Insecurity of A Key-Policy Attribute Based Encryption Scheme With Equality Test

YONGJIAN LIAO[1], (Member, IEEE), HONGJIE CHEN[1], FAGEN LI[2], (Member, IEEE),
SHAOQUAN JIANG[3], SHIJIE ZHOU[1], AND RAMADAN MOHAMMED[1]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[3]Institute of Information Security, Mianyang Normal University, Mianyang 621000, China

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

This work was supported in part by the National Natural Science Foundation of China under Grant 61472066.

ABSTRACT Attribute-based encryption is a popular cryptographic technology to protect the privacy of clients' data in cloud computing. In order to make the scheme have the functionality of comparing ciphertext, Zhu *et al.* combined concepts of key-policy attributed-based encryption with public key encryption with equality test and proposed key-policy attributed-based encryption with equality test. They defined its security model and put forward a scheme from bilinear pairing. In this paper, we first use two methods to show that their scheme is not secure for one-way under chosen ciphertext attack. Next, we show that the scheme is not secure for a test under chosen ciphertext attack defined in their paper yet. Finally, we point out the definition of a test under chosen ciphertext attack is very strong, which causes no scheme to satisfy the model.

INDEX TERMS Attribute-based encryption, one-way, equality test, chosen cihpertext attack, cloud computing, insecurity.

I. INTRODUCTION

Recently, in cloud computing environment, clients are used to store their data to cloud servers in order to save their storage cost. However, to protect the privacy of data of clients, these outsourced data have to be stored in encrypted form which can not efficiently extract some statistical information of the data for users in the future. To keep sharing of encrypted data fine-grained, Sahai and Waters [1] put forth attribute-based encryption (ABE), which is a public key encryption variant that allows clients to access secret information according to their attributes.

In practical applications, there are two different kinds of ABE schemes based on the manner to deploy the access control policy, key-policy ABE encryption (KP-ABE) [2] and ciphertext-policy ABE (CP-ABE) [3]. In the former, The ciphertexts are associated with attributes' sets and the clients' private keys are associated with access policies over these attributes. While in the latter, the ciphertext is associated with an access policy, and the clients' private keys are associated with a set of attributes. However, these schemes cannot perform to compare plaintexts unless it decrypts the corresponding ciphertexts of the plaintexts. Recently, Zhu *et al.* [4] introduced a new concept by combining the concept of public key encryption with equality test (PKE-ET) [5] and the concept of KP-ABE, called key-policy attribute-based encryption with equality test (KP-ABE-ET). They formally defined the security model and constructed a KP-ABE-ET scheme from bilinear pairing. Furthermore, they showed that the KP-ABE-ET scheme was one-way under chosen ciphertext attack (OW-CCA) and a testable property under chosen ciphertext attack (T-CCA).

A. CONTRIBUTION

In this paper, we show that the KP-ABE-ET scheme proposed by Zhu *et al.* is insecure as follows. We firstly show their KP-ABE-ET scheme is not secure under OW-CCA by using two methods to construct a valid ciphertext, respectively. Then we show their KP-ABE-ET scheme isn't secure under T-CCA yet. Finally, we point out that the definition of T-CCA is not reasonable and redefine it.

B. RELATED WORKS

In 2010, Yang *et al.* [5] proposed a new concept — PKE-ET, which is a coalition of public key encryption (PKE) and searchable encryption [6]. The PKE-ET not only has decryption functionality, but also has the property of checking if ciphertexts are encryptions of the same unknown keyword even if they are possible to use different public keys. To make checking algorithm fine-grained, Tang [7] improved the scheme to propose a PKE-ET with fine-grained authorization scheme(FG-PKE-ET), all-or-nothing PKEET (AoN-PKE-ET) [8] and an extension of FG-PKE-ET [9]. Ma *et al.* [10] put forward a PKE-ET supporting flexible authorization (PKE-ET-FA). In their scheme they defined 4 types of flexible authorizations. In order to simplify the PKE-ET's certificate management, Ma [11] combined the concept of PKE-ET with the concept of identity-based encryption (IBE) to propose IBE with equality test (IBE-ET). Recently, to optimize the computational overhead, Wu *et al.* [12] improved the IBE-ET scheme proposed by Ma *et al.* And Lee *et al.* [13] proposed Semi-generic construction of PKE-ET scheme and IBE-ET scheme.

C. ORGANIZATION

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 their KP-ABE-ET scheme and show that the scheme isn't secure based on their security models in Section III. Finally, we conclude the paper in Section IV.

II. PRELIMINARY

Here, we first recall some basic mathematical knowledge which was used in their KP-ABE-ET scheme.

A. BILINEAR PAIRING

Set two multiplicative groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with the same prime order q. Let $\mathbb{Z}_q^*$ be the multiplicative group of $\mathbb{F}_q$ (the finite field). A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ [14], which satisfies the following three properties:

- Bilinearity: For any $\alpha, \beta, \gamma \in \mathbb{G}_1$,

$$e(\alpha, \beta\gamma) = e(\alpha, \beta)e(\alpha, \gamma), \text{ and}$$
$$e(\alpha\beta, \gamma) = e(\alpha, \gamma)e(\beta, \gamma).$$

- Non-degeneracy: There are elements $\alpha, \beta \in \mathbb{G}_1$, such that $e(\alpha, \beta) \neq 1$, where 1 is the identity element of $\mathbb{G}_2$.
- Computability: For any elements $\alpha, \beta \in \mathbb{G}_1$, there is an efficient algorithm to compute $e(\alpha, \gamma)$.

