Abstract—Attribute-based encryption received widespread attention as soon as it was proposed. However, due to its specific characteristics, some restrictions on attribute set in attribute-based encryption are not flexible enough in actual operation. In addition, since access authorities are determined according to users’ attributes, users sharing the same attributes are difficult to be distinguished. Once a malicious user makes illicit gains by their decryption authorities, it is difficult to track down specific users. This paper follows practical demands to propose a more flexible key-policy attribute-based encryption scheme with black-box traceability. The scheme has a constant size of public parameters which can be utilized to construct attribute-related parameters flexibly, and the method of traitor tracing in broadcast encryption is introduced to achieve effective malicious user tracing. In addition, the security and feasibility can be proved by the security proofs and performance evaluation in this paper.

Index Terms—Attribute-based encryption, Key-policy, Traceability, Unbounded

I. INTRODUCTION

The rapid development of the network and communication industry has made communication system architectures more and more diverse. Complex business requirements require a more flexible access control of data. Therefore, the model of the authority management system is no longer confined to the traditional identity-based user management. At the same time, security issues are still one of the most important issues on any updated node. We have always used encryption to ensure data security, and then control user access authorities through key management. Nowadays, traditional encryption systems can hardly meet the current demand for flexible management of authorities. Therefore, attribute-based encryption (ABE) [1] came into being.

Attribute-based encryption makes the user’s access authority or the access threshold of ciphertexts no longer bound to the individual user, but is associated with a set of attributes. Therefore, attribute-based encryption technology can better meet the needs of fine-grained access control. Currently, attribute-based encryption systems are mainly divided into two categories by different settings of access structures: key-policy attribute-based encryption (KP-ABE) and cipher-policy attribute-based encryption (CP-ABE). Systems with key-policy construct access structures corresponding to the attributes owned by users and embeds them into users private keys, whereas the systems with cipher-policy bind such access structures to ciphers. Moreover, researches based on different needs have been proposed solutions one by one in both types.

However, ABE has brought new problems while meeting new demands. First of all, ABE are designed to better adapt to some changes, but there are some inherent limitations in the current structure. The size of the public parameters of most current systems increases linearly with the maximum size of global attribute set, which makes them limited in practical applications. For this problem, a concept called unbounded ABE is proposed which means that the public parameter size is not bound to the global attribute base size. Otherwise, since the relevant authorities are described by sets of attributes, when a malicious user intentionally leaks or sells the key to other unauthorized users in exchange for specified benefits, it will be hard to catch the traitor effectively. For protecting data privacy and interests of users, the traitor tracing mechanism has become indispensable.

A. Related Work

Sahai and Waters first proposed ABE in [2], which solved the problem of fine-grained access control. Since then, Goyal et al. [3] proposed the first KP-ABE, as well as Bethencourt et al. proposed the first CP-ABE in [4], and both of them support any monotonic access tree. At present, there is a series of work on both KP-ABE and CP-ABE [5]–[10] according to different need to obtain better performance and achieve a higher security level.

For some inherent limitations of ABE system design, Lewko and Waters first proposed the concept of Unbounded ABE in [11] and gave their solutions. Since then, Tatsuaki and Katsuyuki have proposed the first unbounded inner-product encryption (IPE) scheme in [12]. In their scheme, public parameters do not impose additional restrictions on the predicates and properties used to encrypt the decryption key. Also, there are many pieces of research [13]–[15] that have been explored in depth. The most recent work from this perspective comes from [5]. This scheme is not only unbounded but also implements selective security, relying on simple difficulties.

While ABE blurs the correspondence between the user’s decryption authority and the user, it also brings some tricky security issues. Because of users’ authorities in the ABE system are determined by the attributes they owned, it will be hard to trace malicious users. To solve this problem, Liu
et al. first proposed their scheme in [8] of implementing white-box tracing to implement malicious user tracing in ABE systems, and introduced the concepts of black-box tracing and white-box tracing. After that, Liu et al. continue to put forward a black-box tracing scheme in [9] to solve the same problem, which more in line with the actual scene. In addition, Ning et al. have further proposed more competitive white-box tracing schemes in [16]–[19]. There are also a number of researches proposed like [10], [20]–[23] aiming at various needs. [24], [25] are recent results of further research on black-box tracing functionality.

B. Motivation and Contribution

There have been many studies that have proposed some solutions to implement the tracing function in the attribute-based encryption system. As we can see, most of the existing schemes with traceability implement related functions in the form of white-box tracing. However, it is clear that the white-box tracing scenario is not very consistent with the actual malicious user tracing requirements. Moreover, according to the existing black-box tracing schemes, there are limitations to a certain extent. According to the actual need, we put forward the scheme with black-box traceability. Our main contributions are as follows:

- **Dynamic attribute addition (Unbounded).** Our scheme is an unbounded system that can associate attributes with a constant number of public parameters.
- **Efficient black-box traceability.** Our scheme can effectively trace the source of the decryption black-box without obtaining any details related to the private key in sublinear time.

Furthermore, we have given the security proof on the hardness assumptions in V. And, from the comparison of efficiency, our solution is also quite competitive in terms of the actual time cost. As follows, we show the comparisons between our scheme and several related work in terms of functionality and efficiency. From the perspective of functionality, we compared black-box traceability, and dynamic attribute addition for five schemes in Table I. For three of these schemes with black-box traceability and similar structure, we compared their efficiency by analyzing their data sizes in Table II.

C. Organization

The remainder of this paper is organized as following. Section 2 presents some preliminaries in cryptographic and security assumptions. Section 3 fully describes the statement of the scheme proposed in this paper, including the system model, the conception of tracing with black-box, and design goals. Section 4 presents the proposed scheme in detail. Section 5 and Section 6 performs the security and performance analyses, respectively.

II. PRELIMINARIES

A. Unbounded Key-Policy Attribute-Based Encryption (KP-ABE)

According to different settings of access structure used in attribute pair authentication, ABE has divided into cipher-policy attribute-based encryption and key-policy attribute-base encryption. A key-policy attribute-based encryption scheme could be described by a tuple of four algorithms (Setup, KeyGen, Encrypt, Decrypt):

- **Setup(λ, S) → (pp, MSK):** The system establishment algorithm includes two input parameters, namely λ, the system security parameter, and S, the global attribute set. After running the system establishment algorithm, the public parameter pp and the system master key MSK will be output.
- **KeyGen(pp, MSK, Λ) → SKΛ:** The function of the key generation algorithm is to generate private keys for users. It takes the system public parameter pp, the system master key MSK and an access policy Λ corresponding to the attributes owned by the user as input, and then outputs the private key SKΛ.
- **Encrypt(pp, x, M) → CTx:** Encryption algorithm is used to encrypt plaintext messages. It takes the system public parameter pp, an attribute set x, and the plaintext message M as input, and outputs the encrypted ciphertext CTx. Note that the attribute set x is publicly given in ciphertext CTx.
- **Decrypt(pp, CTx, SKΛ) → M |⊥:** The decryption algorithm takes the system public parameters pp, a ciphertext CTx and a private key SKΛ as input. If the attribute set in the ciphertext satisfies the access policy in the private key, it would output the corresponding plaintext, otherwise, ⊥.

**Correctness.** It requires that for all ( pp, MSK ) ← Setup ( λ, S ), all SKΛ ← KeyGen ( pp, MSK, Λ) and all CTx ← Encrypt ( pp, x, M ),

\[ Pr[\text{Decrypt}(pp, CTx, SKΛ) = M] = 1 \]

holds, when the x in CTx satisfies the access structure Λ in SKΛ.

