Insecurity of an IBEET Scheme and an ABEET Scheme

YONGJIAN LIAO, (Member, IEEE), HONGJIE CHEN, WEN HUANG, RAMADAN MOHAMMED, HONGTAO PAN, AND SHIJIE ZHOU, (Member, IEEE)

School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Yongjian Liao (liaoyj@uestc.edu.cn)

This work was supported in part by the Sichuan Science and Technology Program under Grant 2018GZ0180, Grant 2018GZ0085, Grant 2017GZDZX0002, and Grant 2017GZDZX0001, and in part by the National Natural Science Foundation of China under Grant 61472066.

ABSTRACT Cloud computing is a novel pattern, which can allow users to outsource their data to the cloud servers in order to save the resource of clients. For the sake of protecting the clients' privacy, the clients' data are stored or computed in the encrypted data. This leads to a new issue: how to quickly search these encrypted data. One of the methods to solve the problem is to construct public-key encryption schemes with the equality test. In this paper, we point out that an identity-based encryption scheme with the equality test and a ciphertext-policy attribute-based encryption scheme with equality test is not secure on their security models, respectively.

INDEX TERMS Identity based encryption, attribute based encryption, equality test, insecurity, cloud computing.

I. INTRODUCTION

In cloud computing environment, users are allowed to outsource their data to cloud severs. In order to protect the privacy of data of users, these outsourced data have to be stored in an encrypted form. However, in order to efficiently extract some statistical information of the data for users in the future, it is necessary to search some information from the encrypted data in the cloud firstly.

This novel application brings many new security issues, such as auditing [1], outsourcing computation [2], [3], outsourcing verification [4] and encrypted data searching [5]. Boneh *et al.* [5] introduced a new concept — public key encryption with keyword search (PKEKS), which can search a keyword over the encrypted data but cannot decrypt it. Later, Yang *et al.* [6] put forward another new concept — public key encryption with equality test (PKEET), which combines the public key encryption (PKE) and searchable encryption (SE). The PKEET cannot only decrypt the encrypted keyword, but also can check if ciphertexts are encryptions of the same unknown keyword even if it is possible to use different public keys. Tang [7] proposed a PKEET with fine-grained authorization scheme (PKEET-FG), an extension of PKEET-FG [8] and all-or-nothing PKEET (AON-PKEET) [9] to improve the scheme. Ma *et al.* [10] presented a PKEET supporting flexible authorization (PKEET-FA). In their scheme there are 4 types of flexible authorizations. In order to simplify the certificate management of PKEET, Ma [11] presented an identity-based encryption with equality test (IBEET), and showed that the scheme was one-way secure under chosen ciphertext attack (OW-CCA). Also, Lee *et al.* [12] presented semi-generic construction of IBEET scheme and PKEET scheme, but their constructions need to use the encryption algorithm twice and a one-time signature, which aren't efficient. Zhang and Xu [13] and Zhang *et al.* [14] proposed two schemes from lattices, which can be considered secure under quantum computing attacks. In order to make the scheme more flexible, Zhu *et al.* [15] and Wang *et al.* [16] proposed a key policy attribute based encryption scheme with equality test (KP-ABEET) and ciphertext policy attribute based encryption scheme with equality test (CP-ABEET) respectively and showed their corresponding security. However, Liao *et al.* [17] showed that the KP-ABEET scheme [15] wasn't secure on their security model.

In this article, we analyze the security of IBEET and CP-ABEET as follows. We firstly prove that the IBEET scheme proposed by Ma isn't one-way under chosen ciphertext attack and then we set forth the reason of insecurity and give some idea to improve the scheme. Next we prove that the CP-ABEET scheme proposed by Wang *et al.* isn't

indistinguishable against chosen plaintext attack in the standard model.

The rest of this paper is organized as follows. In section II, we recall basic concepts which will be used in the paper. We then recall an IBEET scheme proposed by Ma and show that the scheme isn't secure based on their security models and improve the scheme in section III. In section IV we recall a CP-ABEET scheme proposed by Wang *et al.* and show that the scheme isn't IND-CPA secure in the standard model. Finally, we conclude the paper in section V.

II. PRELIMINARY

Here, we first recall some basic mathematical knowledge which will be used.

A. BILINEAR PAIRING

Let $\mathbb{G}$ and $\mathbb{G}_{\mathrm{T}}$ be two multiplicative groups which have the same prime order q, $\mathbb{Z}_q^*$ be the multiplicative group of the finite field $\mathbb{F}_q$. A bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathrm{T}}$ [18], which satisfies the following three properties:

- Bilinearity: For any $u, v, w \in \mathbb{G}$,

$$e(u, vw) = e(u, v)e(u, w), \quad \text{and}$$
$$e(uv, w) = e(u, w)e(v, w).$$

- Non-degeneracy: There are elements $g_1, g_2 \in \mathbb{G}$, such that $e(g_1, g_2) \neq 1_{G_{\mathrm{T}}}$, where $1_{G_{\mathrm{T}}}$ is the identity element of $\mathbb{G}_{\mathrm{T}}$.
- Computability: For any elements $g_1, g_2 \in \mathbb{G}$, there is an efficient algorithm to compute $e(g_1, g_2)$.

Definition 1: Bilinear Diffie-Hellman problem (BDH problem). Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be the groups of the same prime order q above. Given $(g, g^c, g^b, g^a) \in \mathbb{G}^4$ for some $a, b, c \in \mathbb{Z}_q^*$, to compute $e(g, g)^{abc}$. Where g isn't the identity element of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\mathrm{T}}$.

Definition 2: Twin-decision bilinear Diffie-Hellman problem (t-DBDH problem). Given two distributions

$$D_0 = \{g, g^a, g^b, g^c, g^u, g^v, e(g, g)^{abc}, e(g, g)^{auv} : a, b, c, u, v \in \mathbb{Z}_q\}$$
$$D_1 = \{g, g^a, g^b, g^c, g^u, g^v, e(g, g)^{d}, e(g, g)^{w} : a, b, c, d, u, v, w \in \mathbb{Z}_q\}$$

to decide whether $abc \equiv d \bmod q$ and $auv \equiv w \bmod q$ hold or not.

