Unbounded Key-Policy Attribute-based Encryption with Black-Box Traceability

Yunxiu Ye*, Zhenfu Cao*†,Jiachen Shen*
*Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
†Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen and
Shanghai Institute of Intelligent Science and Technology, Tongji University, China
Email: 51184501072@stu.ecnu.edu.cn, {zfcao,jcshen}@sei.ecnu.edu.cn

***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

TABLE I
FUNCTIONALITY

Reference	Black-box Traceability	Unbounded
[5]	×	√
[9]	√	×
[24]	√	×
[25]	√	×
Ours	√	√

TABLE II
EFFICIENCY

Reference	pp	SK	CT
[9]	$\mathcal{O}(\lvert S\rvert +\sqrt{n})$	$\mathcal{O}(\lvert \mathbf{x}\rvert)$	$\mathcal{O}(\sqrt{n}+\iota)$
[25]	$\mathcal{O}(\lvert S\rvert +n)$	$\mathcal{O}(\iota)$	$\mathcal{O}(\lvert \mathbf{x}\rvert)$
Ours	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(\iota)$	$\mathcal{O}(\lvert \mathbf{x}\rvert +\sqrt{n})$

$\mid S \mid$ be the size of the attribute universe, ι the size of an policy, n the number of users in system, $\mathbf{x}$ the size of attribute set of a ciphertext.

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 ($Setup$, $KeyGen$, $Encrypt$, $Decrypt$):

$\mathbf{Setup}(\lambda, \mathcal{S}) \rightarrow (\mathbf{pp}, \mathbf{MSK})$: The system establishment algorithm includes two input parameters, namely λ, the system security parameter, and $\mathcal{S}$, the global attribute set. After running the system establishment algorithm, the public parameter pp and the system master key MSK will be output.

$\mathbf{KeyGen}(\mathbf{pp}, \mathbf{MSK}, \mathbb{A}) \rightarrow \mathbf{SK}_{\mathbb{A}}$: 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 $\mathbb{A}$ corresponding to the attributes owned by the user as input, and then outputs the private key $SK_{\mathbb{A}}$.

$\mathbf{Encrypt}(\mathbf{pp}, \mathbf{x}, \mathbf{M}) \rightarrow \mathbf{CT}_{\mathbf{x}}$: Encryption algorithm is used to encrypt plaintext messages. It takes the system public parameter pp, an attribute set $\mathbf{x}$, and the plaintext message M as input, and outputs the encrypted ciphertext $CT_{\mathbf{x}}$. Note that the attribute set $\mathbf{x}$ is publicly given in ciphertext $CT_{\mathbf{x}}$.

$\mathbf{Decrypt}(\mathbf{pp}, \mathbf{CT}_{\mathbf{x}}, \mathbf{SK}_{\mathbb{A}}) \rightarrow \mathbf{M} \mid \perp$: The decryption algorithm takes the system public parameters pp, a ciphertext CT_x and a private key $SK_{\mathbb{A}}$ as input. If the attribute set in the ciphertext satisfies the access policy in the private key, it would output the corresponding plaintext, otherwise, $\perp$.

Correctness. It requires that for all (pp, MSK) $\leftarrow$ Setup (λ, $\mathcal{S}$), all $SK_{\mathbb{A}} \leftarrow$ KeyGen (pp, MSK, $\mathbb{A}$) and all $CT_{\mathbf{x}} \leftarrow$ Encrypt (pp, x, M),

$$Pr[Decrypt(pp, CT_{\mathbf{x}}, SK_{\mathbb{A}}) = M] = 1 \text{ holds},$$

when the $\mathbf{x}$ in CT_x satisfies the access structure $\mathbb{A}$ in $SK_{\mathbb{A}}$.

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

B. 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 $\mathcal{G}$ be a group generation algorithm with security parameter λ as input and a tupe 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 $N = pp_1p_2p_3$ and $e : G \times H \rightarrow G_T$ 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 and G_T as well the bilinear map e are computable in deterministic polynomial-time respect to λ.