**Unbounded** [5]. An ABE scheme is unbounded if the running time of Setup only depends on λ, otherwise, is bounded.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Black-box Traceability</th>
<th>Unbounded</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5]</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>[9]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>[24]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[25]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Ours</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference</th>
<th>pp</th>
<th>SK</th>
<th>CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>[9]</td>
<td>O(</td>
<td>S</td>
<td>+ √n)</td>
</tr>
<tr>
<td>[25]</td>
<td>O(</td>
<td>S</td>
<td>+ n)</td>
</tr>
<tr>
<td>Ours</td>
<td>O(√n)</td>
<td>O(</td>
<td>x</td>
</tr>
</tbody>
</table>

| S | be the size of the attribute universe, n the number of users in system, x the size of attribute set of a ciphertext.
Bilinear Group of Composite Order

Bilinear group of composite order is firstly proposed in [26] and widely used in a variety of cryptographic systems. The specific definition is as follows.

Let $G$ be a group generation algorithm with security parameter $\lambda$ as input and a type of $(p, p_1, p_2, p_3, G, H, G_T, e)$ as output in which $p, p_1, p_2, p_3$ are four different prime numbers determined by security parameter, $G, H, G_T$ are three cyclic groups of order $N = p p_2 p_3$ and $e : G \times H \to G_T$ is a mapping that satisfies the following conditions:

- **Bilinear:** $\forall g \in G, h \in H$, and $a, b \in \mathbb{Z}_N$, $e(g^a, h^b) = e(g, h)^{ab}$.
- **Non-degenerate:** $\exists g \in G, h \in H$, $e(g, h)$ is an $N$-order element of group $G_T$.

We require that the group operations in $G, H, G_T$ as well the bilinear map $e$ are computable in deterministic polynomial-time respect to $\lambda$.

Let $G_{p_1}, G_{p_2}, G_{p_3}$ be subgroups of order $p, p_1, p_2, p_3$ in $G$, and Let $H_{p_1}, H_{p_2}, H_{p_3}$ be subgroups of order $p, p_1, p_2, p_3$ in $H$ respectively. It is easy to know that these four subgroups are “orthogonal” to each other ($\forall g_i \in G, h_i \in H, i \neq j$, $e(g_i, h_j) = 1$). Further, for any element $T \in G$, $T$ can be uniquely expressed as the product of an element in $G_{p_1}$, an element in $G_{p_2}$, and an element in $G_{p_3}$. The above also applies to group $H$.

**Computational Assumptions.** The scheme proposed in this paper will be based on four assumptions in the composite-order group, used e.g. in [5], [27].

**Subgroup Decision Assumption.** For a generator $G$, we define the following distribution:

$$I := (N = p p_2 p_3, G, H, G_T, e) \leftarrow_R G(\lambda), g_1 \leftarrow_R G_{p_1}, g_2 \leftarrow_R G_{p_2}, g_3 \leftarrow_R G_{p_3}, h_1 \leftarrow_R H_{p_1}, h_3 \leftarrow_R H_{p_3}, h_{12} \leftarrow_R H_{p_1 p_2}, D = (g_1, g_2, g_3, h_1, h_3, h_{12}), T_1 \leftarrow R G_{p_1}, T_2 \leftarrow R G_{p_2, p_3}.$$

Then we define the advantage of an algorithm $A$ in breaking $(p_1 \to p_2)$ - subgroup decision assumption to be:

$$\text{Adv}^{A}(p_1 \to p_2) (\lambda) = |Pr[|A(I, D, T_1) = 1] - Pr[|A(I, D, T_2) = 1]|.$$ 

$(G_{p_1 \to p_2})$ - subgroup decision assumption. We say that $(p_1 \to p_2)$ - subgroup decision assumption holds for generator $G$ if for all polynomial-time algorithms $A$, $\text{Adv}^{A}(p_1 \to p_2) (\lambda)$ is a negligible function of $\lambda$.

By changing the roles of $G$ and $H$ and/or permuting the indices for subgroups, one can define $(G_{p_1 \to p_3})$ - subgroup decision assumption, $(G_{p_2 \to p_3})$ - subgroup decision assumption, $(H_{p_1 \to p_3})$ - subgroup decision assumption, and $(H_{p_1 \to p_2})$ - subgroup decision assumption.

**Subgroup Decision Diffie-Hellman Assumption.** For a generator $G$, we define the following distribution:

$$I := (N = p p_1 p_3, G, H, G_T, e) \leftarrow_R G, g_1 \leftarrow_R G_{p_1}, g_2 \leftarrow_R G_{p_2}, g_3 \leftarrow_R G_{p_3}, h_1 \leftarrow_R H_{p_1}, h_2 \leftarrow_R H_{p_2}, h_3 \leftarrow_R H_{p_3},$$

$$D = (g_1, g_2, g_3, h_1, h_2, h_3), T_1 = (h_1^2, h_1^3, h_2^{g^y}), T_2 = (h_1^2, h_1^3, h_1^{g+2}).$$

Then we define the advantage of an algorithm $A$ in breaking $p_1 \to p_2$ subgroup Diffie-Hellman assumption to be:

$$\text{Adv}^{A}_p (\lambda) = |Pr[|A(I, D, T_1) = 1] - Pr[|A(I, D, T_2) = 1]|.$$ 

By changing the roles of $G$ and $H$ and/or permuting the indices for subgroups, one can define $p_2 \to p_3$ subgroup Diffie-Hellman assumption and $p_3 \to p_1$ subgroup Diffie-Hellman assumption.

**Decisional Linear Assumption.** This is a simple extension of the Decisional Diffie-Hellman (DDH) Assumption. For a generator $G$, we define the following distribution:

$$I := (p, G, G_T, e : G \times G \to G_T) \leftarrow_R G, g \leftarrow_R G, a, b, c, x, y \leftarrow_R \mathbb{Z}_p, D = (g, g^a, g^b, g^c, g^{ax}, g^{by}), T_1 = g^{x+y}, T_2 = g^{x+y}.$$

Then we define the advantage of an algorithm $A$ in breaking decisional linear assumption to be:

$$\text{Adv}^{A}_p (\lambda) = |Pr[|A(I, D, T_1) = 1] - Pr[|A(I, D, T_2) = 1]|.$$ 

**External Diffie-Hellman Assumption.** For an asymmetric bilinear mapping $e : G \times H \to G_T$, the External Diffie-Hellman (XDH) assumption states that the Decisional Diffie-Hellman (DDH) assumption is hard in the group $H$ (Not necessarily hard in $G$) which has been proved in [28].

C. Access Control

According to the definition of the access structure in [29], in ABE, the attributes corresponds to the role of the participant, that is, the access structure $A$ contains the set of authorized attributes. With a collection of all attributes in the system denoted by $\{P_1, ..., P_n\}$, we define $A$ including all the access structures for the attribute set, which has

$$2^{\{P_1, P_2, ..., P_n\}} = \{A|A \subseteq \{P_1, P_2, ..., P_n\}\}.$$ 

If a collection $L \subseteq 2^{\{P_1, P_2, ..., P_n\}}$ has

$$\forall R, Q \subseteq \{P_1, P_2, ..., P_n\}, R \in L \land R \subseteq Q \rightarrow Q \in L,$$

we say $L$ is monotone. For the collection $L \subseteq 2^{\{P_1, P_2, ..., P_n\}} \setminus \{\emptyset\}$, we describe the sets in it as authorized set, and the unauthorized set identifies those not in $L$.