B. MODEL OF IBEET AND ABEET

An IBEET and ABEET include three entities, the key generator center (KGC), users and the cloud server, which are described in FIGURE 1. The KGC generates the private key of a user's identity in IBEET and sets of attributes in ABEET, respectively. The users create their trapdoors by using their private keys and ciphertexts. The cloud server stores the users' data (ciphertexts) and runs the test algorithm when it receives the corresponding trapdoors. The users receive their private keys over secure channels and the cloud server gets the ciphertexts and trapdoors over open channels which can be eavesdropped by adversaries.

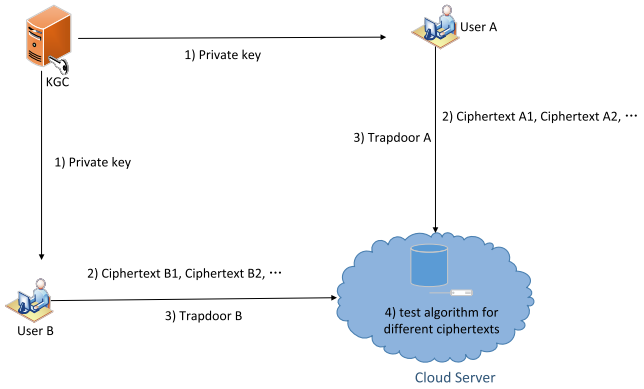

FIGURE 1. Model of IBEET and ABEET.

III. INSECURITY OF AN IBEET SCHEME

A. SYNTAX OF IBEET

An IBEET scheme [11] includes six algorithms: Setup, Extract, Enc, Dec, Trapdoor and Test. Let $\mathbb{M}$ and $\mathbb{C}$ be its plaintext space and ciphertext space, respectively.

- **Setup**(k): On input a security parameter k, the algorithm outputs public system parameters K and a master key msk.
- **Extract**(msk, ID): On input msk and an arbitrary identity $ID \in \{0, 1\}^*$, the algorithm outputs a private key sk_{ID} for an identity ID.
- **Enc**(ID, M): On input an identity $ID \in \{0, 1\}^*$ and a plaintext $M \in \mathbb{M}$, the algorithm outputs a ciphertext $C \in \mathbb{C}$.
- **Dec**(sk_{ID}, C): On input a ciphertext $C \in \mathbb{C}$ and a private key sk_{ID}, the algorithm outputs a plaintext $M \in \mathbb{M}$.
- **Trapdoor**(sk_{ID_A}, C): On input the private key sk_{ID_A} of identity ID_A of a user A and a ciphertext $C \in \mathbb{C}$ encrypted some plaintext by using ID_A, the algorithm outputs a trapdoor td_A. If C is empty string(ciphertext), then that means all ciphertexts correspond to the same trapdoor td_A.
- **Test**(C_A, td_A, C_B, td_B) : On input a ciphertext $C_A \in \mathbb{C}$, a trapdoor td_A for an identity ID_A, and a ciphertext $C_B \in \mathbb{C}$, a trapdoor td_B for an identity ID_B, the algorithm outputs "1" if C_A and C_B are generated by the same plaintext; Otherwise it outputs "0".

B. SECURITY MODEL OF IBEET

We recall the definition of the security concept of one-way against chosen ciphertext security(OW-ID-CCA) for IBEET scheme [11].

GAME 1

- Setup: On input a security parameter k, the challenger $\mathcal{C}$ produces the system parameters K by running the Setup algorithm. Then $\mathcal{C}$ sends K to the adversary and keeps the master key msk by itself.

- The Phase 1
  - Private key queries. The challenger $\mathcal{C}$ produces a private key sk_i of an identity ID_i by running the Extract algorithm. Then $\mathcal{C}$ sends the private key sk_i of identity ID_i to adversary $\mathcal{A}$.
  - Trapdoor queries TD_i. At any time, in order to obtain a trapdoor of an identity ID_i, on input identity ID_i, the adversary $\mathcal{A}$ can query trapdoor oracle. The challenger gets trapdoor td_i by running the above private key queries on ID_i, and then sends the trapdoor td_i to $\mathcal{A}$.
  - Decryption queries (ID_i, C_i). The challenger decrypts the ciphertext C_i by running the decryption oracle, and then sends the M_i, which is an output of the decryption oracle, to adversary $\mathcal{A}$.
- Challenge: Firstly the adversary $\mathcal{A}$ decides to submit an challenge identity ID^* which she/he selects. The only restricted condition is that ID^* didn't appear in the private key queries in the phase 1, however ID^* may be in the decryption queries $(ID^*, \cdot)$ or in the trapdoor queries. Then the challenger $\mathcal{C}$ randomly selects a plaintext $M^* \in \mathcal{M}$ and finally sends the challenge ciphertext $C^* = \text{Enc}(ID^*, M^*)$ to $\mathcal{A}$.
- The Phase 2.
  - Private key queries. If $ID_i \neq ID^*$, then the challenger $\mathcal{C}$ responds it as that in the phase 1.
  - Trapdoor queries TD_i. For any identity, the challenger $\mathcal{C}$ responds it as that in the phase 1.
  - Decryption queries. If any ciphertext $(ID_i, C_i) \neq (ID^*, C^*)$, then the challenger $\mathcal{C}$ responds it as that in the phase 1.
- Guess: $\mathcal{A}$ submits a guess $M' \in \mathcal{M}$.

