
$$d_i = r_i + \alpha \mathcal{H}_1(id_i, \mathcal{R}_i) \pmod q.$$

  The KGC adds the receiver id_i to the user list and transmits the tuple $(\mathcal{R}_i, d_i)$ to the sender via a secure channel.
  2) Upon receiving $(\mathcal{R}_i, d_i)$, the sender proceeds as follows:
     a) Check if $d_i P = \mathcal{R}_i + \mathcal{H}_1(\mathrm{id}_i, \mathcal{R}_i)\mathcal{P}'$. Abort if the check fails.
     b) If the check passes, choose a random integer $x_i \in \mathbb{Z}_q^*$ and compute $\mathcal{X}_i = x_i P$.
     c) Set the decryption private key $\mathrm{dk}_{\mathrm{id}_i} = (\mathrm{dk}^1_{\mathrm{id}_i}, \mathrm{dk}^2_{\mathrm{id}_i}) = (x_i, d_i)$, and output the corresponding public key $\mathrm{dpk}_{\mathrm{id}_i} = (\mathrm{dpk}^1_{\mathrm{id}_i}, \mathrm{dpk}^2_{\mathrm{id}_i}) = (\mathcal{X}_i, \mathcal{R}_i)$.
- **Enc**: Given the public parameter pp, an encryption key $\mathrm{ek}_{\mathrm{id}_0} = (\mathrm{ek}^1_{\mathrm{id}_0}, \mathrm{ek}^2_{\mathrm{id}_0})$ associated with identity id_0, a target identity set $\mathcal{S}$ of size n, the public keys $\mathsf{dpk} = \{\mathrm{dpk}_{\mathrm{id}_i} = (\mathrm{dpk}^1_{\mathrm{id}_i}, \mathrm{dpk}^2_{\mathrm{id}_i}) | \mathrm{id}_i \in \mathcal{S}\}$ corresponding to the decryption private key, and a plaintext $m \in \{0,1\}^{l_1}$, the encryption algorithm proceeds as follows:
  1) Sample random integers $r, s, d_1, d_2 \in \mathbb{Z}_q^*$, and compute $C_1 = rP$ and $C_2 = sP$.
  2) For each $\mathrm{id}_i \in \mathcal{S}$, compute the intermediate values

$$\tau_i = \mathrm{ek}^1_{\mathrm{id}_0} \mathrm{dpk}^1_{\mathrm{id}_i},$$
$$V_{\mathrm{id}_i} = \mathcal{H}(r\mathcal{H}_4(\mathrm{id}_0, \mathrm{id}_i, \tau_i)(\mathrm{dpk}^2_{\mathrm{id}_i} + \mathcal{H}_1(\mathrm{id}_i, \mathrm{dpk}^2_{\mathrm{id}_i})\mathcal{P}')),$$
$$Z_{\mathrm{id}_i} = (s + \mathrm{ek}^2_{\mathrm{id}_0} + \mathrm{ek}^1_{\mathrm{id}_0}\mathcal{H}_4(\mathrm{id}_0, \mathrm{id}_i, \tau_i))\mathrm{dpk}^1_{\mathrm{id}_i}.$$

  3) Construct two degree-n polynomials $f(x)$ and $g(y)$ defined as follows:

$$f(x) = \prod_{\mathrm{id}_k \in \mathcal{S}} (x - V_{\mathrm{id}_k}) + d_1 = \sum_{k=0}^{n-1} a_k x^k + x^n,$$
$$g(y) = \prod_{\mathrm{id}_k \in \mathcal{S}} (y - Z_{\mathrm{id}_k}) + d_2 = \sum_{k=0}^{n-1} b_k y^k + y^n$$

over $\mathbb{Z}_q$ where $a_k, b_k \in \mathbb{Z}_q$ denote the coefficients corresponding to x^k and y^k, respectively.
  4) Compute C_3 and C_4 as follows:

$$C_3 = [\mathcal{H}_2(C_1, C_2, d_1, d_2)]_{l-l_1} \| ([\mathcal{H}_2(C_1, C_2, d_1, d_2)]^{l_1} \oplus m),$$
$$C_4 = \mathcal{H}_3(C_1, C_2, C_3, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}).$$