**Monotone Span Programs [30].** A (monotone) span program for attribute universe $[n]$ is a pair $(A, \rho)$ where $A$ is a $i \times i$ matrix over $\mathbb{Z}_n$ and $\rho : [n] \rightarrow [n]$. Given $x = (x_1, ..., x_n) \in \{0, 1\}^n$, we say that

$$x \text{ satisfies } (A, \rho) \text{ iff } 1 \in \text{span}(A_x),$$

Here, $1 := (1, 0, ..., 0)^T \in \mathbb{Z}^{1 \times i}$ is a row vector; $A_x$ denotes the collection of vectors $\{A_j : x_{\rho(j)} = 1\}$ where $A_j$ denotes
the \( j \)'th row of \( A \); and span refers to linear span of collection of (row) vector over \( \mathbb{Z}_p \).

\[
\sum_{j: \rho(j)=1} \omega_j A_j = 1,
\]

(1)

Observe that the constants \( \omega_j \) can be computed in polynomial-time in the size of the matrix \( A \) via Gaussian elimination. Like in [5], we need to impose a one-use restriction, that is, \( \varrho \) is a permutation and \( \iota = n \). By re-ordering the rows of \( A \), we may assume WLOG that \( \varrho \) is the identity map, which we omit in the rest of this section.

(statistical lemma [5]) For any \( x \) that does not satisfy \( A \), the distributions

\[
\{ \{ v_j \}_{j:x_j=1}, \{ A_j \left( \frac{\alpha}{\bar{u}} \right) + r_j v_j, r_j \}_{j: \bar{u} \in \{u\}} \}
\]

perfectly hide \( \alpha \), where the randomness is taken over \( v_j \sim R Z_p, \bar{u} \leftarrow R Z_p^{\iota-1} \), and for any fixed \( r_j \neq 0 \).

III. Problem Statement

A. System Model

![System Model Diagram]

We use a specific example to describe our system architecture. As showing in Fig.1, there are three types of entities in our system:

- **Cloud server**: The cloud server provides users with seemingly unlimited data storage function and data sharing service. In our system model, the cloud server is honest, that is, it does not tamper with the users’ data. But at the same time, it is curious about the data and the attributes of the users. In other words, the cloud server is a semi-trusted entity in our system.

- **Administrator**: Generating system parameters, distributing user private keys, and tracing malicious users are all functions that the administrator is responsible for. In our system, the administrator is considered a trusted party.

- **User**: In our system, users of the system use their private keys to obtain and decrypt data from the cloud server. There may be malicious users who gain benefits by selling their decryption rights which violates regulations.

The users encrypt their data through the public parameters generated by the system administrator to ensure data confidentiality, and then upload the corresponding ciphertexts to the cloud server to share with other people. Without the system private key, an attacker (including the semi-trusted cloud server) will not be able to obtain anything about the data. The uploaded encrypted data does not contain any information related to the users who send them to the cloud, so they are completely anonymous. In addition, when a malicious key leak occurs, we will obtain the source of the compromised key through a tracing algorithm.

B. Malicious User Tracing with Black-Box

In I, we have mentioned that ABE, due to its inherent characteristics, has some unavoidable disadvantages while implementing fine-grained access function. Unlike identity-based encryption, in an ABE system, users’ authorities are made up of the attributes they own. Once a key leak occurs, it is difficult to accurately trace the malicious user associated with it in the ABE system. To solve this problem, Liu et al. proposed an entity named black-box in [9] to simulate the corresponding scene.

In this article, we use a similar concept to describe the corresponding security requirements scenario: We assume that the compromised key is manufactured into a “Black-Box” with decryption authority by the malicious user in exchange for benefits. In return, a malicious user would sell a “black-box” indicating its value (that is, its maximum decryption rights) without providing any specific information about the key it contained. For a malicious user tracer (or surveillance agency), by interacting with this publicly sold decryption box, in the event that he cannot obtain any details of the decryption key it owns, he can trace back to the source of the “black-box” keys.

C. Security Model

We define the security of the scheme proposed in IV in the following games.

The first game is called a message-hiding game. We can find that this game is exactly the same as the standard key policy attribute-based encryption except that the indexes of private keys is specified during the key query phase. This is a standard semantic security game that includes a challenger and an adversary. At the beginning of the game, both the challenger and the adversary \( A \) get \( K \) and \( \lambda \) as inputs:

**Setup.** The challenger runs \( Setup(\lambda) \) and gives the public parameter \( pp \) to \( A \).

**Phase 1.** For \( k = 1 \) to \( q \), \( A \) adaptively submits \( K_k = (\varrho, A) \), and the challenger responds with \( SK_{k,A} \).

**Challenge.** \( A \) submits two equal-length messages \( M_0, M_1 \) and an attribute set \( \lambda^* \). The challenger flips a random coin \( b \in \{0, 1\} \), and sends \( CT_{k^*} \leftarrow Encrypt(pp, M_b, \lambda^*, 1) \) to \( A \).

**Phase 2.** For \( k = q+1 \) to \( K' (K' \leq \lambda) \), \( A \) adaptively submits \( K_k = (\varrho, A) \), and the challenger responds with \( SK_{k,A} \).

**Guess.** \( A \) outputs a guess \( b' \in \{0, 1\} \) for \( b \).

**Game\textsubscript{MH}**: In the Challenge phase the challenger sends

\( CT \leftarrow Encrypt(pp, M_b, \lambda^*) \) to \( A \). \( A \) wins the game if \( b' = b \) under the restriction that \( \lambda^* \) cannot be satisfied by any of the queried combinations of attributes \( \lambda_1, ..., \lambda_{K'} \). The advantage
of $A$ is defined as $\text{Adv}_{\text{MH}} = |Pr[b' = b] - \frac{1}{2}|$. A scheme is message-hiding if for all polynomial-time adversaries $A$ the advantage $\text{Adv}_{\text{MH}}$ are negligible in $\lambda$.

**Theorem 1.** If the subgroup decision assumptions and the subgroup Diffie–Hellman assumptions hold, then no polynomial-time adversary will win the game $\text{Game}_{\text{MH}}$ with non-negligible advantage.

We describe tracing capability through the next security game called $\text{Game}_{\text{IH}}$. It is worth noting that the ciphertext used to implement the tracing mechanism is different from ordinary ciphertexts. In order to achieve effective malicious user tracing, then it must be guaranteed:

1. When the adversary knows all the private keys except the private key whose matrix position is $(i, j)$, it still cannot distinguish $\text{Encrypt}(pp, M, x, k)$ and $\text{Encrypt}(pp, M, x, k + 1)$.
2. Even if the adversary does not satisfy the access structure $A$, it should not be able to determine whether the index $k$ or $k + 1$ for encryption.

The game takes the index $k$ as input which is provided as input to both the challenger and the adversary.

**Setup.** Challenger runs the setup algorithm and gives the public parameter $pp$ to adversary $A$.

**Phase 1.** For $k = 1$ to $q$, $A$ adaptively submits an access policy $k_i = (\rho, A)$ to challenger to get $SK_{k, A}$.

**Challenge.** $A$ submits a message $M$ and a non-empty attribute set $x^*$. Challenger runs a random algorithm to get a bit $b \in \{0, 1\}$ and sends $\text{Encrypt}(pp, M, x^*, k + b)$ to $A$.

**Phase 2.** For $k = q + 1$ to $k'$ ($k' \leq K$), $A$ adaptively submits an access policy $k_i = (\rho, A)$ to challenger to get $SK_{k, A}$.