We call an adversary $\mathcal{A}$ to be a OW-ID-CCA adversary in the above game [11]. And the advantage of the OW-ID-CCA adversary is the probability that the adversary $\mathcal{A}$ wins the game, that is

$$Adv^{OW-ID-CCA}_{\text{IBEET},\mathcal{A}}(k) = \Pr[M = M'].$$

Definition 3: We call an IBEET scheme to be OW-ID-CCA secure, if for all OW-ID-CCA adversaries, $Adv^{OW-ID-CCA}_{\text{IBEET},\mathcal{A}}(k)$ is negligible in the security parameter k.

C. RECALL THE Ma's IBEET SCHEME AND ITS SECURITY ANALYSIS

Before we analyse the Ma's IBEET scheme [11], we firstly recall it.

1) RECALL THE Ma's IBEET SCHEME

The construction of their scheme is as follows.

- Setup: On input a security parameter k, it works as follows:
  - Produce some public parameters: two multiplicative groups $\mathbb{G}$, $\mathbb{G}_{\text{T}}$ of prime order p, and an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_{\text{T}}$ with a random generator $g \in \mathbb{G}$. Randomly select three cryptographic hash functions: $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G}_{\text{T}} \to \mathbb{G}$, $H_3 : \mathbb{G}_{\text{T}} \to \{0,1\}^{l_1+l_2}$, where l_1 and l_2 are length of elements in $\mathbb{G}$ and $\mathbb{Z}_p$, respectively.
  - Select random elements $s', s \in \mathbb{Z}_p$ and let $g_1 = g^{s'}$ and $g_2 = g^s$. Here let $\mathbb{M} \subset \mathbb{G}$ be the message space and $\mathbb{C} \subset \mathbb{G}^4 \times \{0,1\}^{l_1+l_2}$ be the ciphertext space. The public system parameters are $K_{\text{IBEET}} = (p, \mathbb{G}, \mathbb{G}_{\text{T}}, e, g, g_1, g_2, H_1, H_2, H_3)$. The master key msk is $(s', s) \in \mathbb{Z}_p^2$.
- Extract: On input an identity $ID \in \{0,1\}^*$, it works as follows:
  - Firstly calculate $h_{ID} = H_1(ID) \in \mathbb{G}$, and then calculate $sk_{ID} = (h_{ID}^{s'}, h_{ID}^{s})$ as the private key, where (s', s) is the master key.
- Trapdoor: On input an identity $ID \in \{0,1\}^*$, it works as follows:
  - Firstly calculate $h_{ID} = H_1(ID) \in \mathbb{G}$, then set the trapdoor $td_{ID} = h_{ID}^{s'}$, which is the first element of sk_{ID}.
- Enc: On input a message $M \in \mathbb{G}$ and an identity ID, it works as follows:
  - Firstly calculate $h_{ID} = H_1(ID) \in \mathbb{G}$, and randomly pick three elements $r_1, r_2, r_3 \in \mathbb{Z}_p$;
  - Set the ciphertext to be $C = (C_1, C_2, C_3, C_4, C_5)$, where

$$C_1 = g^{r_1}, \quad C_2 = g^{r_2}, \; C_4 = g^{r_3},$$
$$C_3 = M^{r_1} H_2(U_1^{r_2}), \quad C_5 = (M||r_1) \oplus H_3(U_2^{r_3}),$$

    and where

$$U_1 = e(h_{ID}, g_1) \in \mathbb{G}_{\text{T}}, \quad U_2 = e(h_{ID}, g_2) \in \mathbb{G}_{\text{T}}.$$

- Dec(C, sk_{ID}): On input ciphertext C and the private key $sk_{ID} = (h_{ID}^{s'}, h_{ID}^{s})$, where $C = (C_1, C_2, C_3, C_4, C_5) \in \mathbb{C}$ is a ciphertext encrypted by using the identity ID, the algorithm firstly calculates

$$C_5 \oplus H_3(e(h_{ID}^{s}, C_4)) = M||r_1,$$

  and then it outputs M if the following equalities hold.

$$C_1 = g^{r_1}, \frac{C_3}{M^{r_1}} = H_2(e(h_{ID}^{s'}, C_2))$$

  Otherwise, it outputs "$\perp$".
- Test$(C_A, td_{ID_A}, C_B, td_{ID_B})$: On input ciphertexts C_A, C_B and corresponding trapdoors td_{ID_A}, td_{ID_B} respectively, to determine whether plaintexts M_A and M_B are equal or not, where

$$C_A = (C_{A,1}, C_{A,2}, C_{A,3}, C_{A,4}, C_{A,5}) = \text{Enc}(M_A, ID_A)$$

and

$$C_B = (C_{B,1}, C_{B,2}, C_{B,3}, C_{B,4}, C_{A,5}) = \text{Enc}(M_B, ID_B),$$

$td_{ID_A} = h_{ID_A}^{s'}$ and $td_{ID_B} = h_{ID_B}^{s'}$. It firstly computes:

$$X_A = \frac{C_{A,3}}{H_2(e(h_{ID_A}^{s'}, C_{A,2}))}, \quad X_B = \frac{C_{B,3}}{H_2(e(h_{ID_B}^{s'}, C_{B,2}))},$$

and then it outputs "1" if the equation

$$e(C_{A,1}, X_B) = e(C_{B,1}, X_A)$$

holds; otherwise it outputs "0".

2) INSECURITY OF Ma's SCHEME

Next, we will show that the IBEET scheme doesn't satisfy the above OW-ID-CCA security, which was firstly defined in paper [11]. From the definition of the OW-ID-CCA attack, at any time any adversary can have access to the trapdoor oracle to obtain the trapdoor of any identity including the challenge identity ID^*. That is to say, at least the adversary can get $td_{ID^*} = h_{ID^*}^{s'}$ before guess stage in the attack game. When the adversary knows the challenge identity ID^* and the challenge ciphertext $C^* = (C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)$, it conducts as follows.

