Traceable and Weighted Attribute-Based Encryption Scheme in the Cloud Environment

**XIXI YAN**, **XIAOHAN YUAN**, **QICHAO ZHANG**, AND **YONGLI TANG**

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China

Corresponding author: Yongli Tang (yltang@hpu.edu.cn)

This work was supported by the Projects of Henan Provincial Department of Science and Technology under Grant 192102210280.

**ABSTRACT** In order to solve the problem that the importance of the user’s attribute is seldom considered in the most of the existing attribute-based encryption schemes with traitor tracing, we designed a traceable and weighted attribute-based encryption scheme. In our constructed scheme, the private key of the user consisting of user’s identity information is applied to trace traitors. In addition, the idea of weighted attribute is introduced, and the attributes set is transformed into the segmentation set of weighted attributes through the attributes set segmentation algorithm. Via employing a linear secret sharing scheme, the constructed scheme offers fine-grained and nimble access control mechanism. Under the assumption of q-BDHE in the standard model, we prove that the designed scheme is able to reach security against chosen-plaintext attack.

**INDEX TERMS** Attribute-based encryption, access control, traceable, weighted attribute.

**I. INTRODUCTION**

One-to-many encryption pattern can be achieved by the system of attribute-based encryption (ABE) via fine-grained access control. The thought of ABE was brought up by Sahai and Waters [1]. The earliest opinion of ABE was deemed to fuzzy identity-based encryption. And then the mechanism of ABE was studied by a lot of other workers subsequently. ABE is mainly divided into two classifications: one of them is ciphertext-policy attribute-based encryption (CP-ABE) [2], and the other is key-policy attribute-based encryption (KP-ABE) [3]. Among the scheme of CP-ABE, the attributes set is in relation to the private key of the user, along with the access strategy has relation to the ciphertext of the data. On the contrary, the attributes set is bound up with the ciphertext of the data, as well as the access strategy is interrelated to the private key of the user in the system of KP-ABE. CP-ABE not only protects data privacy and provides fine-grained access control structure, but also approves holders of the data to designate nimble access policies [4], [5]. It is convenient for owners to share data in the cloud system.

In the system of ABE, the holder does not want to disclose the private key thanks to the private key is closely related to the privacy and security of the holder. Nevertheless, because the attributes set is related to the user’s private key in the existing ABE scheme, the same attribute may be assigned to different users, resulting in the same decryption private key held by different users. And the number of attributes is far less than the number of users in practical application. It is usually difficult to determine the real identity of the holder of the private key. As a result, users will sell their private key secretly or be stolen maliciously. Even if the system finds the sold private key, it is troublesome to determine the identity information of the original holder of the private key. Therefore, traitor tracing becomes one of the hot issues in the research of ABE [10]. Imagine a hospital that utilizes a system of CP-ABE. Along with three doctors have their decryption keys which correspond to attributes set, i.e., “{Alice, Doctor, Psychiatric Department}”, “{Bob, Nurse, Psychiatric Department}”, and “{Coral, Doctor, Emergency Department}” respectively. The patient wants to share his medical history by an access policy with attributes, for instance “(Doctor AND (Psychiatric Department OR Emergency Department))” instead of the identity-relevant attributes, for example “(Alice OR Coral)”. In an expressive CP-ABE system, both Alice and Coral can decrypt the ciphertext via their private key. If Bob can decrypt the ciphertext, who helps him or her? Alice or Coral? So when a malevolent authorized user secretly exposes his or her private key, it is hard to determine the original holder of the private key.

Behind that question lurks another question: the doctor can be classified into distinct levels in the real spectacle, such as general doctor, specialist, and so on. So the attributes can be divided into different levels.

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.
as attending doctor, resident doctor, intern doctor, and so on. If we intend to append the whole situation to an access structure, we ought to regard whether all possibilities we have thought over. Hence, the access structure grows relative complicated. We can grant doctor with weight 1, 2, ..., n, denote intern doctor as “Doctor (1)" and let attending doctor represent as “Doctor (2)" respectively, so the weight indicates the importance of the attribute. Both attending doctor and the intern doctor is capable of decrypting the data by using an access structure like {Doctor (1) AND Psychiatric Department}, whereas the intern doctor is unable to employ the access structure like {Doctor (2) AND Psychiatric Department} to decrypt the data. In brief, the weight can symbolize the significance of the attribute in the above circumstances, and weighted ABE is more applicable for the realistic scenario [11], [12]. In real life, the weighted attribute is in worthy consideration. Each user has many attributes, and the same attributes of different users should have different roles. For example, the properties of the two users are the same as that of the engineer, and the access rights they have should be different according to the experience. Engineer with long working experience should have the higher access right. Considering the actual application environment, each attribute plays a different role in the system, the corresponding right in the system is also different, and the status of each attribute may not be the same. It has practical significance to introduce the concept of weight into the ABE system, which makes the ABE scheme closer to the actual situation.

Therefore, a realistic ABE system that holds traceability and the weighted attribute is very critical. In our paper, a weighted attribute-based encryption scheme that holds traitor tracing will be introduced.

A. RELATED WORK

In recent years, ABE has become a research hotspot since Sahai and Waters [1] put forward the earliest opinion of ABE. Lots of ABE systems [2]–[9] were put forward. Li et al. [6] presented a practical extended file hierarchy CP-ABE system that can encrypt multiple documents on the same access level and it’s applicable to large companies and institutions with multi-level departments. In order to better adapt to the leakage of the master key and secret key, Li et al. [7] first gave the formal definition and security model of the hierarchy ABE system with continuous leakage-resilience. And they came up with a ciphertext-policy hierarchy ABE system. Li et al. [8] taken the side-channel attacks into account on most of the existing KP-ABE systems. In their article, they came up with the formal definition and security model of KP-ABE which is resilient to continual auxiliary input leakage and put forward the concrete scheme. Li et al. [9] constructed an ABE scheme with full verifiability for outsourced decryption. In their scheme, it can not only guarantee the correctness of outsourced decryption for the authorized users, but also can check the correctness for transformed ciphertext for the unauthorized users.

The system of ABE controls the user’s authority of decryption via the matching relationship between access structure and attributes set. However, there is a case that the traitor (or the malicious user) obtains illegal interest by disclosing the private key in practical application. Therefore, Abdalla et al. [10] first presented an encryption scheme depended on the identity of traitors. Wang et al. [13] offered an ABE scheme that could trace traitors based on identified users. In their scheme, a secure coding technique was introduced to guarantee the identity of abusers of the key. In general, traceable CP-ABE is capable of realizing the fine-grained access control mechanism. Yet the majority of the current CP-ABE schemes that hold traceable are limited to AND gate and less expressive. For the sake of boosting the expressive capability of the access strategy, Liu et al. [14] constructed a CP-ABE scheme that holds traceable and contains any monotonic access structure. In their scheme, a signature technique was used to generate identity information so that the CP-ABE system can trace malevolent users. As well as they divided the traceability of ABE into two classifications, one of them namely white-box traceability, and the other is black-box traceability. Liu et al. [15] put forward a black-box traceable CP-ABE. Ning et al. [16] came up with a CP-ABE scheme that holds the white-box traceable and contents nimble attributes. Li et al. [18] and Zhou et al. [19] put forward a multi-authority CP-ABE scheme of which polices were expressed in any monotonic access structure and holds traceable. Zhang et al. [20] came up with a CP-ABE scheme with large-universe multi-authority, as well as supported white-box traceability. Jiang et al. [21] came up with a scheme of CP-ABE to hinder the abuse of private key delegation of the user, and broadened it to a CP-ABE system with traceable. In addition, Ning et al. [22] offered a CP-ABE scheme and achieved both public auditing and traceability. The scheme of Guan et al. [23] realized traceability and brought about reliable key delegation. Li et al. [24] provided a CP-ABE scheme with traceable and the outsourced decryption with verifiable was realized at the same time. Wang et al. [25] came up with a CP-ABE scheme with traceable, as well as revocation of the user. Ning et al. [26] and Liu et al. [27] came up with a scheme of CP-ABE and supported both traceable and revocable.