Definition 1: Bilinear Diffie-Hellman problem (BDH problem). Set $\mathbb{G}_1$ and $\mathbb{G}_2$ to be the groups of order q above. Given four elements $g, g^\alpha, g^\beta, g^\gamma \in \mathbb{G}_1$ for some unknown $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$, to calculate $e(g, g)^{\alpha\beta\gamma}$. Where g isn't the identity element of $\mathbb{G}_1$ and $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

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

$$D_0 = \{g, g^a, g^b, g^c, g^\zeta, g^\tau, e(g, g)^{abc}, e(g, g)^{a\zeta\tau} : a, b, c, \zeta, \tau \in \mathbb{Z}_q\}$$
$$D_1 = \{g, g^a, g^b, g^c, g^\zeta, g^\tau, e(g, g)^x, e(g, g)^y : a, b, c, x, \zeta, \tau, y \in \mathbb{Z}_q\}$$

to decide whether $abc \equiv x \bmod q$ and $a\zeta\tau \equiv y \bmod q$ hold or not.

B. ACCESS TREE

Let T be an access tree made up of many non-leaf nodes and many leaf nodes. Each leaf and non-leaf node represent an attribute and a threshold gate, respectively. Each threshold gate is denoted by its children and the threshold value. Set num_x and k_x as the number of children and the threshold value of the node x, respectively. We have $0 \leq k_x \leq num_x$. Then, every leaf node x has a threshold value $k_x = 1$.

Let every node's children do have orders which are from 1 to *num*. Next, we recall the following functions which were defined in [4]. The parent of node x is denoted by function $parent(x)$. An attribute associated with the leaf node x is denoted by the function $att(x)$. And the number associated with node x is denoted by the function $index(x)$.

T_r refers to the tree T rooted at the root r. T_x refers to the subtree of T which is rooted at node x. The set of attributes S satisfies the tree T_x if and only if $T_x(S) = 1$ holds. Here, we compute $T_x(S)$ by using the following recursive algorithm.

- Compute $T_z(S)$ for all children z of x if x is a non-leaf node. $T_x(S)$ outputs 1, if and only if there are at least k_x children to output 1.
- Otherwise (node x is a leaf node), $T_x(S)$ outputs 1 if and only if $att(x) \in S$.

III. ANALYSIS OF ZHU *et al.*'s KP-ABE-ET SCHEME

Here, we first recall the concept of KP-ABE-ET and its secu-rity models. And then we review the KP-ABE-ET scheme put forth by Zhu *et al.* and analyze its security.

A. MODEL OF KP-ABE-ET AND ITS SECURITY MODEL

A KP-ABE-ET scheme [4] is made up of six algorithms: Setup, Encrypt, KeyGen, Trapdoor, Decrypt and Test. $\mathbb{M}$ and $\mathbb{C}$ are the plaintext space and the ciphertext space, respectively. The detailed is described as follows:

- Setup(1^k): On input a security parameter 1^k, generate the public parameter *Param* and the master key *MSK*. Keep *MSK* secret and publish the public parameter *Param*.
- KeyGen($T, T', S, S', Param, MSK$): On input T, T', and S, S'(where $T(S) = 1$ and $T'(S') = 1$), the public parameter *Param*, the master key *MSK*, generate the decryption secret key *SK* for users.
- Encrypt($Param, M, S, S'$): On input the public parameter *Param*, a plaintext M and two sets of attributes S, S', generate the ciphertext $CT \in \mathbb{C}$.
- Trapdoor($Param, S', T', MSK$): On input the public parameter *Param*, an attribute set S', an access trees T' and *MSK*, generate the trapdoor *TD* for users.
- Decrypt(CT, SK, S, S'): On input the ciphertext CT, the decryption secret key *SK* and two attribute sets S, S', produce the corresponding plaintext M if $T(S) = 1$ and $T'(S') = 1$ hold. Otherwise, outputs $' \perp '$.
- Test($CT_A, TD_A, CT_B, TD_B, S'_A, S'_B$): On input two ciphertexts CT_A, CT_B, two trapdoors TD_A, TD_B, and two attribute sets S'_A, S'_B, it outputs '1' if the corresponding

plaintexts of CT_A and CT_B are common; otherwise, it outputs '0'.

Here, we review the definition of two security properties defined in Zhu *et al.*'s paper. First, the one-way against chosen-ciphertext attack (OW-CCA) for KP-ABE-ET under a chosen set of attributes is described as follows [4].

Game 1: Let $\mathcal{A}$ be an adversary which interacts with a challenger $\mathcal{C}$. $\mathcal{A}$ announces a challenged set S of attributes.

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

(2) Phase 1: $\mathcal{A}$ makes the queries below.

- Key retrieve queries: For any access structure T_i, the adversary $\mathcal{A}$ runs queries of the private keys, where $S \notin T_i$ for any i. $\mathcal{C}$ sends the corresponding private key SK to $\mathcal{A}$.
- Decryption queries: when $\mathcal{A}$ runs queries for ciphertexts, $\mathcal{C}$ uses the Decrypt algorithm to produce the corresponding plaintexts of the ciphertexts or $\perp$, and sends it to $\mathcal{A}$.
- Trapdoor queries: $\mathcal{C}$ uses the Trapdoor algorithm to produce the trapdoor TD and responds it to $\mathcal{A}$.