  5) Output a ciphertext

$$\mathsf{ct} = (C_1, C_2, C_3, C_4, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}).$$

- **Dec**: Given the public parameter pp, a decryption private key $\mathrm{dk}_{\mathrm{id}_i} = (\mathrm{dk}^1_{\mathrm{id}_i}, \mathrm{dk}^2_{\mathrm{id}_i})$, a target sender identity $\mathsf{snd} = \mathrm{id}_0$ along with the corresponding encryption public key $\mathrm{epk}_{\mathrm{id}_0} = (\mathrm{epk}^1_{\mathrm{id}_0}, \mathrm{epk}^2_{\mathrm{id}_0})$, and a ciphertext $\mathsf{ct} = (C_1, C_2, C_3, C_4, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1})$, the decryption algorithm proceeds as follows:
  1) Check if the following holds:

$$\mathcal{H}_3(C_1, C_2, C_3, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}) = C_4. \quad (1)$$

  If the check fails, output $\perp$.
  2) If the above check passes, perform the following computations:

$$\tau_i = \mathrm{dk}^1_{\mathrm{id}_i} \mathrm{epk}^1_{\mathrm{id}_0},$$
$$V_{\mathrm{id}_i} = \mathcal{H}(\mathrm{dk}^2_{\mathrm{id}_i} \mathcal{H}_4(\mathrm{id}_0, \mathrm{id}_i, \tau_i) C_1),$$
$$Z_{\mathrm{id}_i} = \mathrm{dk}^1_{\mathrm{id}_i}(C_2 + \mathrm{epk}^2_{\mathrm{id}_0} + \mathcal{H}_1(\mathrm{id}_0, \mathrm{epk}^2_{\mathrm{id}_0})\mathcal{P}' + T)$$

  where $T = \mathcal{H}_4(\mathrm{id}_0, \mathrm{id}_i, \tau_i)\mathrm{epk}^1_{\mathrm{id}_0}$. Using the coefficients $a_0, \ldots, a_{n-1}, b_0, \ldots b_{n-1}$, reconstruct the polynomials

$$f(x) = \sum_{k=0}^{n-1} a_k x^k + x^n \quad \text{and} \quad g(y) = \sum_{k=0}^{n-1} b_k y^k + y^n$$

  and evaluate them at the computed values V_{id_i} and Z_{id_i}, respectively, to obtain

$$d_1 = f(V_{\mathrm{id}_i}) \quad \text{and} \quad d_2 = g(Z_{\mathrm{id}_i}).$$

  3) Output $m = [\mathcal{H}_2(C_1, C_2, d_1, d_2)]^{l_1} \oplus [C_3]^{l_1}$ if

$$[C_3]_{l-l_1} = [\mathcal{H}_2(C_1, C_2, d_1, d_2)]_{l-l_1}. \quad (2)$$

  Otherwise, output $\perp$.

We refer to Theorem 1 of the original paper [8] for the correctness of Yang et al.'s scheme.

III. OUR CCA2 ATTACK ON YANG ET AL.'S LC-MUME SCHEME

In this section, we demonstrate a practical attack against the LC-MUME scheme proposed by Yang et al. [8]. Our analysis is based on the IND-CCA2 attack, which was the security level the original scheme claimed to achieve. The IND-CCA2 security model assumes an adversary who has access to a decryption oracle both before and after receiving the