- When the adversary obtains the challenge (ID^*, C^*), she/he checks whether the challenge identity ID^* is listed in trapdoor queries or not. It can get $td_{ID^*} = h_{ID^*}^{s'}$ by having access to the trapdoor queries before the guess stage if it is not listed in previous trapdoor queries (A better and easier way to do it is to pick the challenge identity from the list of trapdoor queries in the phase 1. Because the challenge identity is chosen by the adversary.).
- Secondly, the adversary calculates

$$V^* = \frac{C_3^*}{H_2(e(h_{ID^*}^{s'}, C_2^*))},$$

and randomly selects $r \in \mathbb{Z}_p^*$ to set $C_2 = (C_2^*)^r$. The adversary checks whether C_2 is a part of a ciphertext having been on the list of decryption queries and encrypted by the identity ID^*. If it is on the list of decryption queries, then repeats above step (renew the random number r); otherwise, the adversary calculates

$$C_3 = V^* H_2(e(h_{ID^*}^{s'}, (C_2^*)^r)).$$

- Thirdly, the adversary modifies the challenge ciphertext C^* and continues to perform the chosen ciphertext attack as follows. The adversary firstly sets

$$C_1' = C_1^*, \quad C_2' = C_2, \; C_3' = C_3,$$
$$C_4' = C_4^*, \quad C_5' = C_5^*,$$

and then submits $C' = (C_1', C_2', C_3', C_4', C_5')$ $C' = (C_1', C_2', C_3', C_4', C_5')$ with the identity ID^* to the decryption queries as a ciphertext. The challenge responds M' if the "ciphertext" C' is a valid ciphertext; otherwise it responds "$\perp$".
- Finally, the adversary outputs $M^*(= M')$ as a plaintext of the challenge ciphertext C^* and the challenge identity ID^*.

For above attack, the ciphertext C' doesn't equal C^*, because

$$C_2' = C_2 = (C_2^*)^r \neq C_2^*.$$

When the decryption oracle receives (ID^*, C') as a decryption query, it computes as follows.

$$C_5' \oplus H_3(e(h_{ID^*}^s, C_4')) = M' || r_1'.$$

Since C^* is a valid ciphertext and $C_1' = C_1^*$, $C_4' = C_4^*$, $C_5' = C_5^*$. We have $r_1' = r_1^*$ and $M' = M^*$. And it is obvious that the following two equalities hold.

$$C_1' = g^{r_1'} \quad \text{and}$$
$$\frac{C_3'}{M'^{r_1'}} = \frac{\frac{C_3^*}{H_2(e(h_{ID^*}^{s'}, C_2^*))} H_2(e(h_{ID^*}^{s'}, (C_2^*)^r))}{M'^{r_1'}}$$
$$= H_2(e(h_{ID^*}^{s'}, C_2')).$$

Thus, the above adversary can successfully get $M'(= M^*)$, which is the plaintext of the challenge ciphertext C^*.

Therefore, the IBEET scheme is not OW-ID-CCA secure.

D. REASON OF INSECURITY

Now, we analyse the IBEET scheme and find out the reason why it is insecure as follows. Firstly, from the construction of the encryption algorithm in the IBEET scheme, it is easy to find out that it encrypts (M, r_1) and M^{r_1} respectively by independently using Boneh and Franklin's identity-based encryption(IBE) scheme [19]. In detail, an encrypter picks a random number $r_3 \in \mathbb{Z}_p$ to encrypt (M, r_1) and sets $C_4 = g^{r_3}$, $C_5 = (M||r_1) \oplus H_3(U_2^{r_3})$ as a corresponding ciphertext, and picks a random number $r_2 \in \mathbb{Z}_p$ to encrypt M^{r_1} and sets $C_2 = g^{r_2}$, $C_3 = M^{r_1} H_2(U_1^{r_2})$ as a corresponding ciphertext. These two parts can independently be done and set as a ciphertext of M. Thus, it is possible to make an adversary tamper the ciphertext.

Secondly, the IBEET scheme is a special IBE scheme, which needs to delegate the trapdoor to perform the test algorithm in the cloud server. So the value of M^{r_1} can be known by the adversary after revealing the trapdoor. Thus the adversary can tamper the ciphertext after finding out M^{r_1}.

Finally, we explain that why the above scheme is not secure but security can have been proved. Because there is an obvious gap between views of two adversaries in the IBEET scheme and "PUBK" scheme in [11], but which was not considered in their paper. Since the adversary in IBEET scheme can obtain *trapdoor* and calculate M^{r_1} $(= \frac{C_4}{\cdot})$, the adversary can change C_2 (but not necessary to know the exponent of C_2) and C_3 to construct a new valid ciphertext. However, the adversary in "PUBK" scheme hasn't this capability. Because there does not exist a trapdoor algorithm to delegate the trapdoor information in the scheme. Therefore, the IBEET scheme is not secure, even security of "PUBK" scheme can be reduced to BDH problem in paper [11].

E. FURTHER CONSIDERATION

From the above analysis, we advise to set $r_2 = r_3$, that will lead it to be difficult to tamper the ciphertext. Because if we want to construct a new ciphertext from M^{r_1}, we will know C_5 of the original ciphertext and the input of hash function H_3 in C_5. However, that needs to solve the hard problem BDH. On the other hand, when we formally prove the security of our idea, we should modify the encryption algorithm in "PUBK" scheme such that $C_3 = M^{r_1}$. That should be able to make adversaries in the IBEET scheme and "PUBK" scheme get the same views. The improvement can be proved by using the idea of Boneh and Franklin [19]. Because we use their scheme to encrypt (M, r_1) and M^{r_1} with the same random element r_2, respectively.

IV. SECURITY ANALYSIS OF A CP-ABEET SCHEME

Here, we first recall the notion of CP-ABEET and its security model. And then we review CP-ABEET scheme proposed by Wang *et al.* [16] and analyze its security.