Let G_p, 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, 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 ($\forall g_i \in G_{p_i}$,$h_i \in H_{p_j}$, $i \neq j$, $e(g_i, h_j) = 1$). Further, for any element T $\in G$, T can be uniquely expressed as the product of an element in G_p, 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 $\mathcal{G}$, we define the following distribution:

$$\begin{gathered}\mathbf{I} := (N = pp_1p_2p_3, G, H, G_T, e) \leftarrow_R \mathcal{G}(\lambda),\\ g_1 \leftarrow_R G_{p_1},\ g_2 \leftarrow_R G_{p_2},\ g_3 \leftarrow_R G_{p_3},\\ h_1 \leftarrow_R H_{p_1},\ h_3 \leftarrow_R H_{p_3},\ h_{12} \leftarrow_R H_{p_1p_2},\\ D = (g_1, g_2, g_3, h_1, h_3, h_{12}),\\ T_1 \leftarrow_R G_{p_1},\ T_2 \leftarrow_R G_{p_1p_2}.\end{gathered}$$

Then we define the advantage of an algorithm $\mathcal{A}$ in breaking $(p_1 \rightarrow p_1p_2) - subgroup\ decision\ assumption$ to be:

$$\begin{aligned}\mathbf{Adv}^{\mathcal{A}}_{(p_1 \rightarrow p_1p_2)}(\lambda) = |Pr[\mathcal{A}(\mathbf{I}, D, T_1) =\\ 1] - Pr[\mathcal{A}(\mathbf{I}, D, T_2) = 1]|.\end{aligned}$$

$\mathbf{(G_{p_1 \rightarrow p_1p_2}) - subgroup\ decision\ assumption}$. We say that $(p_1 \rightarrow p_1p_2) - subgroup\ decision\ assumption$ holds for generator $\mathcal{G}$ if for all polynomial-time algorithms $\mathcal{A}$, $Adv^{\mathcal{A}}_{(p_1 \rightarrow p_1p_2)}(\lambda)$ is a negligible function of λ.

By exchanging the roles of G and H and/or permuting the indices for subgroups, one can define $(G_{p_1 \rightarrow p_1p_3}) - subgroup\ decision\ assumption$, $(G_{p_3 \rightarrow p_3p_2}) - subgroup\ decision\ assumption$, $(H_{p_1 \rightarrow p_1p_2}) - subgroup\ decision\ assumption$, and $(H_{p_1 \rightarrow p_1p_3}) - subgroup\ decision\ assumption$.

Subgroup Decision Diffie-Hellman Assumption. For a generator $\mathcal{G}$, we define the following distribution:

$$\begin{gathered}\mathbf{I} := (N = pp_1p_2p_3, G, H, G_T, e) \leftarrow_R \mathcal{G},\\ g_1 \leftarrow_R G_{p_1},\ g_2 \leftarrow_R G_{p_2},\ g_3 \leftarrow_R G_{p_3},\\ h_1 \leftarrow_R H_{p_1},\ h_2 \leftarrow_R H_{p_2},\ h_3 \leftarrow_R H_{p_3},\\ x, y, z \leftarrow_R \mathbb{Z}_N,\\ D = (g_1, g_2, g_3, h_1, h_2, h_3),\\ T_1 = (h_1^x, h_1^y, h_1^{xy}),\ T_2 = (h_1^x, h_1^y, h_1^{xy+z}).\end{gathered}$$

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

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

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

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

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

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

$$\mathbf{Adv}^{\mathcal{A}}_{p}(\lambda) = |Pr[\mathcal{A}(\mathbf{I}, D, T_1) = 1] - Pr[\mathcal{A}(\mathbf{I}, D, T_2) = 1]|.$$

External Diffie-Hellman Assumption. For an asymmetrical bilinear mapping $e : G \times H \rightarrow G_T$, 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 $\mathbb{A}$ contains the set of authorized attributes. With a collection of all attributes in the system denoted by $\{P_1,...,P_n\}$, we define $\mathcal{A}$ including all the access structures for the attribute set, which has

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

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

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

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

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

$$\text{x satisfies } (\mathrm{A}, \rho) \text{ iff } \mathbf{1} \in \mathrm{span}\langle A_{\mathrm{x}} \rangle,$$

Here, $\mathbf{1}$:=$(1, 0, \ldots, 0)^\top \in \mathbb{Z}^{1 \times \iota'}$ 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 $\mathbb{Z}_p$.

$$\sum_{j:x_{\rho(j)=1}} \omega_j A_j = 1, \tag{1}$$

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, ρ is a permutation and $\iota = n$. By re-ordering the rows of A, we may assume WLOG that ρ 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

$$(\{\mathrm{v_j}\}_{\mathrm{j:x_j=1}}, \{\mathrm{A_j}\begin{pmatrix}\alpha\\ \bar{u}\end{pmatrix} + \mathrm{r_j v_j}, \mathrm{r_j}\}_{\mathrm{j\in[n]}})$$

perfectly hide α, where the randomness is taken over $v_j \leftarrow_R \mathbb{Z}_p$, $\bar{u} \leftarrow_R \mathbb{Z}_p^{\iota'-1}$, and for any fixed $r_j \neq 0$.