Although these schemes can be employed to detect malevolent users, they do not consider the importance of attributes and they cannot support the weighted attribute. Considering the actual application environment, Liu et al. [11] presented the idea of ciphertext-policy weighted attribute-based encryption (CP-WABE), as well as they nominated the weighted attribute into the system of ABE. Chu and Wang [28] constructed a weighted ABE with revocation, but the scheme could not trace malicious users.

B. OUR CONTRIBUTION

Considering that it is difficult for malicious users to be traced after selling their private keys and aiming at the problem that the importance of attribute is seldom considered in the most
of the existing traceable ABE scheme, our scheme introduces the concept of the weighted attribute into the traceable ABE scheme. Based on the comprehensive analysis of the existing CP-ABE scheme, we find that many researchers pay attention to tracing malicious users and weighted attributes independently. Inspiring by the scheme [11], [12], [16], [17], we construct a traceable and weighted attribute-based encryption system. The primary contributions of our constructed scheme are as below.

(1) In our constructed scheme, the malevolent users who expose their private keys for whatever intention can be distinguished through the tracking algorithm. The presented scheme randomizes the identity and the result is embedded into the private key of the user. The traitor can be tracked via tracking the user’s identity information.

(2) By introducing the concept of the weighted attribute into the traceable ABE scheme, the flexibility of attribute expression in the system is improved and the complexity of access tree is reduced. According to the importance of attributes in the system, the trusted authority allocates different weights. Based on the weight of attributes, the attributes set is transformed into the segmentation set of weighted attributes via the attributes set segmentation algorithm. By using linear secret sharing scheme as access structure, fine-grained access control is provided. Because each attribute of the users is assigned weight, only when the attribute contained in the secret key satisfies the access structure of the ciphertext can it be decrypted correctly. Comparing with the related traceable ABE scheme, it is of practical significance to introduce the concept of weight into the traceable ABE scheme, so that our scheme is closer to the actual situation.

(3) Under the assumption of q-BDHE, the constructed scheme is proved to realize security against the chosen-plaintext attack in the standard model. The experimental results show that the scheme is practical in the cloud environment.

C. ORGANIZATION

The backgrounds which consist of bilinear map, access structure, and complexity assumptions are given in the next section. Section 3 gives a formal definition. Our main scheme is represented in section 4. Section 5 represents the security analysis of our constructed scheme. In section 6, the concrete analysis is discussed. In the end, we summarized our constructed scheme in the last section.

II. PRELIMINARIES

A. BILINEAR MAP

Make \( G \) and \( G_T \) denote two multiplication cyclic groups with the order of prime \( p \), and a generator of \( G \) is \( g \). \( e : G \times G \to G_T \) is a bilinear map that contains:

1. Bilinearity: For \( \forall u, v \in G \) and \( a, b \in Z_p \), there is \( e(u^a, v^b) = e(u, v)^{ab} \).

2. Non-degeneracy: \( e(g, g) \neq 1 \).

3. Computability: For \( \forall u, v \in G \), there exists a method that can calculate \( e(u, v) \).

B. ACCESS STRUCTURE

Definition 1: Make \( \{P_1, P_2, \cdots, P_n\} \) indicate a aggregation of participants. An aggregation \( A \subseteq 2^{\{P_1, P_2, \cdots, P_n\}} \) is called monotonie if \( B \in A \) and \( B \subseteq C \) then \( C \in A \). Let \( A \) denote a (monotonic) aggregation of the non-empty sub-aggregations of \( \{P_1, P_2, \cdots, P_n\} \), i.e., \( A \subseteq 2^{\{P_1, P_2, \cdots, P_n\}} \backslash \{\emptyset\} \). Then the non-empty sub-aggregation \( A \) is a (monotonic) access structure. The subsets of \( A \) are considered as the authorized aggregations, otherwise, the rest of the sets are considered as the unauthorized aggregations.

C. LINEAR SECRET SHARING SCHEME (LSSS)

Definition 2: A LSSS over an aggregation of participants \( P \) can be considered as follows:

1. Each secret share comes from a vector in \( Z_p \).

2. There has a matrix \( M \) with \( l \) rows and \( n \) columns which refers to the shared generation matrix. We describe a function \( \rho \) for \( i \in [1, l] \) that maps the \( i \)-th row of the matrix \( M \) to attribute \( \rho(i) \). We think over a column vector \( v = (s, v_2, v_3, \cdots, v_n) \), where \( s \in Z_p \) is a secret value, and we randomly choose \( v_2, v_3, \cdots, v_n \in Z_p \). Then \( M \cdot v \) represents a vector of \( l \) share of \( s \), as well as the share \( (M \cdot v)_i \) relates to attribute \( \rho(i) \). Each LSSS has the feature of linear reconstruction. Make \( A \) denote an access structure, \( S \in A \) indicates any authorized aggregation, along with make \( I \subseteq \{1, 2, \cdots, l\} \) be illustrated as \( I = \{i : \rho(i) \in S\} \). There exists an aggregation of constants \( \{\omega_i\}_{i \in I} \) that make \( \sum_{i \in I} \omega_i \cdot M_i = (1, 0, \cdots, 0) \) true. If \( \lambda_i = M_i \cdot v \) is a valid share of any \( s \), after that \( \sum_{i \in I} \omega_i \lambda_i = s \).

D. COMPLEXITY ASSUMPTIONS

We offer the assumption of l-Strong Diffie-Hellman (l-SDH) and present the assumption of q-Bilinear Diffie-Hellman Exponent (q-BDHE) in this section.

Assumption 1 (l-SDH Assumption): Make \( G \) designate a group with the order of prime \( q \), as well as make \( g \) mean a generator of \( G \). A problem of l-SDH can be presented as follows. After selecting exponent \( x \in Z_p \) casually and giving a \( l + 1 \) tuple \( (g, g^x, g^{x^2}, \cdots, g^{x^l}) \), it produces a pair \( (c, g^{l/(x+i)}) \in Z_p \times G \). The algorithm \( A \) can deal with the problem of 1-SDH with the advantage \( \varepsilon \) if \( \Pr \{A\left(g, g^x, g^{x^2}, \cdots, g^{x^l}\right) = (c, g^{l/(x+i)}) \} \geq \varepsilon \).

Definition 3: The assumption of l-SDH contents if no polynomial-time algorithm \( A \) is able to settle the problem of l-SDH with non-negligible advantage.

Assumption 2 (q-BDHE Assumption): Make \( G \) designate a group, the prime order is \( p \), and \( g \) designates a generator of \( G \). A problem of q-BDHE can be concretely represented as follows. Casually chooses exponent \( d, s \in Z_p \) and given...
\[ \vec{y} = \left(g, g^d, g^{d^2}, \ldots, g^{d^q}, g^{d^{q+2}}, \ldots, g^{d^{2q}} \right) \] it is hard for the algorithm \( A \) to make a distinction between \( e(g, g)^{d^{q+1}S} \in G_T \) and the random component \( Z \in G_T \). The algorithm \( A \) can handle the problem of q-BDHE with the advantage \( \varepsilon \) if
\[
\left| \Pr \left[ A \left( \vec{y}, T = e(g, g)^{d^{q+1}} \right) = 0 \right] - \Pr \left[ A \left( \vec{y}, T = Z \right) = 0 \right] \right| \geq \varepsilon.
\]

**Definition 4:** The assumption of q-BDHE contents if no polynomial-time algorithm \( A \) has a non-negligible advantage in handling the problem of q-BDHE.

### III. DEFINITIONS AND SECURITY MODEL

#### A. DEFINITION OF THE ALGORITHMS

Inspiring by references [11], [12], [16], [17], a traceable and weighted CP-ABE scheme which involves seven algorithms is constructed.