A. MODEL OF CP-ABEET AND ITS SECURITY MODEL

A CP-ABEET scheme includes six algorithms: Setup, KeyGen, Enc, Trapdoor, Dec and Test. The detailed is described as follows.

- Setup. Take a security parameter k as input, and generate a master key MSK and the public parameter $Param$. And keep MSK secure.
- KeyGen. Take as input the public parameter $Param$, the master key MSK and an attribute set AL. Generate the secret key SK for the attribute set W.
- Enc. Take as input $Param$, a plaintext message M and an access structure W. Generate a ciphertext CT.
- Trapdoor. Take as input the public parameter $Param$, an attribute set AL, an access structure T' and MSK, generate the trapdoor TD for users.
- Dec. Take as input the ciphertext CT, the secret key SK, generate the corresponding plaintext M of the ciphertext CT.
- Test. Take as input two ciphertexts CT_A, CT_B, two trapdoors TD_A, TD_B, it outputs 1 if the corresponding plaintext of CT_A and CT_B are the same messages; otherwise, it outputs 0.

Here, we review the definition of the security property defined in Wang *et al.*'s paper as follows [16].

GAME 2. Let $\mathcal{A}$ be an adversary which interacts with a challenger $\mathcal{C}$. $\mathcal{A}$ firstly chooses an access structure W which it wishes to be challenged.

(1) Setup. The challenger $\mathcal{C}$ takes as input a security parameter k, and outputs the public parameter $Param$ and sends it to $\mathcal{A}$.

(2) Phase 1. $\mathcal{A}$ runs the following queries polynomially many times.

- Key retrieve queries: The adversary $\mathcal{A}$ runs queries of the private keys of a set of attributes AL for many access structures T_i. $\mathcal{C}$ sends the corresponding private key SK to $\mathcal{A}$.
- Trapdoor queries: $\mathcal{C}$ runs the Trapdoor algorithm and outputs the trapdoor TD and responds to $\mathcal{A}$.

(3) Challenge: The adversary $\mathcal{A}$ selects two messages M_0 and M_1 with equal length and sends them to $\mathcal{C}$. $\mathcal{C}$ uniformly selects a random bit $b \in \{0, 1\}$, and computes $CT^* = \text{Encrypt}(Param, W, M_b)$ as a challenge, where W is an access structure.

(4) Phase 2: Phase 1 is repeated. The only constraint is that AL which satisfies the access structure W does not appear in the key retrieve queries.

(5) Guess: $\mathcal{A}$ outputs a bit b'.

The adversary $\mathcal{A}$ wins the game if $b = b'$. The advantage of $\mathcal{A}$ is defined as $|\Pr[b = b'] - \frac{1}{2}|$.

Definition 4: The CP-ABEET scheme is selectively IND-CPA secure if the advantage of any polynomial-time adversary is negligible in security parameter k in the above game.

Note 1 : The definition in the paper [16] required that the challenger $\mathcal{C}$ could answer the key retrieve queries for AL of any access structure. That definition is too strong such that there is no scheme satisfying that definition. Because if the adversary can obtain the secret key of AL satisfying the challenge access structure W, then it can decrypt the challenge ciphertext CT^*, and it can win the game with probability 1.

B. RECALL THE CP-ABEET SCHEME AND ITS SECURITY ANALYSIS

We recall the CP-ABEET scheme proposed by Wang *et al.* [16]. Wang *et al.* had shown that their scheme satisfies IND-CPA security. However, we will show that this security property doesn't hold.

1) RECALL THE CP-ABEET SCHEME

- Setup(1^k) : On input a security parameter k, produce $Param$ and the master key MSK as follows.
  - Generate two bilinear groups $\mathbb{G}, \mathbb{G}_T$ with the same prime order p, and generate a random generator $g \in \mathbb{G}$. A map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map.
  - Select two hash functions $H_1 : \mathbb{G}_T \to \mathbb{G} \times \mathbb{Z}_p$, $H_2 : \mathbb{G}_T \to \mathbb{G}$.
  - Randomly choose N elements $r_1, \cdots, r_N \in \mathbb{Z}_p$ and calculate $R_i = g^{r_i}$ for $i = 1$ to N. Where N is the number of system attributes.
  - Randomly choose $\alpha, \alpha', \gamma_1, \gamma_2, \gamma_3 \in \mathbb{Z}_p$ and $W_1, W_2 \in \mathbb{G}$ and calculate

$$
\begin{aligned}
u_1 &= e(g, W_2)^{\alpha\gamma_1} e(g, W_1)^{\alpha\gamma_1}, \\
v_1 &= e(g, W_2)^{\alpha\gamma_2} e(g, W_1)^{\alpha\gamma_2}, \\
u_2 &= e(g, W_2)^{\alpha'\gamma_1} e(g, W_1)^{\alpha'\gamma_1}, \\
v_2 &= e(g, W_2)^{\alpha'\gamma_3} e(g, W_1)^{\alpha'\gamma_3}.
\end{aligned}
$$

  - Set the public parameter $Param = (\mathbb{G}, \mathbb{G}_T, g, p, e, g^{\alpha}, g^{\alpha'}, W_1, W_2, u_1, u_2, v_1, v_2, R_1, \cdots, R_N,$

H_1, H_2) and keep the master key $MSK = (\alpha, \alpha', r_1, \cdots, r_N, \gamma_1, \gamma_2, \gamma_3)$ secret.