**Guess.** $A$ outputs a guess $b' \in \{0, 1\}$ as his guess.

$\text{Game}_{\text{IH}}$: $A$ wins the game if $b' = b$ under the restriction that none of the pairs $(k, A_k)$ satisfies $(k = k') \land (x^* \text{ satisfies } A_k)$. The advantage of $A$ is defined as $\text{Adv}_{\text{IH}} = |Pr[b' = b] - \frac{1}{2}|$. A scheme is index-hiding if for all polynomial-time adversaries $A$ the advantage $\text{Adv}_{\text{IH}}$ are negligible in $\lambda$.

**Theorem 2.** If the XDH assumption and the decisional linear assumption hold, then no polynomial-time adversary can win the game $\text{Game}_{\text{IH}}$ with a non-negligible advantage.

**Theorem 3.** If our system is a message-hiding and index-hiding scheme, then it is secure and traceable.

**IV. THE PROPOSED SCHEME**

**Technical Overview.** Our scheme is built on an asymmetric compound order bilinear group $(G, H, GT)$, whose order $N$ is the product of four prime numbers $p, p_1, p_2, p_3$. And the main challenge in building an unbounded system is associating attributes that can be added dynamically with a constant number of public parameters. We would replace the exponent associated with attribute in bounded systems with $s_k(\omega_0 + k\omega_1)$, where $s_k(k \in \{i\})$ are randomness used in encryption. Next, we need to bind the $s_k(\omega_0 + k\omega_1)$ together via some common randomness $s$. It suffices to use $s\omega + s_k(\omega_0 + k\omega_1)$ in the ciphertext.

B. **User Register**

During the user registration phase, users perform a round of interaction with the system administrator to obtain their private keys. A user applies for registration by sending an access policy, and receives the private key corresponding to the position information, the system administrator assigns the position in the user matrix and generates the private key through the decryption device only by constructing some tracing ciphertext without any details of the private keys.

**Nations.** We use $K$ to represent the total number of users in the system. Each user corresponds to the position in the matrix $M \times m$. The user assigned an index $k = (i - 1) \times m + y$ corresponding to the matrix position $(i, j)$. Let $n$ be a positive integer, then $[n]$ represents the set of integers $\{1, 2, \cdots, n\}$. And, for $g^x = (g^{x_1}, g^{x_2}, \cdots, g^{x_n})$ and $g^{y'} = (g^{y'_1}, g^{y'_2}, \cdots, g^{y'_n})$, there is $g^{y'} g^x = (g^{y'_1 + x_1}, g^{y'_2 + x_2}, \cdots, g^{y'_n + x_n})$. Similarly, $e$ is a bilinear mapping, and $e_n(g^x, g^{y'}) = \Pi_{i \in [n]} e(g^{x_i}, g^{y'_i})$.

**A. Initialization**

The initialization phase is performed by a trusted third party. The main work at this stage is parameter initialization, which corresponds to the Setup algorithm of the standard KP-ABE scheme:

**Setup($\lambda, m$) $\rightarrow$ ($pp, MSK$).** The system setup algorithm takes the system security parameter $\lambda$ and the matrix size $m$ as input. Firstly run the group generation algorithm to get $G(\lambda) \rightarrow (N = p_1 p_2 p_3, G, H, GT, e)$. Then, the algorithm randomly choose exponents $\omega_0, \omega_1 \in \mathbb{Z}_p$, exponents $(c_i, r_i, \omega_i \in \mathbb{Z}_N, i \in [m])$, and randomly choose generators $h, h_1, h_2, g_1$ of cyclic groups $H_{p_1 p_2 p_3}$, $H_2, H_p, G_{p_2}$. It sets public parameters as:

$$pp = ((N, G, H, GT, e), g_1, g_1^{\omega_0}, g_1^{\omega_1}, h_p, E = e(h, g_1)^\omega, \{E_i = e(h_p, g_1)^{s_i}, G_i = g_i^{r_i}, Z_i = g_i^{\omega_i}, i \in [m]\}, \{D_j = h_j^{s_i}, i \in [m]\})$$

The master secret key is set as

$$MSK = (h, h_1, \alpha, \alpha_1, \cdots, \alpha_m, r_1, \cdots, r_m, c_1, \cdots, c_m, \omega, \omega_0, \omega_1)$$
system administrator can correspond to the key generation algorithm in the standard KP-ABE scheme and is described as follows:

**KeyGen**(pp, MSK, $\mathcal{A} = (A, \rho)$) $\rightarrow$ SK$_{(i,j),\mathcal{A}}$. $\mathcal{A}$ is a monotone span program submitted by the user where $A \in \mathbb{Z}_N^{\times n}$ is a matrix. $\rho$ is a mapping which maps each row of $A$ to an attribute. Then, it randomly chooses exponents $\eta_{i,j}, \zeta_1, \ldots, \zeta_t \in \mathbb{Z}_N, \bar{u} \in \mathbb{Z}_N^{-1}$ and computes:

$$K = (K_0 = h_p^{\tau_{\bar{u}} + \alpha}, K_0' = (h_p^{\bar{u}})^{\eta_{i,j}}, K_1 = h_p^{\eta_{i}}, \{(K_{2,k} = h_{k(\omega_0 + k\omega_1)}, K_{3,k} = h_{\bar{k}}^{\zeta_k}, K_{4,k} = h_{h_1}^{\zeta_k})\}_{k \in [t]}$$

Finally, it outputs $SK_{(i,j),\mathcal{A}} = ((i, j), \mathcal{A}, K)$ and sends to the user.

Once the user obtains his due private key, the user registration phase is complete.

**C. File Generation**

Since our cloud server is a semi-trusted party with honest but curious features, users need to encrypt the data before uploading it to the cloud. When a user encrypts the data that he owns, he can specify the set of attributes that the file needs to meet and the range of users that can access the file. And then, he uses the public parameters pp of the system to complete the encryption. The operation of the user to generate an encrypted file for uploading may correspond to the encryption algorithm in the standard KP-ABE scheme. The user’s operation can be described as the encryption algorithm below.

**Encrypt**(pp, M, x, ($\bar{i}, \bar{j}$)) $\rightarrow$ CT$_x$. For a vector of attributes represented by $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, the algorithm randomly chooses $s, \{s_k\}_{k \in [t]} \in \mathbb{Z}_N$ and computes:

$$P = (P_0 = g_1, \{P_{1,\rho(x)} = g_1^{h_p^{\rho(x)}}, P_{2,\rho(x)} = g_1^{h_p^{\bar{u}(\omega_0 + k\omega_1)}}\}_{k \in [t]})$$

And then, it randomly chooses exponents $\kappa, \tau, \gamma_1, \ldots, \gamma_{m+1}, \ldots, \gamma_m \in \mathbb{Z}_N, v_1, v_2, d_1, \ldots, d_m \in \mathbb{Z}_N$ and $v_2 \in \mathbb{Z}_N^2$, which makes $v_1 \cdot v_2 = 0$ true. Let $v'_c := v_c + v_N \cdot v_2$ where $v_N \in \mathbb{Z}_N$, then $v'_c \cdot v_1 = v_c \cdot v_1$.

For each column $j \in [m]$:

- $j < \bar{j}$: It sets: $C_j = D_j^{v'_e} \cdot h_p^{d_j}$, $C_j' = h_p^{d_j}$.

- $j \geq \bar{j}$: It sets: $C_j = D_j^{v'_e} \cdot h_p^{d_j}$, $C_j' = h_p^{d_j}$.