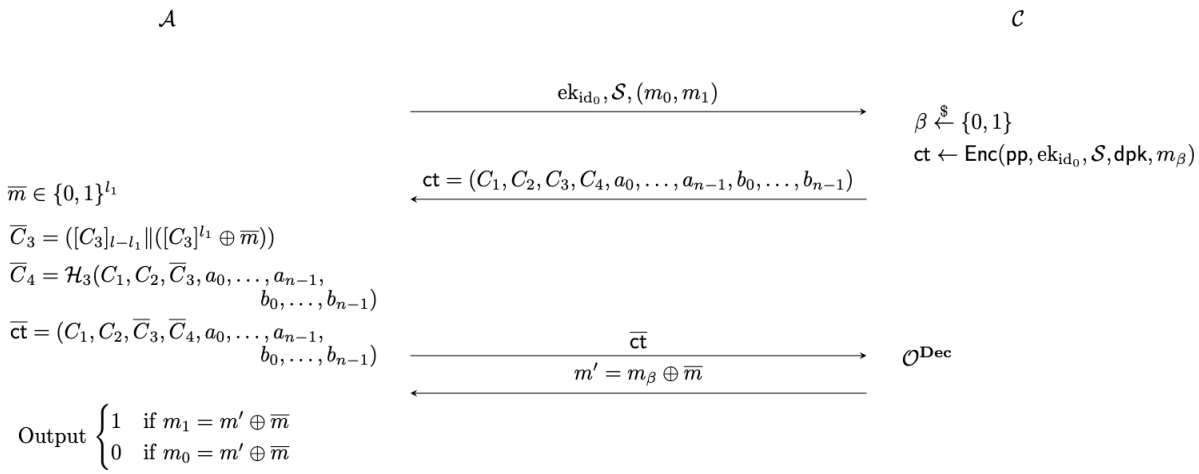


FIGURE 1. Our CCA2 attack against Yang et al.'s LC-MUME [8].

challenge ciphertext and is allowed to decrypt any ciphertext except the challenge ciphertext itself. Our attack shows that by leveraging this oracle access, an adversary can completely break the LC-MUME scheme.

A. OUR ATTACK

We present an algorithm that allows the adversary to recover the entire plaintext of a challenge ciphertext with only a single query to the decryption oracle, thereby proving that the scheme is not IND-CCA2 secure. A graphical illustration of the proposed attack is provided in Figure 1.

In the IND-CCA2 security game, assume that the adversary $\mathcal{A}$ is given a challenge ciphertext $\mathsf{ct} = (C_1, C_2, C_3, C_4, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1})$, which encrypts a message m_β for a randomly chosen bit $\beta \in \{0, 1\}$ by the challenger $\mathcal{C}$. To launch the attack, the adversary first selects a plaintext $\overline{m}$ from the plaintext space $\{0, 1\}^{l_1}$, and then sets $\overline{C}_3$ and $\overline{C}_4$ by calculating

$$\begin{aligned} &\overline{C}_3 = ([C_3]_{l-l_1} \| ([C_3]^{l_1} \oplus \overline{m})) \text{ and} \\ &\overline{C}_4 = \mathcal{H}_3(C_1, C_2, \overline{C}_3, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}). \end{aligned} \quad (3)$$

We claim that the tuple $\overline{\mathsf{ct}} = (C_1, C_2, \overline{C}_3, \overline{C}_4, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1})$ constitutes a valid ciphertext of the message $m_\beta \oplus \overline{m}$. To support this claim, we initially observe that during the **Dec** algorithm, the ciphertext components C_3 and C_4 are used in specific parts:

- Determining whether Eq. (1) holds or not in Step 1),
- Determining whether Eq. (2) holds or not and recovering m in Step 3).

For given $\overline{\mathsf{ct}}$, we note that Eq. (1) is automatically satisfied from Eq. (3). Additionally, by the definition of $\overline{C}_3$, it is clear that $[C_3]_{l-l_1} = [\overline{C}_3]_{l-l_1}$. Consequently, Eq. (2) is always fulfilled if the input ciphertext ct of the **Dec** algorithm is a valid ciphertext. Moreover,

$$\begin{aligned} \overline{C}_3 &= [C_3]_{l-l_1} \| ([C_3]^{l_1} \oplus \overline{m}) \\ &= \mathcal{H}_2(C_1, C_2, d_1, d_2)]_{l-l_1} \| \\ &\qquad ([\mathcal{H}_2(C_1, C_2, d_1, d_2)]^{l_1} \oplus (m_\beta \oplus \overline{m})). \end{aligned}$$

Thus, our claim that $\overline{\mathsf{ct}}$ is a valid ciphertext of $m_\beta \oplus \overline{m}$ holds true, provided that the challenge ciphertext ct is a valid ciphertext of m_β.