(3) Challenge: $\mathcal{C}$ selects a random plaintext $M^* \in \mathbb{M}$, computes $CT^* = \text{Encrypt}(Param, M^*)$ and sends the challenged ciphertext CT^* to $\mathcal{A}$.

(4) Phase 2: Repeat the Phase 1. The only constraint is that CT^* is not queried in the decryption queries.

(5) Guess: At last, the adversary $\mathcal{A}$ outputs a plaintext $M \in \mathbb{M}$ as its guess.

If $M^* = M$ the adversary $\mathcal{A}$ wins the above game. The advantage of $\mathcal{A}$ is defined as the probability $\Pr[M^* = M]$.

Definition 3: The KP-ABE-ET scheme is OW-CCA secure if the advantage of any polynomial-time adversary is negligible in security parameter k in the above game.

Next, we review the definition of a test against chosen-ciphertext attack (T-CCA) of authorization for KP-ABE-ET under the chosen sets of attributes as follows.

Game 2: Let $\mathcal{A}$ be an adversary which interacts with a challenger $\mathcal{C}$. $\mathcal{A}$ publishes two challenged sets of attributes S, S'. Here, $S \cap S' = \Phi$, and S and S' are used for decryption and the trapdoor, respectively.

(1) Setup: The challenger $\mathcal{C}$ takes as input a security parameter 1^k and uses the Setup algorithm of KP-ABE-ET to produce the public parameters *Param* and sends it to $\mathcal{A}$.

(2) Phase 1: $\mathcal{A}$ makes the queries below.

- Key retrieve queries: For access structures T_i and T'_j, $\mathcal{A}$ can make the private keys' queries, where $S \notin T_i$ for all i and $S' \notin T_j$ for all j. $\mathcal{C}$ sends the corresponding private key SK to $\mathcal{A}$.
- Decryption queries: $\mathcal{A}$ can make the decryption queries for ciphertexts. $\mathcal{C}$ uses the decrypt algorithm to produce the corresponding plaintext, and sends it to $\mathcal{A}$.
- Trapdoor queries: When $\mathcal{A}$ makes the Trapdoor queries, $\mathcal{C}$ uses the trapdoor algorithm to produce TD and sends it to $\mathcal{A}$.
- Test queries: When $\mathcal{A}$ makes test queries for some ciphertexts, $\mathcal{C}$ outputs 1 for equality ciphertexts by running the test algorithm; it outputs 0, otherwise.

(3) Challenge: The challenger $\mathcal{C}$ selects a random coin $c \in \{0, 1\}$. If $c = 1$, then $\mathcal{C}$ selects a random plaintext $M \in \mathbb{M}$, outputs

$$CT_1^* = Encrypt(Param, M),\ \ CT_2^* = Encrypt(Param, M)$$

and sends the challenged ciphertexts CT_1^*, CT_2^* to $\mathcal{A}$.

If $c = 0$, then $\mathcal{C}$ randomly selects two distinct plaintext M_1 and M_2, outputs

$$CT_1^* = Encrypt(Param, M_1),\ \ CT_2^* = Encrypt(Param, M_2)$$

and sends the challenged ciphertexts CT_1^*, CT_2^* to $\mathcal{A}$.

(4) Phase 2: Repeat Phase 1, but the constrained conditions are that CT_1^* and CT_2^* are not queried in decryption queries and CT_1^* and CT_2^* are not queried in test queries.

(5) Guess: At last, $\mathcal{A}$ guesses a coin c^*.

$\mathcal{A}$ wins the above game if $c = c^*$. That is to say, $c = c^* = 1$ means $M_1 = M_2$, and $c = c^* = 0$ means $M_1 \neq M_2$. The advantage of $\mathcal{A}$ is defined as the probability $|\Pr[c^* = c] - \frac{1}{2}|$.

Definition 4: The KP-ABE-ET scheme is T-CCA secure if the advantage of any polynomial-time adversary is negligible in the security parameter 1^k in the aforementioned **Game 2**.

B. ANALYSIS OF THE KP-ABE-ET SCHEME

Since the KP-ABE-ET scheme proposed by Zhu *et al.* is somewhat complex, we recall their scheme in APPENDIX A.

Zhu *et al.* had shown that their scheme satisfies OW-CCA security and T-CCA security. However, we will show that both security properties don't hold.

1) THE SCHEME ISN'T SECURE FOR OW-CCA

Now, we analyze the OW-CCA security of the KP-ABE-ET scheme.