For each row $i \in [m]$:

- $i < \bar{i}$: It randomly chooses $\gamma_i \in \mathbb{Z}_p, v_i \in \mathbb{Z}_N^2$ and sets:

$$Q_i = g_1^{\gamma_i}, Q'_i = Q_i \cdot g_1^{v_i}, Q''_i = g_1^{v_i}, T_i = M \cdot E_i^{\gamma_i}.$$

- $i \geq \bar{i}$: It randomly chooses $\gamma_i \in \mathbb{Z}_p, v_i \in \mathbb{Z}_N^2$ which makes $v_i \cdot v'_c \neq v_i \cdot v_c$ true and sets:

$$R_i = G_i^{\gamma_i}, R'_i = G_i^{\gamma_i}.$$  

And then, it could compute

$$D_p = \prod_{i \in [m]} e(P_0, K_{2,k})^{\mu_k} \cdot e(P_{2,\rho(x)}, K_{3,k})^{\mu_k}$$

$$D_I = \frac{e(K_0, Q_i) \cdot e(K_0', Q''_i)}{e(K_1, Q_i')} \cdot \frac{e^2(R_i, C_j)}{e^2(R_i, C_j')}$$

Finally, it could get $M'$ by $M' = \frac{T_i}{D_p D_I}$.

It can be easily verified that $M' = M$ will hold only when the index contained in the user’s key is not less than the number corresponding to the matrix coordinates defined in the ciphertext.

**E. Malicious User Tracing**

Before defining the tracing algorithm, let’s review the fine-grained access mechanism of the KP-ABE system. In the KP-ABE system, the user’s decryption authority is described by an access structure $\mathcal{A} = (A, \rho)$, and $A = \{A_1, \ldots, A_n\}$ is a collection of all minimal forms. For a ciphertext associated with the attribute set $x$, only $A_i$ ($i \in \{1, \ldots, n\}$) exists in $A$ such that $x \supseteq A_i$, the user has the ability to decrypt the ciphertext.

In a real scenario, a malicious user would typically trade in a decryption device that functions similarly to a decryption key. Such a decryption device takes the ciphertext as the only input, and then outputs the decryption result. During the
tracing process, we consider the decryption device provided by the malicious user as a circuit \( O \) with probability \( \epsilon \geq 0 \). And according to the decryption mechanism of the KP-ABE system, we describe its decryption authority as an access structure \( A_k \). From this, our tracing algorithm is as follows:

\[
Trace^O(pp, A_k, \epsilon) \rightarrow K \subseteq \{1, \ldots, K\}:
\]
Express \( A_k \) as its smallest form set \( A_k = \{x_1, \ldots, x_n\} \) (where \( x_n \) is an attribute set), then for \( i \in \{1, \ldots, n\} \), execute:
1. For \( k \in \{1, \ldots, K\} \), execute:
   a. Randomly selects a message \( M \) from plaintext space.
   b. Computes \( CT_{TR} \leftarrow Encrypt_{Trace}(pp, M, x_i, k) \).
   c. Sends \( CT_{TR} \) to oracle \( O \), and compares the output from \( O \) with \( M \).

2. Let \( K_i \) be the set of all \( k \) values that make the inequality
   \[ p_{i,k} - p_{i,k+1} \geq \epsilon/4K \] true.
   Output \( K = \bigcup_{1 \leq i \leq n} K_i \) as the tracing result, that is, the set of malicious users' indices.

V. SECURITY ANALYSIS

![Fig. 2. Sketch of security proof.](image)

The sketch of our security proof is shown in Fig.2. This system IV should be a secure and traceable system, therefore, our need for security is divided into two aspects:

- Message security;
- The effectiveness of the tracing algorithm.

We will reduce these security requirements to different complexity assumptions in later chapters.

A. Proof Process

1) **Proof of Theorem 1:**

**Proof.** The structure of the key-policy attribute-base encryption part is similar to the scheme in [5], hence, proof of Theorem 1 is also analogous to it. Thus, we prove the theorem by reducing the message-hiding property of our scheme in Game_{MH} to the security of the scheme in [5]. The proof details as following:

For simplicity, here we describe the KP-ABE scheme in [5] by \( \sum_{KP} \), and describe our scheme by \( \sum_{TR} \). Thus, if there is a polynomial-time adversary \( A \) that can break \( \sum_{TR} \) with a non-negligible advantage in Game_{MH}, we can construct a polynomial-time algorithm \( B \) to break \( \sum_{KP} \) with the same advantage.

**Setup.** \( B \) receives the public parameter

\[
PK_{\sum_{KP}} = (\tilde{N}, G_{\tilde{N}}, H_{\tilde{N}}, \tilde{G}_{TR}, e), g_1, g_1^{\omega}, g_1^{\omega_1}, e(g_1, h_{\tilde{N}})^{\tilde{i}},
\]

from the challenger, where \( g_1 \in G_{p_1}, h_{\tilde{N}} \in H_{\tilde{N}} \) are the generators of subgroups \( G_{p_1} \) and \( H_{\tilde{N}} \), respectively, and \( \tilde{\alpha}, \omega, \omega_0, \omega_1 \in \mathbb{Z}_{\tilde{N}} \) are random exponents. \( B \) randomly choose \( \{\alpha, r_i, z_i \in \mathbb{Z}_{\tilde{N}}\}_{i \in \{m\}}, \{c_j \in \mathbb{Z}_{\tilde{N}}\}_{j \in \{m\}}, \) a prime number \( p \) with \( N = \tilde{N} \cdot p \) and a generator \( h_p \in G_{p_1} \) of subgroup \( G_{p_1} \).

And then \( B \) gives \( A \) the public parameter \( pp \):

\[
pp_{\sum_{TR}} = ((N, G, H, \tilde{G}_{}, e), g_1, g_1^{\omega}, g_1^{\omega_1}, e(g_1, h_N)^{\tilde{i}}),
\]

where \( \tilde{\eta}, \omega, \omega_0, \omega_1 \in \mathbb{Z}_{\tilde{N}} \) are random exponents and unknown to \( B \). For the first submitted query, \( B \) randomly chooses an exponent \( \eta, j \in \mathbb{Z}_{\tilde{N}} \), two \( t - 1 \) dimensional vectors \( \bar{u}_1, \bar{u}_2 \in \mathbb{Z}_{\tilde{N}}^{t-1} \). \( A \) will accept this response with \( SK_{\sum_{TR}}^{(i,j), A_i} (k, K, K_0), \) where

\[
SK_{\sum_{TR}}^{(i,j), A_i} = \{K_{0,k} = h_{\tilde{N}}^{\tilde{\eta} \omega_1}, K_{1,k} = h_{\tilde{N}}^{\tilde{i} \omega_1}, K_{2,k} = h_{\tilde{N}}^{\tilde{i} \omega}, K_{3,k} = h_{\tilde{N}}^{\tilde{i} \omega_0}\}_{k \in [i]}
\]

The distribution of the private key is the same as that of the real scheme where \( \bar{u}_1 \) is implicitly chosen such that \( \bar{u}_1 \equiv u \) mod \( p \).