- Enc(M, Param, S, S'): On input a message M, public parameter *Param* and an access policy W, which contains: $l_1 \leq L_1$ wildcards occur at positions $J = \{\omega_1, \cdots, \omega_{l_1}\}$, $l_2 \leq L_2$ positive attributes occur at positions $X = \{x_1, \cdots, x_{l_2}\}$, and $l_3 \leq L_3$ negative attributes occur at positions $Y = \{y_1, \cdots, y_{l_3}\}$. By means of the Viéte's formulas, for the wildcard position $\{\omega_k\}_{1,\cdots,l_1}$ in access structure, compute a_{ω_k} and set $t_\omega = \sum_{k=0}^{l_1} a_{\omega_k}$. This algorithm creates the cipheretext CT as follows.
  - Randomly pick $z, z_1, z_2, \in \mathbb{Z}_p$ and calculate

$$C_0 = H_1(u_1^{z_1} v_1^{z_2}) \oplus M \| z, \quad C_1 = M^z H_2(u_2^{z_1} v_2^{z_2}),$$
$$C_2 = g^{\frac{\alpha z_1}{t_\omega}}, \quad C_3 = g^{\frac{z_2}{t_\omega}}, \; C_2' = g^{\frac{\alpha' z_1}{t_\omega}}, \; C_3' = g^z,$$
$$C_4 = (W_1 \prod_{i \in X} R_i^{\frac{\prod_{k=0}^{l_1}(i-\omega_k)}{t_\omega}})^{z_1+z_2},$$
$$C_5 = (W_2 \prod_{i \in Y} R_i^{\frac{\prod_{k=0}^{l_1}(i-\omega_k)}{t_\omega}})^{z_1+z_2}.$$

  - Set $CT = (C_0, C_1, C_2, C_3, C_4, C_5, C_2', C_3', J)$ as the ciphertext.
- KeyGen(Param, MSK, AL): On input the public parameter *Param*, the master key *MSK* and a set of attributes *AL* which contains: $l_2 (\leq L_2)$ positive attributes appear at positions $X = \{x_1', \cdots, x_{l_2}'\}$, $l_3 \leq L_3$ negative attributes appear at positions $Y' = (y_1', \cdots, y_{l_3}')$. By means of the Viète's formula, for all positive positions $\{x_i'\}_{i \in \{1,\cdots,l_2\}}$ and negative positions $\{y_i'\}_{i \in \{1,\cdots,l_3\}}$, compute $\{a_{x_i'}\}$, $\{a_{y_i'}\}$ and set $t_x' = \sum_{k=0}^{l_2} a_{x'}$, $t_y' = \sum_{k=0}^{l_3} a_{y'}$. This algorithm produces the decryption secret key SK as follows:
  - Select a random element $s \in \mathbb{Z}_p$, calculate $s_1 = \gamma_1 + s$, $s_2 = \gamma_2 + s$, $s_3 = \gamma_3 + s$ and generate the decryption secret key as follows.

$$sk_1 = g^{\frac{\alpha s}{t_x'}}, \quad sk_2 = g^{\frac{\alpha s}{t_y'}}, \; sk_1' = g^{\frac{\alpha' s}{t_x'}}, \; sk_2' = g^{\frac{\alpha' s}{t_y'}},$$
$$sk_3 = \{sk_{3,0}, sk_{3,1}, \cdots, sk_{3,L_1}\},$$

where $sk_{3,i} = W_1^{s_1} \prod_{j \in X'} g^{sr_j j^i}$, i from 0 to L_1;

$$sk_3' = \{sk_{3,0}', sk_{3,1}', \cdots, sk_{3,L_1}'\},$$

where $sk_{3,i}' = W_1^{\alpha s_2} \prod_{j \in X'} g^{\alpha s r_j j^i}$, i from 0 to L_1;

$$sk_3'' = \{sk_{3,0}'', sk_{3,1}'', \cdots, sk_{3,L_1}''\},$$

where $sk_{3,i}'' = W_1^{\alpha' s_3} \prod_{j \in X'} g^{\alpha' s r_j j^i}$, i from 0 to L_1;

$$sk_4 = \{sk_{4,0}, sk_{4,1}, \cdots, sk_{4,L_1}\},$$

where $sk_{4,i} = W_2^{s_1} \prod_{j \in Y'} g^{sr_j j^i}$, i from 0 to L_1;

$$sk_4' = \{sk_{4,0}', sk_{4,1}', \cdots, sk_{4,L_1}'\},$$

where $sk_{4,i}' = W_2^{\alpha s_2} \prod_{j \in Y'} g^{\alpha s r_j j^i}$, i from 0 to L_1;

$$sk_4'' = \{sk_{4,0}'', sk_{4,1}'', \cdots, sk_{4,L_1}''\},$$

where $sk_{4,i}'' = W_2^{\alpha' s_3} \prod_{j \in Y'} g^{\alpha' s r_j j^i}$, i from 0 to L_1.

Set $SK = (sk_1, sk_2, sk_1', sk_2', sk_3, sk_3', sk_3'', sk_4, sk_4', sk_4'')$ as the decryption key.

- Trapdoor($Param, AL, SK$): On input the public parameter *Param*, a set of attributes *AL* and the decryption secret key *SK*, output a trapdoor $TD =$

$$(td_1, td_2, (td_{3,i}, td_{3,i}', td_{4,i}, td_{4,i}')_{i \in [0,L_1]}),$$

where $td_1 = sk_1'$, td_2 $td_{3,i} = sk_{3,i}$, $td_{3,i}' = sk_{3,i}''$, $td_{4,i} = sk_{4,i}$, $td_{4,i}' = sk_{4,i}''$, for $i = 0$ to L_1.
- Dec(CT, SK, S, S'): On input the ciphertext CT and the decryption secret key SK, compute the plaintext as follows.

$$V_1 = \frac{e(\prod_{j=1}^{l_1} sk_{3,j}^{a_{\omega_j}}, C_2) e(\prod_{j=1}^{l_1} (sk_{3,j}')^{a_{\omega_j}}, C_3)}{e(sk_1, C_4)^{t_{x'}}} \times \frac{e(\prod_{j=1}^{l_1} sk_{4,j}^{a_{\omega_j}}, C_2) e(\prod_{j=1}^{l_1} (sk_{4,j}')^{a_{\omega_j}}, C_3)}{e(sk_2, C_5)^{t_{y'}}},$$

$$V_2 = \frac{e(\prod_{j=1}^{l_1} sk_{3,j}^{a_{\omega_j}}, C_2') e(\prod_{j=1}^{l_1} (sk_{3,j}'')^{a_{\omega_j}}, C_3)}{e(sk_1', C_4)^{t_{x'}}} \times \frac{e(\prod_{j=1}^{l_1} sk_{4,j}^{a_{\omega_j}}, C_2) e(\prod_{j=1}^{l_1} (sk_{4,j}'')^{a_{\omega_j}}, C_3)}{e(sk_2', C_5)^{t_{y'}}},$$

$M \| z = H_1(V_1) \oplus C_0$.