From the definition of **Game 1**, we know that any adversary can request key retrieve queries, decryption queries and trapdoor queries before and after the challenger generates the challenge ciphertext CT^*. The only restricted condition is that the adversary can't make the decryption query for CT^*. Thus, when the adversary has gotten the challenge ciphertext CT^* of some plaintext M^* which is chosen randomly by the challenger. It can generate a new ciphertext CT according to the following two constructions, respectively. Here, set $CT^* = (S^*, S'^*, C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$, where

$$\begin{aligned}
&C_1^* = g^{r_1},\ \ C_2^* = M^* \| r_1 \oplus H_1(S^*, {Y_1}^{r_2}),\\
&C_3 = (M^*)^{r_1} H_2(S'^*, {Y_2}^{r_3}),\\
&C_4^* = \{E_i = {X_i}^{r_2}\}_{i \in S^*},\ \ C_5^* = \{E_j = {X_j}^{r_3}\}_{j \in S'^*},\\
&C_6^* = H_3((M^*)^{r_1}, C_1^*, C_2^*, C_3^*, C_4^*, C_5^*)
\end{aligned}$$

for some unknown elements $r_1, r_2, r_3 \in \mathbb{Z}_q$.

a: THE FIRST CONSTRUCTION

The adversary firstly submits CT^* and S'^* to request a trapdoor query to get the corresponding trapdoor TD. Next, it uses the trapdoor TD to compute

$$C'_3 = \frac{C^*_3}{H_2(S'^*, {Y_2}^{r_3})} = (M^*)^{r_1},$$

and then it randomly chooses an element $r'_3 \in \mathbb{Z}_q$ to compute

$$C_3 = C'_3 \, H_2(S'^*, {Y_2}^{r'_3}), \quad C_5 = \{E_j = {X_j}^{r'_3}\}_{j \in S'^*}$$
$$C_6 = H_3(C'_3, C^*_1, C^*_2, C_3, C^*_4, C_5).$$

Finally, it sets

$$C = (S^*, S'^*, C^*_1, C^*_2, C_3, C^*_4, C_5, C_6)$$

as a ciphertext.

Obviously, the ciphertext C ($\neq CT^*$) is valid and its corresponding plaintext is M^*. Because the challenged ciphertext is a valid ciphertext, and the value of C'_3 is $(M^*)^{r_1}$, which is a correct value by using the trapdoor TD to calculate. Subsequently, the adversary produces C_3 and C_5 by choosing a random element $r'_3 \in \mathbb{Z}_q$ and running the Encryption algorithm. The probability of that r'_3 equals the unknown element r_3 is $\frac{1}{q}$, which is negligible. Since $(S^*, S'^*, C^*_1, C^*_2, C^*_4)$ is gotten from the challenger which is a part of challenge ciphertext, the adversary can easily compute the value of C_6, that is, $H_3(C'_3, C^*_1, C^*_2, C_3, C^*_4, C_5)$.

Next, according to the **Game 1**, the adversary can request a decryption query for any ciphertext $C \neq C^*$. The challenger decrypts C to get the plaintext by using the decryption algorithm and returns the corresponding plaintext of the query. Actually, the plaintext of C is M^*. Thus, the probability of that the adversary wins the **Game 1** is 1.

b: THE SECOND CONSTRUCTION

This construction is easy and rough, but effective. When the adversary gets the challenge ciphertext CT^*, it randomly chooses C_3, and produces $C_5 = \{E_j\}_{j \in S'}$ for randomly choosing $E_j \in \mathbb{G}_1$, and then it calculates

$$C_6 = H_3(C'_3, C^*_1, C^*_2, C_3, C^*_4, C_5).$$

At last, it sets the ciphertext

$$C = (S^*, S'^*, C^*_1, C^*_2, C_3, C^*_4, C_5, C_6).$$

The ciphertext C ($\neq CT^*$) is also valid and its corresponding plaintext is M^*. Because the challenge ciphertext CT^* is valid firstly, and

$$C^*_1 = g^{r_1}, M^* \| r_1 = C^*_2 \oplus H_1(S^*, {Y_1}^{r_2})$$

always hold for the unknown elements r_1, r_2. From the construction of their decryption algorithm, to compute $M^* \| r_1$ is only to utilize S, C^*_2, C^*_4. Thus, the two equalities

$$C^*_1 = g^{r_1} \text{ and } C_6 = H_3(C'_3, C^*_1, C^*_2, C_3, C^*_4, C_5)$$

hold, and $C = (S^*, S'^*, C^*_1, C^*_2, C_3, C^*_4, C_5, C_6)$ is valid.

Next, according to the **Game 1**, the adversary can request a decryption query for ciphertext $C \neq C^*$. The challenger decrypts C to get the plaintext M^* by using the decryption algorithm and sends M^* to the adversary. Thus, the probability of that the adversary wins the **Game 1** is 1.

Thus, These two methods can show that the KP-ABE-ET scheme isn't OW-CCA secure.

C. THE SCHEME ISN'T SECURE FOR T-CCA

From the definition of **Game 2**, we know that any adversary can request key retrieve queries, decryption queries, trapdoor queries and test queries before and after the challenger generates the challenge ciphertext CT^*_1 and CT^*_2. The restricted conditions are

- the adversary can't request the key retrieve query T, T', where $S^* \in T$, $S'^* \in T'$.
- CT^*_1 and CT^*_2 are not queried by the adversary in the decryption queries.
- CT^*_1 and CT^*_2 are not queried by the adversary in the test queries.

When the adversary receives two challenge ciphertexts CT^*_1, CT^*_2, it requests trapdoor queries to get the trapdoor TD. Where

$$CT^*_1 = (S^*, S'^*, C^*_1, C^*_2, C^*_3, C^*_4, C^*_5, C^*_6)$$

produced by using three unknown random elements r_1, r_2, $r_3 \in \mathbb{Z}_q$,

$$CT^*_2 = (S^*, S'^*, \bar{C}^*_1, \bar{C}^*_2, \bar{C}^*_3, \bar{C}^*_4, \bar{C}^*_5, \bar{C}^*_6)$$

produced by using three unknown random elements r'_1, r'_2, $r'_3 \in \mathbb{Z}_q$.