**Challenge.** \( A \) submits an access policy \( \kappa = (\rho, A) \) and two equal length messages \( M_0, M_1 \) to \( B \). \( B \) submits \( \{A_0, M_0, M_1\} \) to the challenger to get the challenge ciphertext in the form of

\[
CT_{\sum_{KP}} = (\tilde{C}_0 = g_1^s, \{\tilde{C}_{1,k} = g_1^{s \alpha_1 (\omega_0 + k \omega)} \}, \tilde{C}_{2,k} = g_1^{\tilde{\eta}}, \kappa, \tau), C = e(g_1, h_{\tilde{N}})^{\tilde{s} \cdot M_0}
\]

where \( s \) is randomly chosen and unknown to \( B \). And then, \( B \) randomly chooses exponents

\[
\kappa, \tau, \gamma_1, \ldots, \gamma_m, k_1, \ldots, k_m \in \mathbb{Z}_{\tilde{N}}
\]
\[ v_1, \ldots, d_m \in \mathbb{Z}_N^* \]

and chooses \( \mathbf{v}_2 \in \mathbb{Z}_N^* \) which makes \( v_1 \cdot \mathbf{v}_2 = 0 \). Let \( \mathbf{v}'_2 := \mathbf{v}_2 + \mathbf{v}_2 \cdot \mathbf{v}_2 \) where \( \mathbf{v}_2 \in \mathbb{Z}_N^* \), then \( \mathbf{v}'_2 \cdot \mathbf{v}_1 = \mathbf{v}_c \cdot \mathbf{v}_1 \).

For each column \( j \in [m] \):
- \( j < j \): It sets:
  \[ C_j = D_j \cdot \mathbf{v}'_2 \cdot h_p^{d_j}, \]
  \( C'_j = h_p^{d_j} \).
- \( j \geq j \): It sets:
  \[ C_j = D_j \cdot \mathbf{v}'_2 \cdot h_p^{d_j}, \]
  \( C'_j = h_p^{d_j} \).

For each row \( i \in [m] \):
- \( i < i \): It randomly chooses \( \gamma_i^j \in \mathbb{Z}_p \), \( \mathbf{v}_i \in \mathbb{Z}_N^* \) and sets:
  \[ R_i = \gamma_i^j, \quad R'_i = \gamma_i^j \cdot \mathbf{v}_i, \]
  \[ Q_i = g_i^{\gamma_i^j}, \quad Q'_i = g_i^{\gamma_i^j} \cdot \mathbf{v}_i, \quad Q''_i = g_i^{\gamma_i^j} \cdot \mathbf{v}_i \]
  \( T_i = E_i^{\gamma_i^j} \).
- \( i = i \): It randomly chooses \( \gamma_i^j \in \mathbb{Z}_p \), \( \mathbf{v}_i \in \mathbb{Z}_N^* \) which makes \( \mathbf{v}_i \cdot \mathbf{v}_c \neq \mathbf{v}_i \cdot \mathbf{v}_1 \) true and sets:
  \[ R_i = G_i^{\gamma_i^j} \cdot \mathbf{v}_i, \quad R'_i = G_i^{\gamma_i^j} \cdot \mathbf{v}_i \cdot \mathbf{v}_i, \]
  \[ Q_i = g_i^{\gamma_i^j} \cdot \mathbf{v}_i, \quad Q'_i = Q_i \cdot \mathbf{v}_i \cdot \mathbf{v}_i, \quad Q''_i = g_i^{\gamma_i^j} \cdot \mathbf{v}_i \cdot \mathbf{v}_i \]
  \( T_i = \mathbf{c} \cdot e(h_p, \mathbf{c}_0)^{\rho} \cdot E_{i}^{\gamma_i^j} \).

And \( B \) sets:
\[ P = (P_0 = \mathbf{c}_0, \{ P_{1,\rho(z)} = \mathbf{c}_1, P_{2,\rho(z)} = \mathbf{c}_2 \})_{k: x_k = 1}. \]

Finally, \( B \) sends
\[ CT_{X}^{\sum_{R}} = (x, P, \{ R_i, R'_i, Q_i, Q'_i, Q''_i, T_i \})_{j \in [m]}, \{ C_j, C'_j \}_{j \in [m]} \]
to \( A \).

**Phase 2.** As same as Phase 1.

**Guess.** \( A \) submits a \( b' \) to \( B \). And \( B \) submits \( b' \) to challenger.

All the distributions of the public parameters, private keys, and challenge ciphertexts in the game \( B \) gives \( A \) as are same as the real scheme, so we have \( \text{Adv}_{B}^{\sum_{R}} = \text{Adv}_{B}^{\sum_{K}} \) where \( \text{Adv}_{B}^{\sum_{K}} \) is the advantage of \( B \) breaking \( \sum_{K} \).

2) Proof of Theorem 2: Proof. Theorem 2 follows from the following Lemma 1 and Lemma 2.

**Lemma 1.** If the XDH assumption and the decisional linear assumption hold, then for \( j < m \), no polynomial-time adversary can distinguish between the encryptions of \( (i, j) \) and \((i, j) + 1 \).

**Proof.** If there is polynomial-time adversary \( A \) who can win the game \( \text{Game}_{111} \), then we can construct an algorithm \( B \) to solve the XDH problem with the same advantage.

**Initialize.** \( B \) gets an input of the XDH problem:
\[ (h_p, h_p^x, T) \]
This input is given on the \( p \)-order subgroup \( H_p \) of the \( N \)-order bilinear group \( H \), where \( N = p_1 p_2 p_3 \). In addition, \( B \) also obtains the values of prime factors \( p, p_1, p_2, p_3 \). \( B \) can select the elements in subgroup \( H_p \) and group \( G \) according to its own needs. \( A \) submits to \( B \) the set of attributes \( x^* \) to be challenged.

**Setup.** \( B \) randomly chooses exponents \( \alpha, \omega, \omega_1, \omega_1 \in \mathbb{Z}_p \), exponents \( \{ \alpha_i, \omega_i, z_i \in \mathbb{Z}_N \}_{i \in [m]}, \{ c_j \in \mathbb{Z}_N \}_{j \in [m]} \), generators \( g_1 \) of cyclic groups \( G_{p1} \), and element \( h \in H_{p1p2p3} \). \( B \) reveals to \( A \) with:
\[ pp = ((N, G, H, G_T, e), g_1, g_1^\alpha, g_1^{\omega}, g_1^{\omega_1}, h_p, E = e(h_1, g_1)^\alpha, \}
\[ E_i = e(h_{p1}, g_1^{\alpha_i}), Z_i = g_1^{\alpha_i} \cdot e_{i \in [m]}, \{ D_j = h_p^{d_j} \}_{j \in [m]}, \}
\[ \{ G_i = g_i^{\gamma_i^{j}} \}_{i \in [m]} \}, \{ G_i = g_i^{\gamma_i^{j}} \}_{i \in [m]} \}, \{ D_j = h_p^{d_j} \}_{j \in [m]} \}
\]

**Queries.** For responding \( A \)'s query with \( (i, j, \mathbf{a}) \), \( B \) randomly chooses \( \eta_{i,j}, \eta_{i,j}, \cdots, \eta_{i,j} \in \mathbb{Z}_N, \mathbf{a} \in \mathbb{Z}_N^{-1} \) sets:
\[ K_0 = \begin{cases} h_p^{\alpha_i} & i \neq \bar{j}, j \neq \bar{j} \\
    h_p^{\alpha_i} & i \neq \bar{j}, j \neq \bar{j} \\
h_p^{\alpha_i} & i \neq \bar{j}, j \neq \bar{j} \end{cases}, \quad K_0 = (h_p^{\bar{j}})^{\alpha_i}, \]
\[ K_1 = h_p^{\alpha_i}, \quad K_2 = h_p^{\alpha_i}, \quad K_3 = h_p^{\alpha_i} \]
\[ h_1^{\bar{j}} \cdot K_4 = h_1^{\bar{j}} (\omega_1 + \cdots + k) \cdot k \in [0]. \]

And then, \( B \) sends
\[ SK_{(i,j,\mathbf{a})} = (K_0, K_0, K_1, \{ K_2, K_3, K_4 \}_{k: x_k = 1}) \]
to \( A \).