1. **Attributes set segmentation algorithm.** \( \text{Seg}(\Gamma) \rightarrow U \)

   Taking system attributes set \( \Gamma \) as input, it segments attributes set according to the maximum weight of the attribute and outputs the segmented set \( U \).

2. **Initialization algorithm.** \( \text{Init}(\Gamma, U) \rightarrow (\text{MSK}, \text{PP}) \)

   By inputting a security parameter \( \lambda \) and segmented set \( U \), it calculates the system master key \( \text{MSK} \), as well as public parameters \( \text{PP} \).

3. **Key generation algorithm.** \( \text{KeyGen}(\text{PP}, \text{MSK}, \text{S}, id) \rightarrow \text{SK} \)

   After inputting the user’s identity \( id \), attributes set \( S \) of the user, system master key \( \text{MSK} \), along with public parameters \( \text{PP} \), it produces a private key \( \text{SK} \).

4. **Encryption algorithm.** \( \text{Enc}(\text{PP}, m, (M, \rho)) \rightarrow \text{CT} \)

   After entering public parameters \( \text{PP} \), message \( m \), access policies \( (M, \rho) \), it produces ciphertext \( \text{CT} \).

5. **Decryption algorithm.** \( \text{Dec}(\text{PP}, \text{CT}, \text{SK}) \rightarrow m/\bot \)

   Through inputting public parameters \( \text{PP} \), ciphertext \( \text{CT} \), as well as private key \( \text{SK} \), it outputs message \( m \) or failure symbol \( \bot \).

6. **Key sanity check algorithm.** \( \text{KeySan}(\text{PP}, \text{SK}) \rightarrow 1 \text{ or } 0 \)

   By inputting public parameters \( \text{PP} \) and private key \( \text{SK} \), if \( \text{SK} \) meets the conditions of the phase of key sanity check, then it outputs 1; otherwise, it outputs 0.

7. **Tracking algorithm.** \( \text{Track}(\text{PP}, \text{SK}) \rightarrow id/\bot \)

   If the conditions of key sanity check are satisfied, it enters the public parameters \( \text{PP} \), the private key \( \text{SK} \), and generates the identity \( id \) of the user or the failure symbol \( \bot \).

#### B. SELECTIVE CPA SECURITY MODEL

A security model is an indistinguishability game against chosen-plaintext attack (IND-CPA) described between an adversary and a challenger.

**Initialization:** After declaring an access structure \( (M^*, \rho^*) \) that the attacker will challenge, the attacker transmits it to the challenger.

Setup: After carrying out the Initialization algorithm, the challenger transfers \( \text{PP} \) to the adversary.

**Phase 1:** The adversary requires the challenger to obtain the private keys of the user associated with \((id_1, S_1), \ldots, (id_q, S_q)\) without any contending the access policy \((M^*, \rho^*)\).

**Challenge:** The adversary offers messages \( M_0 \) and \( M_1 \) with equal-length to the challenger, along with an access structure \((M^*, \rho^*)\). The challenger selects a \( b \in \{0, 1\} \). After encrypting \( M_b \) to generate \( \text{CT}^* \) with \((M^*, \rho^*)\), the challenger transmits \( \text{CT}^* \) to the adversary.

**Phase 2:** The adversary queries the challenger to get the private keys of the user relevant to \((id_1, S_1), \ldots, (id_q, S_q)\) with the same regulation in Phase 1.

**Guess:** The adversary generates a guess \( b' \) for \( b \).

Via providing \( b' = b \), the adversary wins the game. The game requires that none of the inquired attributes set \( S_1, \ldots, S_q \) can contain \((M^*, \rho^*)\). The advantage of the adversary of winning the game is described as \( |Pr[b' = b] - 1/2| \).

**Definition 5:** Our constructed scheme is secure against IND-CPA if all polynomial-time attackers have a negligible advantage in winning the above indistinguishability game.

### C. TRACEABILITY MODEL

We can employ the following game that runs between a challenger and an adversary to prove the traceability of our constructed scheme.

**Initialization:** After carrying out the Initialization algorithm, the challenger transfers the public parameters \( \text{PP} \) to the adversary.

**Key Query:** The adversary asks the challenger to get the private keys of the user relevant to \((id_1, S_1), \ldots, (id_q, S_q)\).

**Key Forgery:** After querying, the adversary produces the private key \( \text{SK}^* \) of the user.

Via providing \( \text{Track}(PP, SK^*) \rightarrow id_1/\bot \) and \( \text{Track}(PP, SK^*) \notin \{id_1, \ldots, id_q\} \), the adversary can win the above game. The advantage of the adversary winning the above game is represented as \( Pr[\text{Track}(PP, SK^*) \notin \{\bot \} \cup \{id_1, \ldots, id_q\}] \).

**Definition 6:** Our constructed scheme holds traceability if all polynomial-time attackers have a negligible advantage in winning the above traceability game.

### D. KEY SANITY CHECK MODEL

We can employ the following game that runs between a simulator and an adversary to describe the model of key sanity check according to the literature [16]. When the security parameter \( \Gamma^k \) is inputted, the simulator calls an adversary and returns the public parameters \( \text{PP} \), the ciphertext \( CT \), and two different keys \( \text{SK}_{id_k, S_k}, \tilde{SK}_{id_k, S_k} \) corresponding to the same attributes set \( S_k \) for a user \( U_k \) with \( id_k \) to the adversary \( A \). \( A \) will win the game if the following requirements are met.

1. **KeySan** \( \text{PP}, \text{SK}_{id_k, S_k} \rightarrow 1. \)
2. **KeySan** \( \text{PP}, \tilde{SK}_{id_k, S_k} \rightarrow 1. \)
3. \( \text{Dec}(\text{PP}, \text{CT}, \text{SK}_{id_k, S_k}) \neq \bot. \)
4. \( \text{Dec}(\text{PP}, \text{CT}, \tilde{SK}_{id_k, S_k}) \neq \bot. \)
5. \( \text{Dec}(\text{PP}, \text{CT}, \text{SK}_{id_k, S_k}) \neq \text{Dec}(\text{PP}, \text{CT}, \tilde{SK}_{id_k, S_k}). \)
IV. SPECIFIC CONSTRUCTION

1) ATTRIBUTES SET SEGMENTATION ALGORITHM

Seg(Γ) → U. The algorithm is run by a trusted center TA. After inputting the system attributes set Γ, TA assigns different weights based on the importance of attributes in our proposed system. Each attribute $\lambda_i$ is assigned its maximum weight $\theta_i = \text{weight}(\lambda_i)$ (where $\theta_i$ is an integer only) that is allowed in the system. Trusted center TA segments $\lambda_i$ according to their weights. After segmenting, $\lambda_i$ is corresponded to $(\lambda_i, 1) \ldots (\lambda_i, \theta_i)$ and system attributes collection Γ is transformed into attributes segmentation set U of all attributes. Finally, the algorithm outputs U. For example, the position of the teacher in a university can be divided into three categories: Lecturer, associate professor, and professor. They are given weights of 1, 2 and 3 respectively, that is, they are represented by “teacher: 1”, “teacher: 2”, and “teacher: 3” respectively. In this case, only using this attribute, i.e., “teacher”, can achieve the effect of the original three attributes. If there is an access structure like {Professor OR Associate professor OR Lecturer}, its structure can be reduced to {teacher: 1}, that is, the lowest attribute satisfying the policy is used to replace other attributes. In our scheme, we use weighted attribute to improve the expression ability of attribute and simplify the access structure.

2) INITIALIZATION ALGORITHM

Init$^2$ INITIALIZATION ALGORITHM access structure. In this case, only using this attribute, i.e., “teacher”, can improve the expression ability of attribute and simplify the access structure.