Then it can compute

$$M_1{}^{r_1} = \frac{C^*_3}{H_2(S', {Y_2}^{r_3})}, \text{ and}$$
$$M_2{}^{r'_1} = \frac{\bar{C}^*_3}{H_2(S', {Y_2}^{r'_3})}$$

At last, the adversary outputs 1 if the equality

$$e(M_1{}^{r_1}, \bar{C}^*_1) = e(M_2{}^{r'_1}, C^*_1)$$

holds; outputs 0, otherwise.

Actually, because the two challenged ciphertexts CT^*_1 and CT^*_2 are valid and the trapdoor TD obtained from the challenger is correct, the adversary can compute the correct value $M_1{}^{r_1}$ and $M_2{}^{r'_1}$. Since

$$e(M_1{}^{r_1}, \bar{C}^*_1) = e(M_1{}^{r_1}, g^{r'_1}) = e(M_1, g)^{r_1 r'_1},$$
$$e(M_2{}^{r'_1}, C^*_1) = e(M_2{}^{r'_1}, g^{r_1}) = e(M_2, g)^{r'_1 r_1},$$

the equality $e(M_1{}^{r_1}, \bar{C}^*_1) = e(M_2{}^{r'_1}, C^*_1)$ holds if and only if $M_1 = M_2$. So the adversary outputs 1 if and only if the challenger generates two challenged ciphertexts by encrypting the same plaintext. The adversary successfully wins the game **Game 2** with probability 1.

Thus, we show that the KP-ABE-ET scheme is not T-CCA secure.

D. FURTHERMORE DISCUSSION

The KP-ABE-ET has two functionalities: encryption & decryption and test. The concept of OW-CCA formally defines the security of the encryption & decryption algorithms, and the concept of T-CCA formally defines the security of the test algorithm. *However, the definition of T-CCA in* **Game 2** *is very strong in their paper, which causes that their scheme did satisfy this definition.* Furthermore, that definition causes there will be no scheme which satisfies the T-CCA security. Because the goal of the trapdoor algorithm is to output a trapdoor which will be used to the test algorithm. That is to say, anyone who gets the trapdoors can run the test algorithm for some ciphertexts. And any adversary can request the trapdoor queries for any access structure S' in **Game 2** including the challenge access structure. Thus, the adversary can perform the test algorithm which can check ciphertexts to decide whether the corresponding plaintexts are the same or not. This causes that the adversary in **Game 2** always wins. Thus, the definition of T-CCA is not reasonable and we should modify it as follows.

Game 3: Let $\mathcal{A}$ be an adversary which interacts with a challenger $\mathcal{C}$. $\mathcal{A}$ publishes two challenged sets of attributes S, S'. Here, $S \cap S' = \Phi$, and S and S' are used for decryption and the trapdoor, respectively.

(1) Setup: The challenger $\mathcal{C}$ takes as input a security parameter 1^k and uses the Setup algorithm of KP-ABE-ET to generate the public parameters *Param* and sends to $\mathcal{A}$.

(2) Phase 1: $\mathcal{A}$ makes queries below.

- Key retrieve queries: For access structures T_i and T'_j, $\mathcal{A}$ makes many private keys queries, for all i, j $S \notin T_i$ and $S' \notin T_j$, respectively. And $\mathcal{C}$ sends the private key SK to $\mathcal{A}$.
- Decryption queries: $\mathcal{A}$ requests many queries for ciphertexts. $\mathcal{C}$ runs the Decrypt algorithm and outputs the corresponding plaintext to the ciphertext or $\perp$ to $\mathcal{A}$.
- Trapdoor queries: For any access structures T_i and T'_j, $\mathcal{A}$ requests many queries of the trapdoor, where for all i, j, $S \notin T_i$ and $S' \notin T_j$, respectively. $\mathcal{C}$ sends the trapdoor TD to $\mathcal{A}$.
- Test queries: When $\mathcal{A}$ makes test queries for some ciphertexts, $\mathcal{C}$ outputs '1' for equality ciphertexts by running the test algorithm; it outputs '0', otherwise.

(3) Challenge: The challenger $\mathcal{C}$ selects a random coin $c \in \{0, 1\}$. And then $\mathcal{C}$ randomly selects a plaintext M if If $c = 1$, and produces

$$CT_1^* = Encrypt(Param, M), CT_2^* = Encrypt(Param, M),$$

and sends the challenged ciphertexts CT_1^*, CT_2^* to $\mathcal{A}$.

Otherwise ($c = 0$), $\mathcal{C}$ randomly selects two distinct plaintext M_1 and M_2, outputs

$$CT_1^* = Encrypt(Param, M_1), CT_2^* = Encrypt(Param, M_2)$$

and sends the challenged ciphertexts CT_1^*, CT_2^* to $\mathcal{A}$.

(4) Phase 2: Repeat the Phase 1, but the restricted conditions are that CT_1^* and CT_2^* are not queried in decryption queries and in the test queries, respectively.

(5) Guess: $\mathcal{A}$ guesses a coin c^*.

$\mathcal{A}$ wins the above game, **Game 3**, if $c = c^*$. That is to say, $c = c^* = 1$ means $M_1 = M_2$, and $c = c^* = 0$ means $M_1 \neq M_2$. The advantage of $\mathcal{A}$ is defined as the probability $|\Pr[c^* = c] - \frac{1}{2}|$.