III. Problem Statement

A. System Model

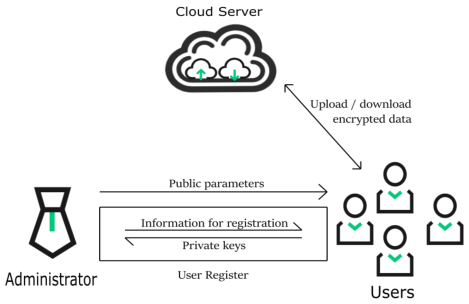


Fig. 1. System Model

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 $\mathcal{A}$ get $\mathcal{K}$ and λ as inputs:

Setup. The challenger runs $Setup(\lambda)$ and gives the public parameter pp to $\mathcal{A}$.

Phase 1. For $k = 1$ to q, $\mathcal{A}$ adaptively submits $\mathbb{A}_k = (\rho, \mathbf{A})$, and the challenger responds with $SK_{k,\mathbb{A}_k}$.

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

Phase 2. For $k = q+1$ to $\mathcal{K}'$ ($\mathcal{K}' \leq \mathcal{K}$), $\mathcal{A}$ adaptively submits $\mathbb{A}_k = (\rho, \mathbf{A})$, and the challenger responds with $SK_{k,\mathbb{A}_k}$.

Guess. $\mathcal{A}$ outputs a guess $b^{'} \in \{0,1\}$ for b.

$\mathbf{Game_{MH}}$: In the Challenge phase the challenger sends $CT \leftarrow Encryt(pp, M_b, \mathbf{x}^*)$ to $\mathcal{A}$. $\mathcal{A}$ wins the game if $b^{'} = b$ under the restriction that $\mathbf{x}^*$ cannot be satisfied by any of the queried combinations of attributes $\mathbb{A}_1$,...,$\mathbb{A}_{\mathcal{K}'}$. The advantage

of $\mathcal{A}$ is defined as $\mathbf{Adv_{MH}} = | Pr[b^{'} = b] - \frac{1}{2} |$. A scheme is message-hiding if for all polynomial-time adversaries $\mathcal{A}$ the advantage $\mathbf{Adv_{MH}}$ are negligible in λ.

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

We describe tracing capability through the next security game called $\mathbf{Game_{IH}}$. 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 $Encrypt\ (pp, M, \mathbf{x}, k)$ and $Encrypt\ (pp, M, \mathbf{x}, k+1)$.
2. Even if the adversary holds the key $SK_{k,\mathbb{A}}$, when $\mathbf{x}$ does not satisfy the access structure $\mathbb{A}$, 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 $\mathcal{A}$.

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

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

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

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

$\mathbf{Game_{IH}}$: $\mathcal{A}$ wins the game if b' = b under the restriction that none of the pairs $(k, \mathbb{A}_k)$ satisfies $(k = \bar{k}) \wedge (\mathbf{x}^*$ satisfies $\mathbb{A}_k)$. The advantage of $\mathcal{A}$ is defined as $\mathbf{Adv_{IH}}$ = $|Pr[b'=b]-\frac{1}{2}|$. A scheme is index-hiding if for all polynomial-time adversaries $\mathcal{A}$ the advantage $\mathbf{Adv_{IH}}$ are negligible in λ.

Theorem 2. If the *XDH assumption* and the *decisional linear assumption* hold, then no polynomial-time adversary can win the game $\mathbf{Game_{IH}}$ 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 in an asymmetric compound order bilinear group (G, H, G_T), 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 $s_{k(k\in[l])}$ are fresh randomness used in encryption. Next, we need to bind the $s_k(\omega_0 + k\omega_1)$s together via some common randomness s. It suffices to use $s\omega + s_k(\omega_0 + k\omega_1)$ in the ciphertext.