2) INITIALIZATION ALGORITHM

Init$^2$ (U, $\lambda$) → (MSK, PP). The algorithm is carried out by TA. TA enters the security parameter $\lambda$ and the attributes segmentation set U. Make $G$ and $G_T$ represent two multiplication cyclic groups. The order is prime $p$, and $g$ is a generator of $G$. $e : G \times G \rightarrow G_T$ is a bilinear map. Let $\zeta = \sum \theta_i$. The trusted center TA implements the following procedures:

1. It naturally chooses $a, \alpha \in Z_p, h \in G$.
2. For each attribute $z$, it selects $U_z \in G (U_z$ is related to the attributes segmentation set $U$).
3. The trusted center TA chooses a probabilistic encryption algorithm (Enc, Dec). This algorithm is a symmetric encryption with a secret key $\vec{k} \in Z_p$. After encrypting the same plaintext, the algorithm can generate the diverse ciphertext.

Finally, the TA publishes $PP = \{p, e, G, G_T, g, h, g^\alpha, e(g, g)^{\alpha} \cdot \{U_z\}_{z \in U}\}$ and lets the MSK $= \{a, \alpha, \vec{k}\}$ secret.

3) KEY GENERATION ALGORITHM

KeyGen$^P$(PP, MSK, $S, id$) → SK. The trusted center TA produces the decryption key SK. The trusted center TA authenticates the attributes set $S_k (S_k \in U)$ of the user $u_k$ and generates SK. The specific procedures are as below.

The trusted center TA casually selects $r_i, r_z \in Z_p$ and computes $c = \text{Enc}_z(id_z)$ for each $z \in S_k (1 \leq z \leq |U|)$, where $id_z$ is the identity of the user $u_k$. We can see that there is no difference between the result $c$ and a random number in $Z_p$.

$TA$ produces $SK = \left\{ K_1 = c, K_2 = g^{\alpha(a+c)h'}, L_1 = g^r, L_2 = g^{ar}, \{K_{1,1} = g^{e}, K_{2,2} = U_z^r\}_{z \in S_k} \right\}$.

4) ENCRYPTION ALGORITHM

Enc $^P$ (PP, $m, (M, \rho)$) → CT. Via encrypting the message $m$, the data owner obtains ciphertext CT. The specific operations are as follows. The data owner inputs public parameters PP, message $m$, access policies $(M, \rho)$, where $M$ is a matrix that has $l \times n$ elements. And the function $\rho$ maps the row of the matrix $M$ into the smallest share of the weighted attributes, i.e., “teacher: 1”. Then, it selects a random column vector $\nu = (s, v_2, v_3, \ldots, v_n)$, $v_2, v_3, \ldots, v_n$ are employed for sharing the encryption factor $s$. The data user computes $\lambda_i = M_i \cdot \nu$ for $\forall i \in [1, l]$, where $M_i$ is a vector associated with the $i$-th row of the matrix $M$. After casual selecting an exponent $\tau_i \in Z_p$, the data owner outputs CT as follows:

$CT = \left\{ (M, \rho), C = m \cdot e(g, g)^{\alpha}, C_0 = g^\alpha, C_1 = g^{\alpha}, \{C_{1,1} = h^{\delta_i}, C_{1,2} = U^{\nu_i \cdot h^\rho}, C_{1,3} = g^s\}_{i \in [1, l]} \right\}$.

5) DECRYPTION ALGORITHM

Dec $^P$ (PP, CT, SK) → $m$ | $\perp$. For the user $u_k$, there exist two circumstances:

(1) If the user’s attributes aggregation $S_k \neq (M, \rho)$, i.e., the attributes aggregation $S_k$ of the $u_k$ is unable to fulfill the access policy $(M, \rho)$, the algorithm aborts.
(2) If the user’s attributes aggregation $S_k \in (M, \rho)$, the algorithm sets $I = \{i : \rho(i) \in S\}, I \subseteq [1, l]$. As well as there has a aggregation of constants $\{\omega_i \in Z_p\}_{i \in [1, l]}$ that can make $\sum_{i \in I} \omega_i = s$ true.

The algorithm computes as below:

$D = \left[ e(l_1^{K_1'}, L_2, C_{1,1})e(K_{1,2}, C_{1,2})e(K_{1,3}, C_{1,3}) \right]^{\omega_i}$

$e(g, h)^{(a+c)\rho}$

$E = e \left( K_2', C_0^{K_1'} \cdot C_1 \right) = e(g, g)^{\alpha} e(g, h)^{(a+c)\rho}$

$F = \frac{E}{D} = (g, g)^{\alpha}$

In the end, via computing $m = C/F$, the algorithm obtains the message $m$.

6) KEY SANITY CHECK ALGORITHM

KeySan $^P$ (PP, SK) → 1 or 0. The key SKk of the user is checked by the trusted center TA to determine whether it satisfies the conditions of key sanity check algorithm, the specific procedures are as follows:

1. Check the form of the key $SK_k$ with $SK = \left\{ K_{1,1}', K_{2,1}, L_1, L_2, \{K_{1,1}, K_{2,2}\}_{z \in S_k} \right\}$ and $K_{1,1} \in Z_p$, $K_{2,1}, L_1, L_2, K_{1,1}, K_{2,2} \in G$.
2. Check $e(L_2, g) = e(L_1, g^\alpha)$.
3. Check $e \left( K_{2,1}', g^{(a+c)\rho} \right) = e(g, g)^{\alpha} e(L_2L_{1,1}', h)$.
4. Check $\exists z \in S_k, s.t.e \left( K_{2,2}, g \right) = e(U_z, K_{2,1}) \neq 1$. 

VOLUME 8, 2020

38289
The algorithm outputs 1 if the key SK_k contains the conditions of key sanity check, otherwise, it outputs 0.

7) TRACKING ALGORITHM

Track (PP, SK) \rightarrow id \perp. TA conducts this phase. If the key SK_k of the user is unable to pass the procedure of key sanity check, i.e., KeySan (PP, SK) \rightarrow 0, then the algorithm outputs \perp; otherwise, the algorithm extracts identity information id_k from the key SK_k of the user via Dec_k (K'_1).

V. SCHEME ANALYSIS

A. SECURITY ANALYSIS

Theorem 1: There is no probabilistic polynomial time (PPT) adversary who can break our scheme with non-negligible advantage under chosen-plaintext attack if the q-BDHE assumption holds, where q > |U| - 2, |U| is the amount of users of our constructed system.

Proof: We can suppose that there is a PPT adversary A who has the advantage of \epsilon in winning this game. And in this circumstance, we are able to build a simulator B who is capable of breaking the q-BDHE problem with the advantage of \epsilon/2. The simulation actions are stated as below.

Let G and \mathcal{G} indicate two multiplication cyclic groups. The order is prime p. g represents a generator of G and e : G \times G \rightarrow \mathcal{G} is a bilinear map. Given \bar{y} = (g^s, g^{sd}, \ldots, g^{s\rho}, g^{s\rho^2}, \ldots, g^{s\rho^k}), the simulator B casts a fair coin \mu. If \mu = 0, B defines T = e(g, g)^{\rho^{q+1}}; otherwise, B defines T = Z, where Z is a random component in \mathcal{G}.

Initialization: Adversary A chooses an access policy (M^*, \rho^*), M^* is a l \times n matrix and n \leq q.

Attributes set segmentation: According to the weight \theta_i, the simulator B divides each attribute \lambda_i into the attributes set \Gamma and obtains the weighted segmented set U. Let u = \sum \theta_i.

Setup: The simulator B carries out the following procedures.

B casually chooses a' \in Z_p that meets e(g, g)^{a'} = e(g, g)^{a'} e(g^d, g^{d^k}), where \alpha = a' + d^{q+1}.

B casually selects a value \zeta_1 \in Z_p for each attribute \lambda \in U (1 \leq \lambda \leq |U|), and then each component of the group U_\lambda \in G is produced as below. And then if there has i \in \{1, 2, \ldots, l\} that can make \rho^*(i) = x true, B sets U_\lambda = g^{\rho^*(i) a} g^{d^{\rho^*/2}} \ldots g^{d^{\rho^*/\lambda}} M_i^{\rho^*}. Otherwise, B sets U_\lambda = g^{\rho^*}. \rho^* is a single-shot function, i.e., for any \lambda, there at most exists one i that can make \rho^*(i) = x true.