Definition 5: The KP-ABE-ET scheme is weak-T-CCA secure if the advantage of any polynomial-time adversary is negligible in the security parameter 1^k in the aforementioned **Game 3**.

On the other hand, although the the KP-ABE-ET scheme proposed by Zhu *et al.* was not OW-CCA secure, it seems that simply defining C_2 by computing $M\|r_1 \oplus H_1(S, {Y_1}^{r_2}, C_1, C_3, C_4, C_5)$ instead of $M\|r_1 \oplus H_1(S, {Y_1}^{r_2})$ can make their scheme satisfy OW-CCA security. Since the adversary in **Game 1** cannot get ${Y_1}^{r_2}$, it cannot generate a new valid C_2 from the challenged ciphertexts when C_i is falsified, for $i \in \{1, 3, 4, 5\}$. The modification also make the scheme satisfy our weak-T-CCA security. The detailed proof of security of modification can be shown by using the method similar to [4, Ths. 2 and 3].

IV. CONCLUSION

ABE is a popular cryptographic technology to protect the security of clients' data in cloud computing. KP-ABE-ET not only utilizes the flexible property of ABE, but also can compare the ciphertexts to determine whether the corresponding plaintexts are the same or not. In this paper, we firstly showed a KP-ABE-ET scheme proposed by Zhu *et al.* isn't secure under OW-CCA security. Then we proved that the scheme isn't secure for T-CCA security defined in their paper yet. Finally, we pointed out the definition of T-CCA is very strong, which causes no scheme to satisfy the security model, and we redefined the T-CCA security.

APPENDIX
RECALL THE KP-ABE-ET SCHEME

We recall the KP-ABE-ET scheme proposed by Zhu *et al.* [4] as follows.

Setup(1^k): On input a security parameter 1^k, generate the public parameter *Param* and the master key *MSK* as follows.

- Generate two bilinear groups $\mathbb{G}_1, \mathbb{G}_2$ with the same prime order q, and randomly generate a generator $g \in \mathbb{G}_1$, and a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$.
- Set A to be a universe of properties of attributes. In order to simplify the notations, set the first A elements of $\mathbb{Z}_q^*$ to be the universe, i.e., $1, 2, \cdots, |A| (\mathrm{mod}\ q)$.
- Define three secure hash functions $H_1 : \{0, 1\}^{|A|} \times \mathbb{G}_2 \rightarrow \{0, 1\}^{k+l}$, $H_2 : \{0, 1\}^{|A|} \times \mathbb{G}_2 \rightarrow \mathbb{G}_1$, and $H_3 : (\mathbb{G}_1)^5 \times \{0, 1\}^{k+l} \rightarrow \{0, 1\}^k$, where the length of q is l.
- Select $x_1, x_2, \cdots, x_{|A|}, y_1, y_2 \in \mathbb{Z}_q^*$ randomly, and then compute

$$X_1 = g^{x_1}, \ X_2 = g^{x_2}, \cdots, X_{|A|} = g^{x_{|A|}},$$
$$Y_1 = g^{y_1}, \ Y_1 = g^{y_2},$$

output the public parameter $Param = (\mathbb{G}_1, \mathbb{G}_2, g, e, H_1, H_2, H_3, k, l, q, X_1, X_2, \cdots, X_{|A|}, Y_1, Y_2)$ and the master key $MSK = (x_1, x_2, \cdots, x_{|A|}, y_1, y_2)$.

Encrypt(M, Param, S, S'): On input a message M, public parameter *Param* and two sets of attributes S, S' (here, $S \cap S' = \Phi$). Then, output the ciphertext as follows.

Randomly choose $r_1, r_2, r_3 \in \mathbb{Z}_q$, and then compute:

$$\begin{aligned} &C_1 = g^{r_1}, \quad C_2 = M\|r_1 \oplus H_1(S, {Y_1}^{r_2}), \\ &C_3 = M^{r_1} H_2(S', {Y_2}^{r_3}), \\ &C_4 = \{E_i = {X_i}^{r_2}\}_{i\in S}, \quad C_5 = \{E_j = {X_j}^{r_3}\}_{j\in S'}, \\ &C_6 = H_3(M^{r_1}, C_1, C_2, C_3, C_4, C_5). \end{aligned}$$

Output the ciphertext $CT =$

$$(S, S', C_1, C_2, C_3, C_4, C_5, C_6).$$

KeyGen(T,T',S, S', Param, MSK): On input two access structures T, T', the master key MSK and two attributes sets S, S' which satisfy the following conditions: $T(S) = 1$, $T'(S') = 1$ and $S' \cap S = \Phi$. And output the private key as follows.

(1) For every node x in T, pick a polynomial q_x from top to bottom, starting from the root node r. That is,

- For every node x in T, set d_x to be the degree of the polynomial q_x such that $d_x = k_x - 1$, where k_x is the threshold value of that node.
- Then set $q_r(0) = y_1$ for the root node r. Select d_r other points of the polynomial q_r randomly to define the polynomial $q_r)$ completely.
- For any other node x, set $q_x(0)$ to be $q_{parent(x)}(index(x))$. And select d_x other points randomly to completely define q_x.
- For every leaf node x, output $D_x = g^{\frac{q_x(0)}{x_i}}$, for $i = att(x)$.