If $C_3' = g^z$ and $H_2(V_2) = \frac{C_1}{M^z}$, then output the plaintext M. Here all a_k above are coefficients in the unfolding polynomial $\prod_{k=0}^{l_1}(i - \omega_k)$.
- Test($CT_A, CT_B, TD_A, TD_B, S'$) : On input two ciphertexts CT_A, CT_B and the corresponding trapdoors TD_A, TD_B, respectively. This algorithm decides that the planitexts M_A and M_B are equal or not as follows.
Compute

$$Q_A' = \frac{e(\prod_{j=1}^{l_1} td_{3,j,A}^{a_{\omega_j,A}}, C_{2,A}') e(\prod_{j=1}^{l_1} (td_{3,j,A}')^{a_{\omega_j,A}}, C_{3,A})}{e(td_{1,A}, C_{4,A})^{t_{x'_A}}} \times \frac{e(\prod_{j=1}^{l_1} td_{4,j,A}^{a_{\omega_j,A}}, C_{2,A}') e(\prod_{j=1}^{l_1} (td_{4,j,A}')^{a_{\omega_j,A}}, C_{3,A})}{e(td_{2,A}, C_{5,A})^{t_{y'_A}}},$$

$$Q_A = \frac{C_{1,A}}{H_2(Q_A')}.$$

$$Q_B' = \frac{e(\prod_{j=1}^{l_1} td_{3,j,B}^{a_{\omega_j,B}}, C_{2,B}') e(\prod_{j=1}^{l_1} (td_{3,j,B}')^{a_{\omega_j,B}}, C_{3,B})}{e(td_{1,B}, C_{4,B})^{t_{x'_B}}} \times \frac{e(\prod_{j=1}^{l_1} td_{4,j,B}^{a_{\omega_j,B}}, C_{2,B}') e(\prod_{j=1}^{l_1} (td_{4,j,B}')^{a_{\omega_j,B}}, C_{3,B})}{e(td_{2,B}, C_{5,B})^{t_{y'_B}}},$$

$$Q_B = \frac{C_{1,B}}{H_2(Q_B')}.$$

and if $e(Q_B, C_{3,A}') = e(Q_A, C_{3,B}')$ it outputs 1; Otherwise, it outputs 0.

2) THE CP-ABEET SCHEME ISN'T SECURE FOR IND-CPA

Now, we analyze the IND-CPA security of the CP-ABEET scheme.

From the definition of *GAME* 2, we know that any adversary $\mathcal{A}$ selects the challenge messages M_0 and M_1 and obtains the challenge ciphertext CT^* which is a ciphertext of M_0 or M_1. Next, the adversary $\mathcal{A}$ continues to make trapdoor query get the trapdoor TD of the attribute set AL satisfying the access structure W. This is important to our attack. The adversary uses TD and CT^* to compute

$$V = \frac{e(\prod_{j=1}^{l_1} sk_{3,j}^{a_{\omega_j}}, C_2')e(\prod_{j=1}^{l_1}(sk_{3,j}'')^{a_{\omega_j}}, C_3)}{e(sk_1', C_4)^{t_{x'}}} \times \frac{e(\prod_{j=1}^{l_1} sk_{4,j}^{a_{\omega_j}}, C_2)e(\prod_{j=1}^{l_1}(sk_{4,j}'')^{a_{\omega_j}}, C_3)}{e(sk_2', C_5)^{t_{y'}}},$$

and

$$X_{M_b} = \frac{C_1}{H_2(V)}.$$

Then, the adversary verifies the following equality.

$$e(X_{M_b}, g) \stackrel{?}{=} e(M_0, C_3')$$

If the equality holds, the adversary outputs M_0; otherwise, it outputs M_1.

Obviously, if the challenge ciphertext CT^* is a valid ciphertext and the trapdoor TD is valid, then the value of V is correct and such that

$$M_b^z = X_{M_b} = \frac{C_1}{H_2(V)}.$$

Since

$$e(X_{M_b}, g) = e(M_b^z, g) = e(M_b, g^z),$$

while

$$e(M_0, C_3') = e(M_0, g^z).$$

Thus, the equality $e(X_{M_b}, g) = e(M_0, C_3')$ holding means $M_b = M_0$; otherwise, it means $M_b = M_1$.

Thus, the attack can show that the CP-ABEET scheme isn't IND-CPA secure.

C. BRIEF SUMMARY AND FUTURE WORK

Wang *et al.* wanted to construct a CP-ABEET scheme which is IND-CPA secure without random oracle. However, they omitted the adversary can access to the trapdoor oracle to get the trapdoor of any attribute set, which can be used to the Test algorithm. Furthermore, the adversary can choose a message M_b from $\{M_0, M_1\}$ to get a ciphertext CT' and use the Test algorithm to output M_b is M_0 or M_1. If we only want to construct a CP-ABEET scheme which is IND-CPA secure in random oracle model, we can add some hash functions to construct a secure CP-ABEET scheme. However, to construct an IND-CPA secure CP-ABEET in the standard model is not easy. This is our future work.