B casually chooses a \in Z_p, then computes g^{\alpha'} and sets h = g^{d^k}.

The simulator B generates PP = \{p, e, G, \mathcal{G}, g, h, e(g, g)^a, \langle U_\lambda \rangle_{\lambda \in U}\} and MSK = \{a, \alpha, \bar{\kappa}\}.

Phase 1: The adversary A submits S and requests the corresponding decryption key, where S does not contain M^*.

The simulator B performs the following operations:

B finds a vector \bar{w} = (\omega_1, \omega_2, \ldots, \omega_p) \in Z_p^n that can make \omega_1 = -1 true. For \rho^*(i) \in S, there exists M_i^\rho \cdot \bar{w} = 0.

As we all know that there exists such a vector according to the concept of LSSS.

After selecting random numbers c, r_i' \in Z_p, B defines K'_1 = c.

After selecting a random number \iota \in Z_p, B makes r' = \frac{1}{\alpha \Gamma} (t + \omega_1 d^q + \omega_2 d^{q-1} + \ldots + \omega_p d^{q-k+1}).

B computes K''_1, L'_1, L''_1, K'_1, as below:

\[
K'_2 = g^{(a' + d)(\alpha + c)} \prod_{i=2}^{n} g^{d\rho_{i-1}(\alpha + c)} = g^{a' + d} h' \cdot r',
\]

\[
L'_1 = g^{d^2(\alpha + c)} \prod_{i=1}^{n} g^{d\rho_{i-1}(\alpha + c)} = g^{\rho'},
\]

\[
L''_1 = g^{d^2(\alpha + c)} \prod_{i=1}^{n} g^{d\rho_{i-1}(\alpha + c)} a^{(\alpha + c)} = g^{ar'},
\]

\[
K''_1, = g^{\rho'},
\]

If there is no exists i that can make \rho^*(i) = x true for \forall x \in S, then make K''_1 = g^{\rho'^*c}. Otherwise, B sets K''_1 = g^{\rho'^*c} \prod_{i=1}^{n} g^{d\rho_{i-1}(\alpha + c)} = g^{\rho'^*c} \prod_{i=1}^{n} g^{d\rho_{i-1}(\alpha + c)} = U_i'.

Challenge: B builds the challenge ciphertext. After adversary A enters messages m_0, m_1 with equal-length to B, B casts a coin v \in \{0, 1\} and performs the following operations.

The simulator B computes C = m_v \cdot T \cdot e(g^{\alpha'}, g^{\alpha'}) = c = g^{\alpha'},

and C_1 = (g^v a'). Then B randomly selects y_2, y_3, \ldots, y_{n'} \in Z_p and uses \bar{v} = (s, sd + y_2, sd^2 + y_3, \ldots, sd^{n'-1} + y_{n'}) \in Z_p for sharing the secret value.

\[
C_{i, 1} = \prod_{j=2}^{n'} \left( g^{d^k} \right)^{y_j} M_i^\rho \prod_{j=1}^{n'} \left( g^{d^k} \right)^{-y_j},
\]

\[
C_{i, 2} = (g^{\rho^*(0)})^{-ad} \prod_{j=1}^{n'} g^{d\rho_{i-1}^j} \prod_{j=1}^{n'} g^{d\rho_{i-1}^{-j}},
\]

\[
C_{i, 3} = g^{a' d}. \]

Phase 2: This phase is similar to Phase 1.

Guess: A produces a guess v'. If v' = v, the simulator B outputs \mu' = 0 which indicates T is a valid tuple of q-BDHE. Otherwise, B generates \mu' = 1 which indicates T is a random component in \mathcal{G}. The decryption key and public parameters which are produced by simulating in the above game are similar to those in the real system.

The adversary A is unable to obtain any information on v when \mu' = 1. And we can get \Pr[v' \neq v' | \mu = 1] = \frac{1}{2}. B guesses \mu' = 1 when v \neq v', we can obtain \Pr[\mu = \mu' | \mu = 1] = \frac{1}{2}.
When $\mu' = 0$, $A$ receives ciphertext of $m_0$. If the advantage of $A$ is $\epsilon$, we can get $Pr[\nu = v' | \mu = 0] = \frac{1}{2} + \epsilon$. $B$ guesses $\mu' = 0$ as $v = v'$, and we obtain $Pr[\mu = \mu' | \mu = 0] = \frac{1}{2} + \epsilon$.

In the game of q-BDHE, the advantage of the simulator $B$ is described as

$$Adv = \left| \frac{1}{2} Pr[\mu = \mu' | \mu = 0] + \frac{1}{2} Pr[\mu = \mu' | \mu = 1] \right| - \frac{1}{2} = \frac{1}{2} \left( \frac{1}{2} + \epsilon \right) = \frac{1}{2} \frac{1}{2} \frac{1}{2} = \frac{1}{2} \epsilon.$$

$A$ is able to win the game with a non-negligible advantage, hence, the simulator $B$ is also able to simulate the game of q-BDHE with non-negligible advantage.

**B. TRACEABILITY**

**Theorem 2**: Suppose $q < l$, where $q$ is the amount of key asks built by the adversary, our scheme is traceable if the assumption of 1-SDH holds.

**Proof**: We are able to suppose that there is a PPT adversary $A$. $A$ is capable of winning this game with the advantage of $\epsilon$. And we let $l = q + 1$. As well as we can set up a PPT algorithm $B$. $B$ is capable of breaking the hardness problem of 1-SDH with non-negligible advantage.

Make $G$ and $G_T$ represent two multiplication cyclic groups. And the order of them is prime $p$. Let $g$ be a generator of $G$, and make $e : G \times G \rightarrow G_T$ be a bilinear map. To break the 1-SDH problem, the algorithm $B$ is able to simulate an entity that the adversary $A$ wants to challenge.

After inputting $(\tilde{g}, \tilde{g}^a, \tilde{g}^{a_2}, \ldots, \tilde{g}^{a_q})$, where $a \in Z_p^*$, the algorithm $B$ can produce a tuple $(c_r, \omega_r = \tilde{g}^{1/(a+c)}_r)$ that makes $A_l = \tilde{g}^a_r (i = 0, 1, \ldots, l)$ true. The specific operations are stated below.

**Initialization**: The algorithm $B$ casually selects $q$ different values $c_1, c_2, \ldots, c_q \in Z_p^*$. And make $f(y) = \sum_{i=0}^{q-1} a_i y^i$, where $a_0, a_1, \ldots, a_q \in Z_p^*$ are the coefficient of the polynomial $f(y)$. Later $B$ performs subsequent algorithms.

Set $g = \prod_{i=0}^{q} (A_i)^{a_i} = \tilde{g}^{f(a)}_r, g^{a_1} = \prod_{i=0}^{q} (A_i)^{a_i-1} = \tilde{g}^{f(a)-a}$.

After casual selecting $\alpha, \theta \in Z_p$ and $h \in G$, $B$ chooses a random number $u_k \in Z_p$ and computes $U_x = \tilde{g}^u_x$ for each attribute $x$. In the end, $B$ releases $PP = \left\{ p, e, G, G_T, g, h \right\}$. The master key is $MSK = \left\{ a, \alpha, \tilde{k} \right\}$.

**Key Query**: Adversary $A$ offers a set of attributes $(id_l, S_l)$ to $B$. After that $A$ inquires the relevant decryption keys. When $A$ goes on the $i$-th query, we set $i \leq q$ and let $f_i(y) = \sum_{j=0}^{q-1} y^{j}$. Then we can get $f_i(y) = \sum_{j=0}^{q-1} Y^{j}$. $B$ computes $\sigma_i = \prod_{j=0}^{q-1} (A_j)^{\beta_j} = \prod_{j=0}^{q-1} (\tilde{g}^{a_j})^{\beta_j} = \tilde{g}^{f(a)/(a+c)}_i = \tilde{g}^{1/(a+c)}_i$.