(2) For every node t in T', pick a polynomial q_t from top to bottom, starting from the root node r'. That is,

- For every node t in T', set d_t to be the degree of the polynomial q_t such that $d_t = k_t - 1$, where k_t is the threshold value of that node.
- Then set $q_{r'}(0) = y_2$ for the root node r'. Select $d_{r'}$ other points of the polynomial $q_{r'}$ randomly to define the polynomial $q_{r'}(x)$ completely.
- For any other non-leaf node t, set $q_t(0) = q_{parent(t)}(index(t))$ and select d_t other points randomly to define the polynomial q_t completely.
- For every leaf node t, output $T_t = g^{\frac{q_t(0)}{x_j}}$, for $j = att(t)$.

(3) Output SK as the secret key. $SK =$

$$(D_x = g^{\frac{q_x(0)}{x_i}}, T_t = g^{\frac{q_t(0)}{x_j}}),$$

for $i = att(x), j = att(t)$.

Trapdoor(T', S', MSK): On input T', S' and MSK, if $T'(S') = 1$, output a trapdoor TD which can test the ciphertexts as follows.

For every node t in T', pick a polynomial q_t from top to bottom, starting from the root node r'. That is,

- For every node t in T', set d_t to be the degree of the polynomial q_t such that $d_t = k_t - 1$, where k_t is the threshold value of that node.
- Then set $q_{r'}(0) = y_2$ for the root node r'. Select $d_{r'}$ other points of the polynomial $q_{r'}$ randomly to define the polynomial $q_{r'}$ completely.
- For any other non-leaf node t, set $q_t(0)$ to be $q_{parent(t)}(index(t))$ and select d_t other points randomly to define the polynomial q_t completely.
- For every leaf node t, output $T_t = g^{\frac{q_t(0)}{x_j}}$, for $j = att(t)$.

Output $TD = g^{\frac{q_t(0)}{x_j}}$ as a trapdoor for $j = att(t)$.

Decrypt (CT,SK, S,S'): We use a recursive algorithm $DecryptNode(CT, SK, x)$ to define the decryption algorithm described as follows. On input the ciphertext CT, the private key SK and a node x of the tree T, output $\perp$ or a plaintext which is an element of the group $\mathbb{G}_2$.

For a leaf node x if $i \in S$ for $i = att(x)$, the calculate:

$$DecryptNode(CT, SK, x) = e(D_x, E_i)(= e(g, g)^{r_2 q_x(0)}).$$

Otherwise, $DecryptNode(CT, SK, x) = \perp$.

For a non-leaf node x, then the algorithm *DecryptNode* (CT, SK, x) runs as follows.

Let z be a child of x. $DecryptNode(CT, SK, z)$ stores and outputs O_z. Let F_x be an arbitrary k_x-sized subset of child nodes z such that $O_z \neq \perp$. If no the subset exists (the node is not satisfied), then return $\perp$. Otherwise, compute:

$$O_x = e(g, g)^{r_2 q_x(0)}.$$

Let $DecryptNode(CT, SK, r) = {Y_1}^{r_2}$ $(=e(g, g)^{y_1 r_2})$. Calculate $M\|r_1 = C_2 \oplus H_1(S, {Y_1}^{r_2})$.

If $C_1 = g^{r_1}$ and $C_6 = H_3(M^{r_1}, C_1, C_2, C_3, C_4, C_5)$ hold, then output M; otherwise, output $\perp$.

Test (CT_A, CT_B, TD_A, TD_B, S'): Let CT_A, CT_B be two ciphertexts which are encrypted by (S_A, S'_A) and (S_B, S'_B) independently such that $T'_A(S'_A) = 1$ and $T'_B(S'_B) = 1$, for

$$CT_A = (S_A, S'_A, C_{A,1}, C_{A,2}, C_{A,3}, C_{A,4}, C_{A,5}, C_{A,6})$$

and

$$CT_B = (S_B, S'_B, C_{B,1}, C_{B,2}, C_{B,3}, C_{B,4}, C_{B,5}, C_{B,6}).$$

Then calculate this algorithm as follows.

For a leaf node t_A, set $j = att(t_A)$. If $j \in S'_A$, then

$$\begin{aligned} DecryptNode(CT_A, T'_{t_A}, t_A) &= e(T'_{t_A}, C_{A,5}) \\ &= e(T'_{t_A}, E_{A,j}) \\ &= e(g^{\frac{q_{t_A}(0)}{t_A}}, g^{r_{A,3} t_{A,j}}) \\ &= e(g, g)^{r_{A,3} q_{t_A}(0)}; \end{aligned}$$

Otherwise, set $DecryptNode(CT_A, T'_{t_A}, t_A) = \perp$.

For a non-leaf node t_A, $DecryptNode(CT_A, T'_{t_A}, t_A)$ calculates as follows.