If the adversary has access to the decryption oracle $\mathcal{O}^{\mathbf{Dec}}$, it can derive the plaintext $m_\beta \oplus \overline{m}$ by requesting a decryption query on the valid ciphertext $\overline{\mathsf{ct}}$. Since the message $\overline{m}$ is selected by the adversary, it is then able to recover the original message m_β. This breaks the indistinguishability requirement of the IND-CCA2 security model, implying that Yang et al.'s LC-MUME scheme [8] does not achieve IND-CCA2 security. Finally, we note that our attack requires only a single query to the decryption oracle, combined with a few lightweight computations, including hashing and XOR operations.

B. FLAWS IN THE DESIGN AND SECURITY ANALYSIS OF THE LC-MUME SCHEME IN [8]

We analyze the design and security proof flaws in the target LC-MUME scheme proposed by Yang et al. [8].

1) FLAWS IN THE DESIGN

We observe that, in Yang et al.'s construction [8], the fourth component C_4 of the ciphertext is computed as follows:

$$C_4 = \mathcal{H}_3(C_1, C_2, C_3, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}).$$

Notably, all inputs to the hash function $\mathcal{H}_3$ are public, because they are entirely composed of ciphertext components. As a result, an adversary can modify at least one component of the challenge ciphertext and subsequently recompute a valid C_4 using the updated values and the public hash function $\mathcal{H}_3$. This allows the manipulated ciphertext to pass the validity check enforced by Eq. (1).

2) FLAWS IN THE SECURITY PROOF

The critical flaw in the proof of [8, Theorem 2] lies in the incorrect construction of the decryption oracle $\mathcal{O}^{\mathbf{Dec}}$. In their security analysis, the decryption oracle $\mathcal{O}^{\mathbf{Dec}}$ is designed to abort the game whenever $\mathrm{id}_i = \mathrm{id}_\Phi$, where id_i is the receiver identity in the ciphertext queried to $\mathcal{O}^{\mathbf{Dec}}$, and id_Φ is the challenge receiver identity in the target receiver set $\mathcal{S}$. The authors of [8] claimed that the probability of this event, denoted as E_1 in their proof, is negligible in the security parameter. However, as demonstrated by our attacks, the event E_1 can occur with non-negligible probability, depending on the adversary's strategy, particularly during Phase 2 of the IND-CCA2 security game. As a result, their security reduction presented in [8] fails to correctly guarantee the IND-CCA2 security of their LU-MUME scheme.

IV. OUR MODIFICATION

We now propose a potential solution to address the identified vulnerability in Yang et al.'s LC-MUME scheme [8]. Our strategy is to incorporate both the message m and the encryption randomness r into the input of the hash function $\mathcal{H}_3$, following a similar approach to that used in [9]. In this revised construction, the decryption algorithm must recover the value of r to check the validity of the C_4 component, which is computed using $\mathcal{H}_3$. To facilitate this, we modify the C_3 component to include r as well as m.

To this end, we exploit new (collision-resistant) hash functions

$$\mathcal{H}_2' : \mathbb{G}^2 \times (\mathbb{Z}_q^*)^2 \to \{0,1\}^{l_1+l_2}, \text{ and}$$
$$\mathcal{H}_3' : \mathbb{G}^2 \times \{0,1\}^{l_1+l_2} \times (\mathbb{Z}_q^*)^{2n} \times \{0,1\}^{l_1+l_2} \to \{0,1\}^{\mathrm{len}},$$

instead of $\mathcal{H}_2$ and $\mathcal{H}_3$, respectively, where all elements in $\mathbb{Z}_q^*$ can be represented in l_2 bits and len is an integer of appropriate size. We note that $\mathcal{H}_2'$ and $\mathcal{H}_3'$ are obtained by adjusting the image of $\mathcal{H}_2$ and the domain of $\mathcal{H}_3$ from $\{0,1\}^l$ and $\mathbb{G}^2 \times \{0,1\}^l \times (\mathbb{Z}_q^*)^{2n}$, respectively, where $l > l_1$ is an integer of proper size.