V. CONCLUSION

IBEET and ABEET are important cryptographic schemes to solve the searching encrypted data in cloud computing. They not only have the functionality of decryption, but also can compare the ciphertexts to determine whether the corresponding plaintexts are the same or not. However, some of the constructions have been omitted that the adversary could get the trapdoors in their security models, and that caused the schemes to be not secure. We analyzed the security of two schemes in this paper. We firstly proved that the IBEET scheme wasn't one-way under chosen ciphertext attack and gave some idea to improve the scheme. Then we proved that the CP-ABEET scheme wasn't indistinguishable against chosen plaintext attack in the standard model.

REFERENCES

[1] G. Ateniese *et al.*, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2007, pp. 598–609.

[2] Y. Liao, Y. He, F. Li, S. Jiang, and S. Zhou, "Analysis of an ABE scheme with verifiable outsourced decryption," *Sensors*, vol. 18, no. 1, p. E176, Jan. 2018. doi: 10.3390/s18010176.

[3] Q. Su, J. Yu, C. Tian, H. Zhang, and R. Hao, "How to securely outsource the inversion modulo a large composite number," *J. Syst. Softw.*, col. 129, pp. 26–34, Jul. 2017.

[4] Y. Liao, Y. He, F. Li, and S. Zhou, "Analysis of a mobile payment protocol with outsourced verification in cloud server and the improvement," *Comput. Standards Interfaces*, vol. 56, pp. 101–106, Feb. 2018.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2004, pp. 506–522.

[6] G. Yang, C. H. Tan, Q. Huang, and D. S. Wong, "Probabilistic public key encryption with equality test," in *Proc. Cryptographers' Track RSA Conf.* Berlin, Germany: Springer, vol. 5985, 2010, pp. 119–131.

[7] Q. Tang, "Towards public key encryption scheme supporting equality test with fine-grained authorization," in *Proc. Australas. Conf. Inf. Secur. Privacy*, in Lecture Notes in Computer Science, vol. 6812. Berlin, Germany: Springer, 2011, pp. 389–406.

[8] Q. Tang, "Public key encryption schemes supporting equality test with authorization of different granularity," *Int. J. Appl. Cryptogr.*, vol. 2, no. 4, pp. 304–321, Jul. 2012.

[9] Q. Tang, "Public key encryption supporting plaintext equality test and user-specified authorization," *Secur. Commun. Netw.*, vol. 5, no. 12, pp. 1351–1362, Dec. 2012.

[10] S. Ma, Q. Huang, M. Zhang, and B. Yang, "Efficient public key encryption with equality test supporting flexible authorization," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 458–470, Mar. 2015.

[11] S. Ma, "Identity-based encryption with outsourced equality test in cloud computing," *Inf. Sci.*, vol. 328, pp. 389–402, Jan. 2016.

[12] H. T. Lee, S. Ling, J. H. Seo, and H. Wang, "Semi-generic construction of public key encryption and identity-based encryption with equality test" *Inf. Sci.*, vol. 373, pp. 419–440, Dec. 2016.

[13] X. Zhang and C. Xu, "Trapdoor security lattice-based public-key searchable encryption with a designated cloud server," *Wireless Pers. Commun.*, vol. 100, no. 3, pp. 907–921, Jun. 2018.

[14] X. Zhang, C. Xu, L. Mu, and J. Zhao, "Identity-based encryption with keyword search from lattice assumption," *China Commun.*, vol. 15, no. 4, pp. 164–178, Apr. 2018.

[15] H. Zhu, L. Wang, H. Ahmad, and X. Niu, "Key-policy attribute-based encryption with equality test in cloud computing," *IEEE Access*, vol. 5, pp. 20428–20439, 2017. doi: 10.1109/ACCESS.2017.2756070.

[16] Q. Wang, L. Peng, H. Xiong, J. Sun, and Z. Qin, "Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing," *IEEE Access*, vol. 6, pp. 760–771, 2018.

[17] Y. Liao, H. Chen, F. Li, S. Jiang, S. Zhou, and R. Mohammed, "Insecurity of A key-policy attribute based encryption scheme with equality test," *IEEE Access*, vol. 6, pp. 10189–10196, 2018. doi: 10.1109/ACCESS.2018.2808944.

[18] D. Boneh, B. Lymn, and H. Shacham, "Short signatures from the weil pairing," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 2248. Berlin, Germany: Springer, 2001, pp. 514–532.
[19] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 2139. Berlin, Germany: Springer, 2001, pp. 213–229.

YONGJIAN LIAO (M'17) received the Ph.D. degree in applied electronic science and tech-nology from the College of Information Science and Electronic Engineering, Zhejiang University, in 2007. He is currently an Associate Profes-sor with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His main research interests include public-key cryptography and information security, in particular, cryptographic protocols.

HONGJIE CHEN received the B.S. degree in information security from the Chongqing Univer-sity of Posts and Telecommunications, in 2017. She is currently pursuing the master's degree in cryptography with the University of Electronic Science and Technology of China. Her research interests include cryptography and information security.

WEN HUANG received the B.S. degree in com-puter science and technology from the University of Electronic Science and Technology of China, in 2016, where he is currently pursuing the M.S. degree in information security. His research inter-ests include cryptography and information secu-rity. He is a member of the ACM and CCF.

RAMADAN MOHAMMED received the Ph.D. degree in computer science and technology from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, in 2016, where he is currently a Post-doctoral Staff of the School of Information and Software Engineering. His main research interests include public-key cryptography and information security, in particular, some protocols for cloud security.

HONGTAO PAN received the B.S. degree in com-puter science and technology from the University of Helongjiang, in 2017. He is currently pursuing the M.S. degree in information security with the University of Electronic Science and Technology of China. His research interests include cryptogra-phy and information security.

SHIJIE ZHOU (M'06) received the Ph.D. degree in computer science and technology from the Uni-versity of Electronic Science and Technology of China, in 2004, where he is currently a Professor with the School of Information and Software Engi-neering. His research interests include communi-cation and security in computer networks, peer-to-peer networks, sensor networks, cloud security, and big data.