Besides, in order to implement an effective tracing algorithm, we assume that the number of users in the system is m^2. If the number of users is not a square, then fill with some virtual users until the nearest square is satisfied. Thus, we can associate each user in the system with a location in the $m \times m$ matrix M. In addition, our ciphertext is composed of row components and column components. Through such a structure to ensure that the ciphertext with (i, j) as the encryption parameter, only the users whose index $k \leq (i-1) \times m + y$ can decrypt the message. In this way, we can locate the users involved in the construction of the decryption device only by constructing some tracing ciphertext without any details of the private keys.

Nations. We use $\mathcal{K}$ to represent the total number of users in the system. Each user corresponds to the position in the matrix $M^{m\times m}$. The user assigned an index $k = (i-1)\times m+j$ 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 $g^{\boldsymbol{v}} = (g^{v_1}, g^{v_2}, \cdots, g^{v_n})$ and $g^{\boldsymbol{v'}} = (g^{v_1'}, g^{v_2'}, \cdots, g^{v_n'})$, there is $g^{\boldsymbol{v}} \cdot g^{\boldsymbol{v'}} = (g^{v_1+v_1'}, g^{v_2+v_2'}, \cdots, g^{v_n+v_n'})$. Similarly, e is a bilinear mapping, and $e_{\boldsymbol{n}}(g^{\boldsymbol{v}}, g^{\boldsymbol{v'}}) = \Pi_{i\in[n]} e(g^{v_i}, g^{v_i'})$.

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:

$\mathbf{Setup}(\lambda, \mathbf{m}) \rightarrow (\mathbf{pp}, \mathbf{MSK})$. The system setup algorithm takes the system security parameter λ and the matrix size m as input. Firstly run the group generation algorithm to get $\mathcal{G}(\lambda)\rightarrow(N{=}pp_1p_2p_3, G, H, G_T, \mathrm{e})$. Then, the algorithm randomly choose exponents α, $\omega, \omega_0, \omega_1 \in \mathbb{Z}_p$, exponents $\{\alpha_i, r_i, z_i \in \mathbb{Z}_N\}_{i\in[m]}$, $\{c_j \in \mathbb{Z}_N\}_{j\in[m]}$, and randomly choose generators h, h_1, h_p, g_1 of cyclic groups $H_{p_1p_2p_3}, H_{p_1}, H_p, G_{p_1}$. It sets public parameters as:

$$pp = ((N, G, H, G_T, e), g_1, g_1^{\omega}, g_1^{\omega_0}, g_1^{\omega_1}, h_p,\ E = e(h, g_1)^{\alpha},$$
$$\{E_i = e(h_p, g_1)^{\alpha_i}, G_i = g_1^{r_i}, Z_i = g_1^{z_i}\}_{i\in[m]}, \{D_j = h_p^{c_j}\}_{j\in[m]})$$

The master secret key is set as

$$MSK = (h, h_1, \alpha, \alpha_1, ..., \alpha_m, r_1, ..., r_m, c_1, ..., c_m, \omega, \omega_0, \omega_1)$$

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 structure expressed in a monotone span program to the system administrator. After receiving the user's registration application information, the system administrator assigns the position in the user matrix and generates the private key through the access structure provided by the user. The operation of the

system administrator can correspond to the key generation algorithm in the standard KP-ABE scheme and is described as follows:

$\mathbf{KeyGen}(\mathbf{pp}, \mathbf{MSK}, \mathbb{A} = (\mathbf{A}, \rho)) \rightarrow \mathbf{SK}_{(\mathbf{i},\mathbf{j}),\mathbb{A}}$. $\mathbb{A}$ is a monotone span program submitted by the user where $A \in \mathbb{Z}_N^{\iota \times n}$ is a matrix. ρ is a mapping which maps each row of A to an attribute. Then, it randomly chooses exponents $\eta_{i,j}$,ξ_1,...,ξ_ι $\in \mathbb{Z}_N$, $\bar{\boldsymbol{u}} \in \mathbb{Z}_N^{n-1}$ and computes:

$K = (K_0 = h_p^{r_i c_j + \alpha_i + \eta_{i,j}}, K_0' = (h_p^{z_i})^{\eta_{i,j}}, K_1 = h_p^{\eta_{i,j}}, \{K_{2,k} = h^{A_k \begin{pmatrix} \alpha \\ \bar{\boldsymbol{u}} \end{pmatrix}} h_p^{A_k \begin{pmatrix} \eta_{i,j} \\ \bar{\boldsymbol{u}} \end{pmatrix}} h_1^{\xi_k \omega}, K_{3,k} = h_1^{\xi_k}, K_{4,k} = h_1^{\xi_k(\omega_0 + k\omega_1)}\}_{k \in [\iota]})$

Finally, it outputs

$$SK_{(i,j),\mathbb{A}} = ((i,j), \mathbb{A}, K)$$

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.