Let z'_A be a child of t_A. The algorithm $DecryptNode(CT_A, T'_{t_A}, t_A)$ outputs O_{A,z'_A}. Let F_{t_A} be an arbitrary $k_{A,t}$-sized set of child nodes z'_A such that $O_{A,z'_A} \neq\perp$. If no such set exists (the node is not satisfied), then return $\perp$. Otherwise, calculate:

$$\begin{aligned} O_{t_A} &= \prod_{z'_A \in F_{t_A}} Q_{z'_A}^{\Delta_{i,F'_{t_A}}(0)} \\ &= \prod_{i \in F_{t_A}} (e(g,g)^{r_{A,3} q_{t_A}(0)})^{\Delta_{i,F'_{t_A}}(0)} \\ &= \prod_{i \in F_{t_A}} (e(g,g)^{r_{A,3} q_{parent(z'_A)}(index(z'_A))})^{\Delta_{i,F'_{t_A}}(0)} \\ &= \prod_{i \in F_{t_A}} (e(g,g)^{r_{A,3} q_{t_A}(i)})^{\Delta_{i,F'_{t_A}}(0)} \\ &= e(g,g)^{r_{A,3} q_{t_A}(0)} \end{aligned}$$

where $i = index(z'_A)$ and $F'_{t_A} = \{index(z'_A) : z'_A \in F_{t_A}\}$.

Then calculate $DecryptNode(CT_A, T'_{t_A}, r'_A) = {Y_{A,2}}^{r_{A,3}}$ $(= e(g,g)^{y_{A,1} r_{A,3}})$.

Finally, calculate

$$ {M_A}^{r_{A,1}} = \frac{C_{A,3}}{H_2(S'_A, {Y_{A,2}}^{r_{A,3}})}. $$

Use the same method to calculate ${M_B}^{r_{B,1}} = \frac{C_{B,3}}{H_2(S'_B, {Y_{B,2}}^{r_{B,3}})}$.

- If two equality

$$ C_{A,6} = H_3({M_A}^{r_{A,1}}, C_{A,1}, C_{A,2}, C_{A,3}, C_{A,4}, C_{A,5}) \text{ and} $$
$$ C_{B,6} = H_3({M_B}^{r_{B,1}}, C_{B,1}, C_{B,2}, C_{B,3}, C_{B,4}, C_{B,5}) $$

  hold, then compute $e({M_A}^{r_{A,1}}, C_{B,1})$ and $e({M_B}^{r_{B,1}}, C_{A,1})$. Output 1 if $e({M_A}^{r_{A,1}}, C_{B,1}) = e({M_B}^{r_{B,1}}, C_{A,1})$ holds; Output 0, otherwise. Here, $r_{A,1}$, $r_{A,3}$ and $r_{B,1}$, $r_{B,3}$ are the random elements used in the generation of CT_A and CT_B, respectively.
- Otherwise, it outputs $\perp$.

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Int. Conf. Adv. Cryptol.*, 2005, pp. 457–473.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.

[4] 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.

[5] G. Yang, C. Tan, Q. Huang, and D. S. Wong, "Probabilistic public key encryption with equality test," in *Proc. Cryptogr. Track RSA Conf.*, vol. 5985. San Francisco, CA, USA, 2010, pp. 119–131.

[6] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2004, pp. 506–522.

[7] Q. Tang, "Towards public key encryption scheme supporting equality test with fine-grained authorization," in *Proc. 16th Australasian Conf. Inf. Secur. Privacy*, vol. 6812. Melbourne, Australia, 2011, pp. 389–406.

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

[9] Q. Tang, "Public key encryption schemes supporting equality test with authorisation of different granularity," *Int. J. Appl. Cryptogr.*, vol. 2, no. 4, pp. 304–321, 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] L. Wu, Y. Zhang, K.-K. R. Choo, and D. He, "Efficient and secure identity-based encryption scheme with equality test in cloud computing," *Future Generat. Comput. Syst.*, vol. 73, pp. 22–31, Aug. 2017.

[13] H. 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.

[14] D. Boneh, B. Lymn, and H. Shacham, "Short signatures from the weil pairing," in *Advances in Cryptology—ASIACRYPT*, vol. 2248. Berlin, Germany: Springer, 2001, pp. 514–532.

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 cryp-tography from the University of Electronic Science and Technology of China. Her research interests are in the areas of cryptography and information security.

FAGEN LI (M'14) received the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2007. He is currently a Professor with the School of Computer Science and Engineer-ing, University of Electronic Science and Tech-nology of China, Chengdu, China. From 2008 to 2009, he was a Post-Doctoral Fellow with Future University-Hakodate, Hokkaido, Japan, which is supported by the Japan Society for the Promotion of Science. He was a Research Fellow with the Institute of Mathematics for Industry, Kyushu University, Fukuoka, Japan, from 2010 to 2012. His recent research interests include cryptography and network security. He has authored over 80 papers in international journals and conferences.

SHAOQUAN JIANG received the B.S. degree in applied mathematics from the University of Sci-ence and Technology of China, Hefei, in 1996, and the M.S. degree in cryptography from the State Key Laboratory of Information Security, USTC, in 1999, and the Ph.D. degree in electri-cal and computer engineering from the University of Waterloo, Canada, in 2005. He is currently a Professor with Mianyang Normal University, Mianyang. His research interests include public key encryption, information theoretical security, cryptographic protocol, network security, and wireless sensor network security.

SHIJIE ZHOU (M'06) received the Ph.D. degree in computer science and technology from Uni-versity of Electronic Science and Technology of China (UESTC) in 2004. He is currently a Pro-fessor with the School of Information and Soft-ware Engineering, UESTC. His research interests include communication and security in computer networks, peer-to-peer networks, sensor networks, cloud security, and big data.

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. He is currently a Post-Doctoral Staff 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, some protocols for cloud security.