**Challenge.** \( B \) randomly chooses exponents
\[ \kappa, \tau_1, \gamma_1, \cdots, \gamma_m, d_1, \cdots, d_m \in \mathbb{Z}_N \]
and chooses \( \mathbf{v}_2 \in \mathbb{Z}_N^* \) which makes \( \mathbf{v}_1 \cdot \mathbf{v}_2 = 0 \). Let \( \mathbf{v}'_2 := \mathbf{v}_2 + \mathbf{v}_2 \cdot \mathbf{v}_2 \) when \( \mathbf{v}_2 \in \mathbb{Z}_N^* \), then \( \mathbf{v}'_2 \cdot \mathbf{v}_1 = \mathbf{v}_c \cdot \mathbf{v}_1 \).

For each column \( j \in [m] \):
- \( j < j \): It sets:
  \[ C_j = h_p^{\gamma_i^{j}} \cdot \mathbf{v}'_2 \cdot h_p^{d_j}, \quad C'_j = h_p^{d_j} \]
- \( j = j \): It sets:
  \[ C_j = h_p^{\gamma_i^{j}} \cdot \mathbf{v}'_2 \cdot h_p^{d_j}, \quad C'_j = h_p^{d_j} \]
- \( j > j \): It sets:
  \[ C_j = h_p^{\gamma_i^{j}} \cdot \mathbf{v}'_2 \cdot h_p^{d_j}, \quad C'_j = h_p^{d_j} \]

The rest is exactly the same as the settings in IV-C. Finally, \( B \) sends
\[ CT_{X^*} = (x^*, P, \{ R_i, R'_i, Q_i, Q'_i, Q''_i, T_i \})_{j \in [m]}, \{ C_j, C'_j \}_{j \in [m]} \]
to \( A \).

It should be noted here that when \( T = h_p^{x^*} \), \( CT_{X^*} \) is normally encrypted according to \( (i, j) \), and when \( T \) is a random element from group \( H_p \). It is the same distribution as the encryption based on \( (i, j + 1) \).

**Guess.** \( A \) gives \( B \) a \( b' \), \( B \) outputs this \( b' \) as the solution to the XDH problem.

The above, \( B \) gives \( A \) the same distribution of public parameters, private keys, and challenge ciphertext as the real soltion, so \( B \)'s advantage in solving the XDH problem is the
Lemma 2. If the XDH assumption and the decisional linear assumption hold, then no polynomial-time adversary can distinguish between an encryptions of \((i, m)\) and \((i+1, 1)\) in Game\textsubscript{I\textsubscript{BE}} with non-negligible advantage.

\textbf{Proof.} To prove this lemma, we define three hybrid games:

- H1: Encrypt with \((i, j = m)\),
- H2: Encrypt with \((i, j = m + 1)\),
- H3: Encrypt with \((i + 1, 1)\).

From the following Claim 1 and Claim 2, we can see that Lemma 4 holds.

\textbf{Claim 1.} If the XDH assumption and the decisional linear assumption hold, no polynomial-time adversary can distinguish H1 and H2 with a non-negligible advantage in the game.

\textbf{Proof:} The proof of Proposition 1 is the same as the proof of Lemma 1.

\textbf{Claim 2.} If the XDH assumption and the decisional linear assumption hold, no polynomial-time adversary can distinguish H2 and H3 with a non-negligible advantage in the selection mode.

\textbf{Proof:} The indistinguishability of H2 and H3 can be proved by methods similar to Claim 5.5, 5.6 and 5.7 in [27]. Thus, we prove the theorem by reducing the message-hiding property of our scheme in Game\textsubscript{I\textsubscript{BE}} to the security of the scheme in [27].

For simplicity, here we describe the scheme in [27] by \(\sum_{I\text{BE}}\), and still describe our scheme by \(\sum_{TR}\). Thus, if there is a polynomial-time adversary \(\mathcal{A}\) that can break \(\sum_{TR}\) with a non-negligible advantage in Game\textsubscript{I\textsubscript{BE}}, we can construct a polynomial-time algorithm \(\mathcal{B}\) to break \(\sum_{I\text{BE}}\) with the same advantage.

\textbf{Setup.} \(\mathcal{B}\) receives public parameters

\[
pp_{\sum_{I\text{BE}}} = (h_p, g_1, \{G_i = g_i^{u_i}, E_i = (h_p, g_1)^{\alpha_i}, u_i \}_{i \in [m]}, \{D_j = h_p^{\omega_j} \}_{j \in [m]}).
\]

Since \((i, m + 1) \notin \{(i, j) \mid 1 \leq i, j \leq m\}\), \(\mathcal{B}\) can get all private keys of \(\sum_{I\text{BE}}\)

\[
SK_{\sum_{I\text{BE}}}^{(i,j)} = (\bar{K}_0, \bar{K}_1, \{\bar{K}_j \}_{1 \leq j \leq m, j \neq i}) = (h_p^{\omega}, h_p, \{g_1^{\alpha_i}, u_i \}_{1 \leq i \leq m, i \neq j}).
\]

\(\mathcal{B}\) randomly chooses exponents \(\omega, \omega_0, \omega_1, \alpha, \{z_i \}_{i \in [m]} \in \mathbb{Z}_N\), then sends

\[
pp_{\sum_{TR}} = (g_1, g_1^{\omega_0}, g_1^{\omega_1}, h_p, E = c(h_p, g_1)^{\alpha_i}, \{E_i, G_i, \bar{Z}_i = g_1^{z_i} \}_{i \in [m]}, \{D_j \}_{j \in [m]}).
\]

\textbf{Phase 1.} For responding \(\mathcal{A}\)'s query with \((i, j), \bar{A}\), \(\mathcal{B}\) randomly chooses \(\xi_1,..,\xi_t \in \mathbb{Z}_N, \bar{u} \in \mathbb{Z}_N^{t-1}\) sets

\[
K = (K_0 = \bar{K}_0, \prod_{1 \leq i \leq m, j \neq i} \bar{K}_j, K_0' = \bar{K}_1', K_1 = \bar{K}_1,
\]

\[
\{K_{2,k} = h_A^{(\alpha \cdot \bar{u})}, h_A^{(\bar{u}j)}, h_A^{(\bar{u}j^k)}, h_A^{(\bar{u}j^{k\omega})}, K_{3,k} = h_A^{\omega_k}, K_{4,k} = h_A^{(\bar{u}j^{k\omega+k\omega_1})} \}_{1 \leq k \leq t}.
\]

And then, it sends \(SK_{\sum_{I\text{BE}}}^{(i,j), A} = ((i, j), \bar{A}, K)\) to \(\mathcal{A}\) as response.