We next modify the encryption algorithm, denoted by $\mathbf{Enc}'$, as follows. The modification introduces two key differences compared to the original construction:

- Use $\mathcal{H}_2'$ and $\mathcal{H}_3'$, instead of $\mathcal{H}_2$ and $\mathcal{H}_3$, respectively.
- Modify components of C_3 and C_4: The encryption algorithm computes C_3 and C_4 as

$$C_3 = \mathcal{H}_2'(C_1, C_2, d_1, d_2) \oplus (m \| r) \text{ and}$$
$$C_4 = \mathcal{H}_3'(C_1, C_2, C_3, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}, m, r).$$

The ciphertext of our modification consists of

$$\mathsf{ct} = (C_1, C_2, C_3, C_4, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}).$$

Then, we adjust the decryption algorithm accordingly to first recover both m and r, and subsequently check the validity of the ciphertext. Refer to the following for further details.

1) DESCRIPTION OF OUR MODIFICATION

The formal description of our modification is given below. The **Setup**, **SKGen**, **RKGen** algorithms of our modification are almost the same as those of Yang et al.'s original scheme, except that the hash functions $\mathcal{H}_2$ and $\mathcal{H}_3$ are replaced with

$$\mathcal{H}_2' : \mathbb{G}^2 \times (\mathbb{Z}_q^*)^2 \to \{0,1\}^{l_1+l_2}, \text{ and}$$
$$\mathcal{H}_3' : \mathbb{G}^2 \times \{0,1\}^{l_1+l_2} \times (\mathbb{Z}_q^*)^{2n} \times \{0,1\}^{l_1+l_2} \to \{0,1\}^{\mathrm{len}},$$

where all elements in $\mathbb{Z}_q^*$ can be represented in l_2 bits and len is an integer of appropriate size.

- $\mathbf{Enc}'$: Given the public parameter pp, an encryption key $\mathrm{ek}_{\mathrm{id}_0} = (\mathrm{ek}^1_{\mathrm{id}_0}, \mathrm{ek}^2_{\mathrm{id}_0})$ associated with identity id_0, a target identity set $\mathcal{S}$ of size n, the public keys $\mathsf{dpk} = \{\mathrm{dpk}_{\mathrm{id}_i} = (\mathrm{dpk}^1_{\mathrm{id}_i}, \mathrm{dpk}^2_{\mathrm{id}_i}) | \mathrm{id}_i \in \mathcal{S}\}$ corresponding to the decryption private key, and a plaintext $m \in \{0,1\}^{l_1}$, the encryption algorithm proceeds as follows:
  1) Sample random integers $r, s, d_1, d_2 \in \mathbb{Z}_q^*$, and compute $C_1 = rP$ and $C_2 = sP$.
  2) For each $\mathrm{id}_i \in \mathcal{S}$, compute the intermediate values
$$\tau_i = \mathrm{ek}^1_{\mathrm{id}_0} \mathrm{dpk}^1_{\mathrm{id}_i}$$
$$V_{\mathrm{id}_i} = \mathcal{H}(r\mathcal{H}_4(\mathrm{id}_0, \mathrm{id}_i, \tau_i)(\mathrm{dpk}^2_{\mathrm{id}_i} + \mathcal{H}_1(\mathrm{id}_i, \mathrm{dpk}^2_{\mathrm{id}_i})\mathcal{P}')),$$
$$Z_{\mathrm{id}_i} = (s + \mathrm{ek}^2_{\mathrm{id}_0} + \mathrm{ek}^1_{\mathrm{id}_0}\mathcal{H}_4(\mathrm{id}_0, \mathrm{id}_i, \tau_i))\mathrm{dpk}^1_{\mathrm{id}_i}.$$
  3) Construct two degree-n polynomials $f(x)$ and $g(y)$ defined as follows:
$$f(x) = \prod_{\mathrm{id}_k \in \mathcal{S}} (x - V_{\mathrm{id}_k}) + d_1 = \sum_{k=0}^{n-1} a_k x^k + x^n,$$
$$g(y) = \prod_{\mathrm{id}_k \in \mathcal{S}} (y - Z_{\mathrm{id}_k}) + d_2 = \sum_{k=0}^{n-1} b_k y^k + y^n$$
over $\mathbb{Z}_q$ where $a_k, b_k \in \mathbb{Z}_q$ denote the coefficients corresponding to x^k and y^k, respectively.
  4) Compute C_3 and C_4 as follows:
$$C_3 = \mathcal{H}_2'(C_1, C_2, d_1, d_2) \oplus (m \| r)$$
$$C_4 = \mathcal{H}_3'(C_1, C_2, C_3, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}, m, r).$$
  5) Output a ciphertext
$$\mathsf{ct} = (C_1, C_2, C_3, C_4, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}).$$
- $\mathbf{Dec}'$: Given the public parameter pp, a decryption private key $\mathrm{dk}_{\mathrm{id}_i} = (\mathrm{dk}^1_{\mathrm{id}_i}, \mathrm{dk}^2_{\mathrm{id}_i})$, a target sender identity $\mathsf{snd} = \mathrm{id}_0$ along with the corresponding encryption public key $\mathrm{epk}_{\mathrm{id}_0} = (\mathrm{epk}^1_{\mathrm{id}_0}, \mathrm{epk}^2_{\mathrm{id}_0})$, and a ciphertext $\mathsf{ct} = (C_1, C_2, C_3, C_4, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1})$, the decryption algorithm proceeds as follows:
  1) Compute
$$\tau_i = \mathrm{dk}^1_{\mathrm{id}_i} \mathrm{epk}^1_{\mathrm{id}_0},$$
$$V_{\mathrm{id}_i} = \mathcal{H}(\mathrm{dk}^2_{\mathrm{id}_i}\mathcal{H}_4(\mathrm{id}_0, \mathrm{id}_i, \tau_i)C_1),$$
$$Z_{\mathrm{id}_i} = \mathrm{dk}^1_{\mathrm{id}_i}(C_2 + \mathrm{epk}^2_{\mathrm{id}_0} + \mathcal{H}_1(\mathrm{id}_0, \mathrm{ek}^2_{\mathrm{id}_0})\mathcal{P}' + T)$$