After selecting $r, r_k \in Z_p$ casually, $B$ computes $SK_i = \left\{ K'_1 = c_i, K'_2 = (\sigma_i)^{h'}_1, L_1 = g^{c_i}, L_2 = g^{a_i}r^{1}, [K_{z,1} = g^{c_i}, K_{z,2} = U^{r_{K_{z,1}}} z \in S_k] \right\}$.

**Key Forgery**: Adversary $A$ offers a forged decryption key $SK^* \rightarrow B$. And make $e_A$ denote the case that $A$ wins the game, i.e., $SK^*$ meets the conditions of key sanity check and $K'_1 \notin \{ c_1, c_2, \ldots, c_q \}$. If the case $e_A$ does not occur, $B$ chooses a random tuple $(c_r, \omega_r) \in Z_p \times G$ as a solution to the hardness problem of 1-SDH. If the case $e_A$ takes place, $B$ rewrites $f(y)$ to $f(y) = \gamma(y) \cdot (y + K'_1) + \gamma_{-1}$, where $\gamma(y) = \prod_{i=0}^{q-1} (\gamma_i y^i)$, $\gamma_{-1} \in Z_p^*$. Due to $f(y) = \prod_{i=0}^{q-1} (y + c_i), c_i \in Z_p^*$, and $K'_1 \notin \{ c_1, c_2, \ldots, c_q \}$, $f(y)$ is unable to be divided by $(y + K'_1)$ according to the basic theorem of algebra. $B$ computes the tuple $(c_r, \omega_r) \in Z_p \times G$ according to the subsequent operations. Suppose $L_1 = g^{e_A}, h \in Z_p$. On the basis of the equality $e(L_2, g) = e(L_1, g^{e_A})$ in the phase of key sanity check, we can obtain $L_2 = g^{e_A}$. In line with the equality $e \left( K'_2, g^{a_i}g^{K_{z,1}} \right) = e(g, g)^{e_A} e \left( L_2L_{K_{z,1}}^{K_{z,2}}, h \right)$, we can get $K_2 = g^{e_A(a+c)/h'}$. Then $B$ performs the subsequent operations.

$$\sigma = (K_2/L_1)^{K_1} = g^{1/(a+c)} = \tilde{g}^{\gamma\omega_1}/g^{(\gamma_{-1}/(a+c))}$$

$$\omega_r = (\sigma \cdot \prod_{i=0}^{q-1} A_i^{a_i - \gamma_i})^{1/(\gamma_{-1})} = \tilde{g}^{\gamma\omega_1}/g^{(\gamma_{-1}/(a+c))}$$

$$e \left( g^{a_i}, \tilde{g}^{a_i}, \omega_r \right) = e \left( g^{a_i}, \tilde{g}^{a_i}, \tilde{g}^{1/(a+c)} \right) = e \left( g, \tilde{g} \right).$$

The tuple $(c_r, \omega_r)$ is a solution to the problem of 1-SDH. Thus, the theorem is proved successfully.

**C. KEY SANITY CHECK**

**Theorem 3**: The advantage of the adversary $A$ in winning the game of key sanity check is negligible.

**Proof**: We refer to the proof method of literature [16]. The ciphertext returned to the adversary $A$ is $CT = \left\{ (M, \rho), C, C_0 = g^a, C_1 = g^{a_1}, C_1, C_2, C_3 \right\}_{\in [1, l]}.2$ and two different keys are $SK_{id_A, S_k} = \left\{ K'_1, K_2, L_1, L_2, K_{z,1}, K_{z,2} \right\}_{z \in S_k}$. $A$ will win the game if the following requirements are met. Requirements (1)-(5):

1. **KeySan** $(PP, SK_{id_A, S_k}) \rightarrow 1$.
2. **KeySan** $(PP, \tilde{SK}_{id_A, S_k}) \rightarrow 1$.
3. **Dec** $(PP, CT, SK_{id_A, S_k}) \neq \perp$.
4. $(PP, CT, \tilde{SK}_{id_A, S_k}) \neq \perp$.
5. $(PP, CT, SK_{id_A, S_k}) \neq Dec(PP, CT, \tilde{SK}_{id_A, S_k})$.

Requirements (1) implies:

1. $(K_1' \in Z_p^*, K_2', L_1, L_2, K_{z,1}, K_{z,2} \in G)$.
2. $e(L_2, g) = e(L_1, g^{e_A})$.
3. $e \left( K'_2, g^{a_i} \right) = e \left( g, g \right)^{e_A} e \left( L_2L_{K_{z,1}}^{K_{z,2}}, h \right)$.
4. $\exists z \in S_k, s.t. e \left( K_{z,2}, g \right) = e \left( L_z, K_{z,1} \right) \neq 1$.  

VOLUME 8, 2020
Requirements (2) implies similarly:

1. \( \tilde{K}_1 \in Z_p^* \), \( \tilde{K}_2, \tilde{L}_1, \tilde{L}_2, \tilde{K}_{c,1}, \tilde{K}_{c,2} \in G \).
2. \( e(\tilde{L}_2, g) = e(\tilde{L}_1, g^a) \).
3. \( e(\tilde{K}_{c,2}, g^a e^{\tilde{K}_1}) = e(g, g^a e(\tilde{L}_2 \tilde{L}_1^a, h)) \).
4. \( \exists z \in S_k, \text{s.t.} e(\tilde{K}_{c,2}, g) = e(U_z, \tilde{K}_{c,2}) \neq 1 \).

From requirements (1) and (3), we can obtain:

\[
D = \left[ e(L_1^a L_2, C_{1,1}) e(K_{c,1}, C_{1,2}) e(K_{c,2}, C_{1,3}) \right]^{\omega_1} = e(g, h)^{(a+c)rs},
E = e(K_2', C_0^a K_1 \cdot C_1) = e(g, g)^{as} e(g, h)^{(a+c)rs},
F = \frac{E}{D} = (g, g)^{as},
m = C/F.
\]

Similarly, we can get the following information from requirements (2) and (4):

\[
\tilde{D} = \left[ e(L_1^a \tilde{L}_2, C_{1,1}) e(\tilde{K}_{c,1}, C_{1,2}) e(\tilde{K}_{c,2}, C_{1,3}) \right]^{\omega_1} = e(g, h)^{(a+c)rs},
\tilde{E} = e(\tilde{K}_2', \tilde{C}_0^a \tilde{K}_1 \cdot C_1) = e(g, g)^{as} e(g, h)^{(a+c)rs},
\tilde{F} = \frac{\tilde{E}}{\tilde{D}} = (g, g)^{as},
m = \tilde{C}/\tilde{F}.
\]

From requirements (1)-(4), we can get the following information and express them as (6):

\[
(6) \quad F = \frac{E}{D} = (g, g)^{as} = \tilde{E}/\tilde{D} = \tilde{F}, \quad m = C/F = \tilde{C}/\tilde{F}.
\]

Nevertheless, we can know from the requirement (5):

\[
C/F \neq \tilde{C}/\tilde{F}, \quad \text{where} \quad F = E/D, \quad \tilde{F} = \tilde{E}/\tilde{D}, \quad \text{which contradicts to (6)}.
\]

Therefore, the advantage of the adversary \( A \) in winning the game of key sanity check is negligible.

VI. APPLICATION AND PERFORMANCE ANALYSIS

A. APPLICATION ANALYSIS

In order to better illustrate the idea of this paper, we take the electronic medical record (EMR) system as an example to describe. The system includes patient, TA, cloud server, and doctor. Among them, the doctor can be divided into different levels in the real scene, such as attending doctor, resident doctor, intern doctors, etc. TA assigns weights based on the importance of the attributes. The patient shares his E-record with the doctor who satisfies the access policy. At the same time, if the doctor’s private key satisfying the policy is sold, the proposed scheme can track and determine the identity information of the owner of the private key.