\textbf{Challenge.} For responding the challenge with \((M, x^t), \mathcal{B}\) lets \(Y = \{(i, j) \mid 1 \leq i, j \leq m\}\) and submit \((M, x^t, Y)\) to \(\sum_{I\text{BE}}\) to get \(CT_{\sum_{I\text{BE}}} = ((\bar{R}_i, \bar{R}_i', Q_i, Q_i', T_i)_{i \in [m]}), \{C_j, C'_j \}_{j \in [m]}\) which is in the form of:

- For every row \(i \in [m]:\)
  \[- i < i:\]
  \[
  \bar{R}_i = g_1^{\omega_i}, \bar{R}_i' = g_1^{\omega_i'}, Q_i = g_1^{\omega_i}, Q_i' = (\prod_{j \in [Y]} u_j)^{\gamma_i},
  \bar{T}_i = E_i^{\omega_i}.
  \]
  \[- i \geq i:\]
  \[
  \bar{R}_i = g_1^{\omega_i}, \bar{R}_i' = g_1^{\omega_i'}, Q_i = g_1^{\gamma_i}, Q_i' = (\prod_{j \in [Y]} u_j)^{\gamma_i},
  \bar{T}_i = E_i^{\omega_i}.
  \]

- For every column \(j \in [m]:\)
  \[- j < j:\]
  \[
  C_j = D_j^{\omega_j} \cdot h_p^{\omega j}, C'_j = h_p^{\omega j}.
  \]
  \[- j \geq j:\]
  \[
  C_j = D_j^{\omega_j} \cdot h_p^{\omega j}, C'_j = h_p^{\omega j}.
  \]

For a vector of attributes represented by \(x := (x_1, ..., x_n) \in \{0,1\}^n\), \(\mathcal{B}\) randomly chooses \(s, \{s_k \}_{k \in [\mathbb{Z}_N} \in \mathbb{Z}_N\) and computes:

\[
P = (P_0 = g_1^{\delta}, \{P_{1,p(x)} = g_1^{x_1}, P_{2,p(x)} = g_1^{\delta} \}_{k=x=1})
\]

And \(\mathcal{B}\) sets:

- For every row \(i \in [m]:\)
  \[- i < i:\]
  \[
  R_i = \bar{R}_i, R_i' = \bar{R}_i', Q_i = \bar{Q}_i, Q_i' = \bar{Q}_i' Z_i^{\bar{Z}_i g_1^{\delta}}, \bar{T}_i = \bar{T}_i.
  \]
  \[- i \geq i:\]
  \[
  R_i = \bar{R}_i, R_i' = \bar{R}_i', Q_i = \bar{Q}_i, Q_i' = \bar{Q}_i' Z_i^{\bar{Z}_i g_1^{\delta}}, \bar{T}_i = \bar{T}_i \cdot E^*.
  \]

- For every column \(j \in [m]:\)
  \[
  C_j = \bar{C}_j, C'_j = \bar{C}_j.
  \]

\(\mathcal{B}\) implicitly chooses \(\delta\) such that \(\prod_{j \in [Y]} u_j \equiv g_1^{\delta / t - \delta}\). Finally, \(\mathcal{B}\) sends

\[
CT_{x} = (x, P, \{R_i, R_i', Q_i, Q_i', T_i \}_{i \in [m]}, \{C_j, C'_j \}_{j \in [m]}).
\]

\(\mathcal{B}\) sends to \(\mathcal{A}\).

\textbf{Phase 2.} As same as Phase 1.

\textbf{Guess.} \(\mathcal{A}\) outputs a guess \(b' \in \{0,1\}\) as his guess.

\textbf{3) Proof of Theorem 3: Proof.} \textbf{Theorem 3} follows from \textbf{Lemma 3} and \textbf{Lemma 4}.

\textbf{Lemma 3.} If the scheme proposed in IV is message-hiding, then it is secure.

\textbf{Proof.} We can see that in our scheme, the default index is set to 1 when users encrypt data. In this way, the non-tracing ciphertext is only a special case in Game\textsubscript{I\textsubscript{MH}}, so the advantage of adversaries breaking through ordinary ciphertext
is the same as winning the game Game_{MH}. That is, if our scheme is message-hiding, then it is secure.

Lemma 4. If the scheme proposed in IV is index-hiding and message-hiding, then it is traceable.

Proof. The proof is similar to that in [9], [27], [31]. As in the tracing algorithm, $\mathbb{A}_O$ is expressed as its smallest form set $A_O = \{x_1, \cdots, x_n\}$. We define

$$\hat{p}_{i,k} = \Pr[\mathcal{O}(\text{Encrypt}(pp, M, x, k)) = M].$$

When $\mathcal{O}$ is a valid decryption device and $S_O$ satisfies $\mathbb{A}_O$, $p_{i,1} \geq \epsilon$. Because the ciphertext encrypted with the serial number $K + 1$ (that is, $(m, 1, 1)$) does not contain any information related to the message provided by the adversary, $p_{i,K+1}$ is negligible. Therefore, there must be $k \in [K]$ making the inequality $\hat{p}_{i,k} - \hat{p}_{i,k+1} \geq \epsilon/2K$ founded. By the Chernoff bound, $p_{i,k} - p_{i,k+1} \geq \epsilon/4K$ holds with an overwhelming probability. As a result, $K_i \neq \emptyset$. For $k \in K_i$, $\hat{p}_{i,k} - \hat{p}_{i,k+1} \geq \epsilon/4K$ holds with an overwhelming probability by the Chernoff bound. Hence, $k \in K_O$ and $x_i$ satisfying $A_k$ are both hold. In that way, $K_i \subseteq K_O$ and $\{x_i$ satisfying $A_k\}_{k \in K_i}$ are established at the same time.

VI. PERFORMANCE EVALUATION

In this section, we simulate our scheme using the C++ programming language with the GMP Library(gmp-6.1.2) and PBC Library (pbc-0.5.14). All experiments are implemented on the same computer with the following features: 1) CPU: Intel Core i7-4720; 2) RAM: 8GB; 3) OS: Ubuntu 16.04 over VMware workstation player 15.

In order to analyze the feasibility of our scheme more intuitively, we also performed simulation experiments on the [5] and [9] schemes in the same way. Specifically, our simulation experiment is divided into two parts: the evaluation of the setup phase and the evaluation of the encryption phase. For the setup phase, we performed simulation experiments on the three schemes using the two-tuple (the size of the group/the size of the index) and the length of the attribute vector used for the access control part as variables. The experimental results are presented in (a) and (b) in Fig.3.

In (a) of Fig.3, we can see that as the size of the groups and the size of the indices gradually increase the time cost in the setup phases of these three schemes has a similar upward trend. However, because the designs of the solutions are different, the actual values of the time cost are distinctly different. Overall, the time cost of our scheme at this stage is higher than the unbounded KP-ABE scheme without the tracing function from [5], and lower than the CP-ABE scheme with the same type of tracing function from [9].

The result of experiments described in (b) of Fig.3 uses attribute vectors as variables to perform simulations in different situations. We can see that for the two schemes with the Unbounded property, the time cost during the setup phase will not be affected by the length of the attribute vector at all. However, for the scheme without that, as the length of the attribute vector increases, the time cost increases significantly.

Besides, in order to realize the function of black-box tracing, our scheme and the scheme in [9] both add extra parts to the ciphertext. In the encrypt phase, the extra parts are the main reason that the schemes with black-box traceability have more time cost than the traditional ABE encryption schemes. Therefore, we performed a simulation experiment on the generation of the ciphertext added to the two schemes respectively during the encryption phase. The experimental results are displayed in (c) and (d) of Fig.3.

Fig.3 shows the change of the time cost required to generate additional ciphertext parts as the sizes of the group and the index increase while the size of the matrix is unchanged in (c), as well as (d) shows the results in the opposite case. We can find that no matter the increase of the matrix or the increase of the group and index, the time cost of the two schemes increases significantly. However, under the same circumstances, the time cost and growth rate of the scheme proposed in this paper should be smaller, and the larger the variable, the more obvious the gap.
VII. ACKNOWLEDGMENT

This work was supported in part by the National Natural Foundation of China (Grant No.61632012 and 61672239), in part by the Peng Cheng Laboratory Project of Guangdong Province (Grant No. PCL2018KP004), and in part by "the Fundamental Research Funds for the Central Universities".

REFERENCES