TABLE 1. Efficiency comparison of the original and modified schemes. The costs are presented in terms of major cryptographic operations, where *n* is the number of receivers and λ is the security parameter. SM denotes an elliptic curve scalar multiplication.

Category	Operation	Yang et al. (Original)	Proposed (Modified)	Difference
Encryption	SM	$2n+2$	$2n+2$	None
	Hashing	$\mathcal{H}_2, \mathcal{H}_3$ computations	$\mathcal{H}'_2, \mathcal{H}'_3$ computations	Negligible[1]
Decryption	SM	3	3	None
	Hashing	$\mathcal{H}_2, \mathcal{H}_3$ verification	$\mathcal{H}'_2, \mathcal{H}'_3$ verification	Negligible
Ciphertext Size		$(4n+12)\lambda$ bits	$(4n+14)\lambda$ bits	$+2\lambda$ bits

where $T = \mathcal{H}_4(\mathrm{id}_0, \mathrm{id}_i, \tau_i)\mathrm{epk}^1_{\mathrm{id}_0}$. Using the coefficients $a_0, \ldots, a_{n-1}, b_0, \ldots b_{n-1}$, reconstruct the polynomials

$$f(x) = \sum_{k=0}^{n-1} a_k x^k + x^n \quad \text{and} \quad g(y) = \sum_{k=0}^{n-1} b_k y^k + y^n$$

and evaluate them at the computed values V_{id_i} and Z_{id_i}, respectively, to obtain

$$d_1 = f(V_{\mathrm{id}_i}) \quad \text{and} \quad d_2 = g(Z_{\mathrm{id}_i}).$$

2) Compute

$$m \| r = C_3 \oplus \mathcal{H}'_2(C_1, C_2, d_1, d_2).$$

3) Check if the following equality holds:

$$C_4 = \mathcal{H}'_3(C_1, C_2, C_3, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}, m, r) \quad (4)$$

Return m if the check passes; otherwise, return $\perp$.