Fig. 1 shows the application processes of the proposed scheme in the cloud-based EMR system. Firstly, TA allocates different weights according to the importance of attributes in the system and divides the system attributes set according to the maximum weights of attributes. After generating the public parameters and the master secret key in the initialization phase, TA generates the corresponding private key for the user. The patient encrypts and uploads the E-record to the cloud server via using the access strategy simplified by the weighted attributes. Then, doctors with different attributes request and obtain E-record ciphertext of the patient. Finally, only when the doctor’s attributes meet the access policy specified by the patient can the doctor decrypt and get the E-record of the patient. In addition, for doctors who do not have decryption authority to obtain the secret key in some way and get plaintext, the key sanity check algorithm and tracking algorithm can track the identity information of the owner who leaks the decryption key.

B. COMPARISON ANALYSIS

Via comparing our constructed scheme with other related schemes in the size of public key, the size of private key, the size of ciphertext, whether supporting weighted attributes and traceability, the particular conclusions are represented in Table 1. Let \( |u| \) indicate the amount of attributes in the attributes set, \( |S| \) represents the quantity of the attributes owned by the users, and \( |l| \) represents the size of the access policy.

We introduce the concept of weighted attribute into the traceable ABE scheme, which will not affect the performance of tracking. Because the weight of attributes in the system has been determined by TA before the initialization phase, there is no cost in managing the weighted value of attributes such as user/attribute revocation, key re-generation, and policy update.

As represented in Table 1, the size of the public key, the private key of the user and ciphertext in [28] are \( 3 + |u|, |S| + 3 \) and \( 2|l| + 2 \) respectively, which are all superior to other schemes, but this scheme does not support traceability. While the scheme in [20] supports traceability, it does not consider the weighted attribute. In terms of performance, the length of public key of scheme in [20] is \( 4|u| \) which is 4 times longer than the proposed scheme, the size of private key is \( 4|S| + 1 \)

\[
\text{FIGURE 1. Traceable and weighted attribute-based encryption scheme in the cloud environment.}
\]
and the length of ciphertext is $6|l| + 1$ which are almost twice as long as that of our scheme, so the communication cost is much higher than other schemes. Both the length of the public key of our constructed scheme and the system in [25] are $7 + |u|$, but the length of the private key is $2|S| + 4$ which is shortened by $|S|$ in [25], and the size of the ciphertext is $3|l| + 3$ which is shortened by $|l|$ at the same time.

The comparisons of the costs of computation are represented in Table 2. In Table 2, let $E$ indicate the exponential action in $G$, $G_T$, $P$ is a bilinear pairing operation, $M$ represents a multiplication activity in group $G$, and $M_T$ indicates a multiplication activity in group $G_T$, as well as $|l|$ denotes the amount of attributes in the decryption key that meet the conditions in the access strategy.

Though the computation costs of key generation phase, encryption stage, decryption procedure of scheme in [28] are $(4 + |S|)E + M$, $(2 + |l|)E$, $(4|l| + 3)P + 2M_T$ respectively, which are the most efficient of other schemes, it does not hold traceability. For the sake of supporting traceability, the cost of computational of the scheme in [20] is higher than that of the scheme in [28]. In the phase of key generation, our scheme requires to calculate $(4 + 2|u|)E + M$, which is interrelated to the amount of attributes in the constructed system. It is more efficient than the efficiency of the key generation of [25] which is $(4 + 3|S|)E + M$, and our scheme slightly less efficient than the key generation algorithm of [28]. In the procedure of encryption, both our scheme and scheme [25] need to calculate $(3 + 3|l|)E$, so the efficiency of encryption of two schemes is associated with the length of access strategy. Because pairing operation takes a lot of time, the efficiency of the constructed scheme and scheme [25] is better than that of the scheme in [20]. In the phase of decryption, our constructed scheme needs to calculate $(3|l| + 1)P + 2E + 2M + M_T$, with the increase of the amount of attributes meeting the requirements of access policy in the private key of the user, the efficiency of decryption increases. The decryption efficiency of the scheme [25] is $(4|l| + 1)P + M_T + (|l| + 3)E + (|l| + 1)M$. It can be seen that its decryption efficiency is higher than our constructed system. Although the scheme [25] can achieve traceability, our scheme can achieve traceability and weighted attribute at the same time. Because the cost of the multiplication operation is much more complex than that of bilinear pairing operation, the efficiency of our scheme is lower than that in [28].

In a word, comparing with other schemes in terms of function and performance, the constructed scheme realizes the functions of weighted attribute and traceability at the cost of less performance loss. It can not only track malicious users from the key information leaked by the system, but also set the weight according to the importance of the attributes, which is closer to the actual environment.

### C. EXPERIMENT TEST

Through the above theoretical analysis, our scheme has advantages in function and efficiency. For the sake of estimating the actual characteristic of our constructed scheme more accurately, the time cost on the terminal device is further analyzed via experiments, including time of key generation procedure, encryption step, and decryption phase, as represented in Fig. 2.

We program the scheme via employing the Java pairing-based cryptography (JPBC) library and MyEclipse development environment. A Type A pairing is selected to accomplish the activities in the prime order group and construct an elliptic curve $y^2 = x^3 + x$. The experiments are carried out on the Windows 10 system, and the parameters are Inter (R) Core (TM) i5-8300H CPU at 2.30 GHz and 8GB RAM. By calling pairing function to achieve pairing() operation, module power and module multiplication are tested by calling powzn() function and mul() function respectively.

### TABLE 1. Communication cost comparison.

| Scheme       | Public key size | Private key size | Ciphertext size | weighted | traceability |
|--------------|----------------|-----------------|-----------------|----------|--------------|
| Scheme[28]   | $3^{|u|}$       | $|S|^{|u|}$      | $2|l|+2$        | ✓        | ×            |
| Scheme[20]   | $4|u|$          | $4|S|+1$         | $6|l|+1$        | ×        | ✓            |
| Scheme[25]   | $7^{|u|}$       | $3|S|^{|u|}$     | $4|l|+3$        | ×        | ✓            |
| Our scheme   | $7^{|u|}$       | $2|S|+4$         | $3|l|+3$        | ✓        | ✓            |

### TABLE 2. Computation cost comparison.

| Scheme       | Key generation | Encryption | Decryption |
|--------------|----------------|------------|------------|
| Scheme[28]   | $(4 + |S|)E + M$ | $(2 + |l|)E$ | $(4|l| + 3)P + 2M_T$ |
| Scheme[20]   | $(4 + |S|)E + (1 + |S|)M$ | $(1 + 8|l|)E + P + (1 + |l|)M_T + M$ | $3|l|P + (3|l| + 1)M_T + (2|l| + 2)E + (2|l| + 1)M$ |
| Scheme[25]   | $(4 + 3|S|)E + M$ | $(3 + 3|l|)E$ | $(4|l| + 1)P + M_T + (|l| + 3)E + (|l| + 1)M$ |
| Our scheme   | $(4 + 2|S|)E + M$ | $(3 + 3|l|)E$ | $(3|l| + 1)P + M_T + 2E + 2M$ |
We represent the tendencies of time cost of key generation procedure, encryption step and decryption phase in Fig. 2(a), (b) and (c), as the amounts of attributes raises. Effectively, the characteristics of key generation procedure, encryption step, and decryption phase are decided by the amount of attributes \(|S|\) of the user, the number of attributes in access policy \(|I|\) and the amount of attributes in the decryption key that contains the access policy \(|I|\). We are able to describe the abscissa axis labels in Fig. 2 as “Number of attributes” because of \(|S|, |I|, \text{and} |I|\) primarily mean the amount of attributes.