2) ANALYSIS OF OUR MODIFICATION

The correctness of the above modification is straightforward. It is also expected to be secure against our attack. More precisely, when attempting to manipulate the challenge ciphertext in a similar manner to our attack in the IND-CCA2 security game, the adversary faces the following obstacles:

- If C_1 and C_2 remain unchanged, C_3 must be modified. In this case, the adversary must know r used in generating the challenge ciphertext to recalculate C_4 so that the manipulated ciphertext passes Eq. (4). Here, the randomness r is no longer just a component of encryption; it acts as a secret, single-use authenticator that uniquely binds the message m to the integrity check component C_4 of the ciphertext.
- Otherwise, if at least one of C_1 or C_2 should be changed, the adversary must know appropriate d_1, d_2, and r used in generating the challenge ciphertext to correctly recalculate both C_3 and C_4 so that the manipulated ciphertext passes Eq. (4).

However, it is likely computationally infeasible for the adversary to obtain r or the required values to compute d_1, d_2 and r under the computational Diffie-Hellman (CDH) assumption and the collision-resistance of employed hash functions in the random oracle model. Specifically, the hash function $\mathcal{H}'_3$ creates an unforgeable cryptographic commitment to the entire set of inputs including the secret message m and randomness r. An adversary cannot compute a valid C'_4 for a manipulated ciphertext without knowing these secret inputs, a task made infeasible by the pre-image resistance property of the hash function. Therefore, this modification appears to effectively thwarts our attack.

We revise the formal proof in the original paper to incorporate our modification, ensuring that it achieves IND-CCA2 security. From a formal security perspective, the security of our modification can be shown by properly adjusting the decryption oracle $\mathcal{O}^{\mathbf{Dec}}$ in the original security proof as follows: If $\mathrm{id}_i \neq \mathrm{id}_\Phi$, the oracle behaves as in the original proof. Otherwise, it searches the list $L_{\mathcal{H}'_3}$ for the tuple $(C_1, C_2, C_3, a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1})$ to retrieve the corresponding message m and randomness r. Here, $L_{\mathcal{H}'_3}$ is a list initialized as an empty set, which stores query-response pairs for the hash function $\mathcal{H}'_3$. The oracle then verifies the validity of the remaining parts of the input ciphertext using m and r. If the input ciphertext passes all validity checks, it returns m. Otherwise, it selects another m and r, and repeats the above process until the validity check succeeds. Therefore, $\mathcal{O}^{\mathbf{Dec}}$ can respond correctly to all ciphertexts, except the challenge ciphertext, and our modification achieves IND-CCA2 security.

3) EFFICIENCY COMPARISON

Our proposed modification is simple and incurs only minimal overhead. The primary changes apply to the C_3 and C_4 components of the ciphertext. For C_3, we enhance the XOR computation by incorporating the encryption randomness r, which is a 2λ-bit value, where λ denotes the security parameter. As a result, the size of C_3 increases by 2λ bits, and total ciphertext size becomes $(4n + 14)\lambda$, compared to $(4n + 12)\lambda$ in the original construction, assuming a target identity set $\mathcal{S}$ of size n. For C_4, the input to $\mathcal{H}'_3$ is extended to include the message $m \in \{0,1\}^{l_1}$ and the encryption randomness $r \in \mathbb{Z}^*_q$. However, this additional input contributes only a small increase in size, since the original domain of $\mathcal{H}_3$–namely, $\mathbb{G}^2 \times \{0,1\}^l \times (\mathbb{Z}^*_q)^{2n}$–is already substantially larger than m and r. Furthermore, since hash evaluations are considerably less expensive than elliptic curve operations, the computational impact of our modification is also negligible. The summary of efficiency comparison is given in Table 1.

[1]The modification uses hash functions $\mathcal{H}'_2$ and $\mathcal{H}'_3$ which process slightly larger inputs. As stated in the paper, since hash evaluations are considerably less expensive than elliptic curve operations, the impact of this change on the overall computation time is negligible.

V. CONCLUSION

This paper identifies a critical security flaw in Yang et al.'s multi-user matchmaking encryption scheme [8], demonstrating that it fails to achieve the claimed IND-CCA2 security. At the heart of the vulnerability is a design oversight in which the ciphertext component responsible for integrity verification, namely C_4, is computed solely from public known values. This flaw results in ciphertext malleability, allowing an adversary to generate a modified ciphertext that passes integrity checks and subsequently recover the original plaintext through a single decryption oracle query.

To mitigate this weakness, we propose a security-enhanced variant of the scheme that strengthens ciphertext integrity by embedding both the secret plaintext message m and the encryption randomness r into the computation of C_3 and C_4. This modification makes it computationally infeasible for an adversary to create a valid manipulated ciphertext without knowing these secret values. The proposed enhancement successfully restores IND-CCA2 security while preserving the scheme's efficiency, incurring only minimal computational and communication overhead.

This work highlights how subtle design oversights in cryptographic constructions can undermine overall system security. Through a concrete case study, it demonstrates how ciphertext malleability can be exploited to break IND-CCA2 security and presents an effective fix that restores the intended security guarantees. As such, it offers valuable guidance for the secure design of matchmaking encryption schemes and their variants. Moreover, since our modification still relies on the random oracle model, we leave the construction of a multi-user matchmaking encryption scheme in the standard model as an open problem for future research.

REFERENCES

[1] G. Ateniese, D. Francati, D. Nuñez, and D. Venturi, "Match me if you can: Matchmaking encryption and its applications," in *Proc. Adv. Cryptol.*, in Lecture Notes in Computer Science, vol. 11693, A. Boldyreva and D. Micciancio, Eds., Cham, Switzerland: Springer, 2019, pp. 701–731.

[2] B. Chen, T. Xiang, M. Ma, D. He, and X. Liao, "CL-ME: Efficient certificateless matchmaking encryption for Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 15010–15023, Oct. 2021.

[3] D. Francati, A. Guidi, L. Russo, and D. Venturi, "Identity-based matchmaking encryption without random oracles," in *Proc. Prog. Cryptol.*, A. Adhikari, R. Küsters, and B. Preneel, Eds., Cham, Switzerland: Springer, 2021, pp. 415–435.

[4] A. Wu, W. Luo, J. Weng, A. Yang, and J. Wen, "Fuzzy identity-based matchmaking encryption and its application," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 5592–5607, 2023.

[5] Z. Jiang, X. Wang, K. Zhang, J. Gong, J. Chen, and H. Qian, "Revocable identity-based matchmaking encryption in the standard model," *IET Inf. Secur.*, vol. 17, no. 4, pp. 567–581, Jul. 2023.

[6] J. Sun, G. Xu, T. Zhang, X. Yang, M. Alazab, and R. H. Deng, "Privacy-aware and security-enhanced efficient matchmaking encryption," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4345–4360, 2023.

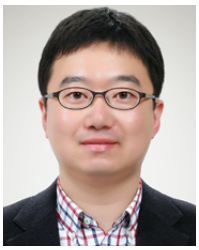
[7] Z. Yan, H. Qu, X. Zhang, J.-L. Xu, and X.-J. Lin, "Identity-based proxy matchmaking encryption for cloud-based anonymous messaging systems," *J. Syst. Archit.*, vol. 142, Sep. 2023, Art. no. 102950.

[8] N. Yang, C. Tang, and D. He, "A lightweight certificateless multi-user matchmaking encryption for mobile devices: Enhancing security and performance," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 251–264, 2024.

[9] H. Lin, F. Gao, H. Zhang, Z. Jin, W. Li, and Q. Wen, "Public key encryption with equality test supporting flexible designated authorization in cloud storage," *IEEE Syst. J.*, vol. 16, no. 1, pp. 1460–1470, Mar. 2022.



JISEUNG KIM received the B.S. degree in mathematics from Chonnam National University, in 2014, and the Ph.D. degree in mathematical sciences from Seoul National University, Seoul, Republic of Korea, in 2020. From 2020 to 2022, he was a Research Scientist with the School of Computational Science, Korea Institute for Advanced Study. He is currently an Assistant Professor with the Department of Computer Science and Artificial Intelligence, Jeonbuk National University, Jeonju, Republic of Korea. His research interests include the mathematical analysis of lattice-based cryptography and hard problems.



HYUNG TAE LEE received the B.Sc., M.Sc., and Ph.D. degrees in mathematics from Seoul National University, Republic of Korea, in 2006, 2008, and 2013, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, Chung-Ang University, Republic of Korea. Prior to that, he was a Research Fellow with Nanyang Technological University, Singapore, and then was an Assistant Professor with Jeonbuk National University, Republic of Korea. His research interests include computational number theory, cryptography, and information security.