In Fig. 2(a), we are able to discover that the time cost of key generation procedure increases linearly with the amount of attributes of the user. The experimental consequences of encryption and decryption represented in Fig. 2(b) and (c) are resembled to that of key generation. When the amount of user attributes in the experiment is 50, it will cost 1024ms in order to produce the key for the user. If the data owner sets the amounts of attributes in access policy at 50, the time of encryption will cost 1528ms. While the amount of attributes in the decryption key of the user that contains the access policy is 50, the user will spend 1066ms on decrypting the ciphertext. We are able to discover the time cost of encryption is maximum, whereas the time cost of key generation procedure and decryption phase are temperate. Actually, the encryption operation is not often performed. The key generation action is carried out by a trusted center, thus the overhead of decryption on the user side is the primary criterion measuring the efficiency of the constructed scheme. However, the amounts of attributes in the decryption key of the user that contains the access strategy is far less than the total number of the attributes, so it can’t reach 50 in the real application.

Hence, the time cost of our proposed system is acceptable. All in all, the experimental outcomes represented in Fig. 2 are consistent with the theoretical analysis represented in Table 2.

Based on the above analysis, according to the function and performance, the constructed scheme can not only track malicious user from the key information leaked by the system, but also set the weight of attributes according to the importance degree, which is more suitable to the actual environment, and the performance has been greatly optimized.

VII. CONCLUSION

In this paper, a traceable attribute-based encryption scheme with weight is constructed. The constructed scheme can achieve high expression, efficiency, and security. In particular, in our constructed scheme, it can detect the original holder of the key from the private key of the user, as well as prevent malevolent key holder from leaking his private key for whatever intention without getting caught. In addition, the attributes are marked weight with the importance, and the access policy can be designed according to the weighted attributes. Under the assumption of q-BDHE in the standard model, the constructed scheme is proved to be secure against chosen-plaintext attack. Due to the strong security of lattice theory, our future study will target at building up a traceable CP-ABE on the basis of lattice theory.

REFERENCES

[1] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Proc. Annu. Int’l Conf. Theory Appl. Cryptograph. TECH. Berlin, Germany: Springer, 2005, pp. 457–473.
[2] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in Proc. IEEE Symp. Secur. Privacy, May 2007, pp. 321–334.
[3] 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. (CCS), 2006, pp. 89–98.
[4] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, “Flexible and fine-grained attribute-based data storage in cloud computing,” IEEE Trans. Services Comput., vol. 10, no. 5, pp. 785–796, Sep. 2017.
[5] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, “User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage,” IEEE Syst. J., vol. 12, no. 2, pp. 1767–1777, Jun. 2018.
[6] J. Li, N. Chen, and Y. Zhang, “Extended file hierarchy access control scheme with attribute based encryption in cloud computing,” IEEE Trans. Emerg. Topics Comput., to be published, doi: 10.1109/TETC.2019.2904637.
[7] J. Li, Q. Yu, and Y. Zhang, “Hierarchical attribute based encryption with continuous leakage-resilience,” Inf. Sci., vol. 484, pp. 113–134, May 2019.
[8] J. Li, Q. Yu, Y. Zhang, and J. Shen, “Key-policy attribute-based encryption against continual auxiliary input leakage,” Inf. Sci., vol. 470, pp. 175–188, Jan. 2019.
[9] J. Li, Y. Wang, Y. Zhang, and J. Han, “Full verifiability for outsourced decryption in attribute based encryption,” IEEE Trans. Services Comput., to be published, doi: 10.1109/TSC.2017.2710190.
[10] M. Abdalla, A. W. Dent, J. Malone-Lee, G. Neven, D. H. Phan, and N. P. Smart, “Identity-based traitor tracing,” in Proc. Int. Workshop Public Key Cryptogr. Berlin, Germany: Springer, 2007, pp. 361–376.

[11] X. Liu, J. Ma, J. Xiong, Q. Li, and T. Zhang, “Ciphertext policy weighted attribute based encryption scheme,” J. Xi’an Jiaotong Univ., vol. 47, no. 8, pp. 44–48, 2013.

[12] X. Liu, J. Ma, J. Xiong, Q. Li, and J. Ma, “Ciphertext-policy weighted attribute based encryption for fine-grained access control,” in Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst. Sep. 2013, pp. 51–57.

[13] Y. T. Wang, K. F. Chen, and J. H. Chen, “Attribute-based traitor tracing,” J. Inf. Sci. Eng., vol. 27, no. 1, pp. 181–195, 2011.

[14] Z. Liu, Z. Cao, and D. S. Wong, “White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures,” IEEE Trans. Inf. Forensics Security, vol. 8, no. 1, pp. 76–88, Jan. 2013.

[15] Z. Liu, Z. Cao, and D. S. Wong, “Traceable CP-ABE: How to trace decryption devices found in the wild,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 1, pp. 55–68, Jan. 2015.

[16] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, “White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 6, pp. 1274–1288, Jun. 2015.

[17] J. Ning, Z. Cao, X. Dong, L. Wei, and X. Lin, “Large universe ciphertext-policy attribute-based encryption with white-box traceability,” in Proc. Eur. Symp. Res. Comput. Secur. Cham, Switzerland: Springer, 2014, pp. 55–72.

[18] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, “Multi-authority ciphertext-policy attribute-based encryption with accountability,” in Proc. 6th ACM Symp. Inf. Comput. Commun. Secur. (ASIACCS), 2011, pp. 386–390.

[19] J. Zhou, Z. Cao, X. Dong, and X. Lin, “TR-MABE: White-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2015, pp. 2398–2406.

[20] K. Zhang, H. Li, J. Ma, and X. Liu, “Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability,” Sci. China Inf. Sci., vol. 61, no. 3, Jun. 2017, Art. no. 032102.

[21] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, “Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing,” Future Gener. Comput. Syst., vol. 78, pp. 720–729, Jan. 2018.

[22] J. Ning, X. Dong, Z. Cao, and L. Wei, “Accountable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud,” in Proc. Eur. Symp. Res. Comput. Secur. Cham, Switzerland: Springer, 2015, pp. 270–289.

[23] Z. Guan, J. Li, Y. Zhang, R. Xu, Z. Wang, and T. Yang, “An efficient traceable access control scheme with reliable key delegation in mobile cloud computing,” EURASIP J. Wireless Commun. Netw., vol. 2016, no. 1, p. 208, Sep. 2016.

[24] Q. Li, H. Zhu, Z. Ying, and T. Zhang, “Traceable ciphertext-policy attribute-based encryption with verifiable outsourced decryption in eHealth cloud,” Wireless Commun. Mobile Comput., vol. 2018, pp. 1–12, Jun. 2018.

[25] S. Wang, K. Guo, and Y. Zhang, “Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage,” PLoS ONE, vol. 13, no. 9, Sep. 2018, Art. no. e0203225.

[26] J. Ning, Z. Cao, X. Dong, and L. Wei, “Traceable and revocable CP-ABE with shorter ciphertexts,” Sci. China Inf. Sci., vol. 59, no. 11, Sep. 2016, Art. no. 119102.

[27] Z. Liu, S. Duan, P. Zhou, and B. Wang, “Traceable-then-revocable ciphertext-policy attribute-based encryption scheme,” Future Gener. Comput. Syst., vol. 93, pp. 903–913, Apr. 2019.

[28] Z. Chu and Z. Wang, “Improved weighted attribute-based encryption scheme with revocation,” Comput. Eng. Appl., vol. 13, no. 3, pp. 82–86, 2018.

XIXI YAN received the B.S. degree from the Huazhong University of Science and Technology, in 2006, the M.S. degree from Henan Polytechnic University, in 2009, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2012. She is currently an Associate Professor with the School of Computer Science and Technology, Henan Polytechnic University. She is mainly involved in digital content security and cloud computing.

QICHAO ZHANG received the B.S. degree from Dalian Polytechnic University, in 2015. He is currently pursuing the M.S. degree with Henan Polytechnic University. His research interests include modern cryptography, networks, and information security.

YONGLI TANG received the M.S. degree from Henan Polytechnic University, in 2005, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2008. He visited Dublin City University, Ireland, in 2013. He is currently a Professor with the School of Computer Science and Technology, Henan Polytechnic University. His research interests include modern cryptography, networks, and information security.