$\mathbf{Encrypt}(\mathbf{pp}, \mathbf{M}, \mathbf{x}, (\bar{\mathbf{i}}, \bar{\mathbf{j}})) \rightarrow \mathbf{CT}_\mathbf{x}$. For a vector of attributes represented by $\mathbf{x}$:= $(x_1, ..., x_n) \in \{0,1\}^n$, the algorithm randomly chooses s, $\{s_k\}_{k \in \iota} \in \mathbb{Z}_N$ and computes: $P = (P_0 = g_1^s, \{P_{1,\rho(x)} = g_1^{s\omega} g_1^{s_k(\omega_0 + k \cdot \omega_1)}, P_{2,\rho(x)} = g_1^{s_k}\}_{k:x_k=1})$

And then, it randomly chooses exponents

$$\kappa,\tau,\gamma_1,...,\gamma_m,t_1,...,t_m \in \mathbb{Z}_N$$
$$\boldsymbol{v}_1, \boldsymbol{v}_c, \boldsymbol{d}_1,...,\boldsymbol{d}_m \in \mathbb{Z}_N^2$$

and $\boldsymbol{v}_2 \in \mathbb{Z}_N^2$ which makes $\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = 0$ true. Let $\boldsymbol{v}_c' := \boldsymbol{v}_c + v_N \cdot \boldsymbol{v}_2$ where $v_N \in \mathbb{Z}_N$, then $\boldsymbol{v}_c' \cdot \boldsymbol{v}_1 = \boldsymbol{v}_c \cdot \boldsymbol{v}_1$.

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

- $j < \bar{j}$: It sets:
$$C_j = D_j^{\tau \boldsymbol{v}_c'} \cdot h_p^{\kappa \boldsymbol{d}_j}, C_j' = h_p^{\boldsymbol{d}_j}.$$
- $j \geq \bar{j}$: It sets:
$$C_j = D_j^{\tau \boldsymbol{v}_c} \cdot h_p^{\kappa \boldsymbol{d}_j}, C_j' = h_p^{\boldsymbol{d}_j}.$$

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

- $i < \bar{i}$: It randomly chooses $\gamma_i' \in \mathbb{Z}_p$, $\boldsymbol{v}_i \in \mathbb{Z}_N^2$ and sets:
$$R_i = g_1^{\boldsymbol{v}_i}, R_i' = g_1^{\kappa \boldsymbol{v}_i},$$
$$Q_i = g_1^{\gamma_i}, Q_i' = Q_i Z_i^{t_i} g_1^s, Q_i'' = g_1^{t_i},$$
$$T_i = E_i^{\gamma_i'}$$
- $i = \bar{i}$: It randomly chooses $\boldsymbol{v}_i \in \mathbb{Z}_N^2$ which makes $\boldsymbol{v}_i \cdot \boldsymbol{v}_c' \neq \boldsymbol{v}_i \cdot \boldsymbol{v}_c$ true and sets:
$$R_i = G_i^{\gamma_i \boldsymbol{v}_i}, R_i' = G_i^{\kappa \gamma_i \boldsymbol{v}_i},$$
$$Q_i = g_1^{\tau \gamma_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, Q_i' = Q_i Z_i^{t_i} g_1^s, Q_i'' = g_1^{t_i},$$
$$T_i = M \cdot E_i^{\tau \gamma_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} \cdot E^s$$
- $i > \bar{i}$: It randomly chooses $v_N' \in \mathbb{Z}_N$. Let $\boldsymbol{v}_i := v_N' \cdot \boldsymbol{v}_1$, then $\boldsymbol{v}_i \cdot \boldsymbol{v}_c' = \boldsymbol{v}_i \cdot \boldsymbol{v}_c$. And it computes:
$$R_i = G_i^{\gamma_i \boldsymbol{v}_i}, R_i' = G_i^{\kappa \gamma_i \boldsymbol{v}_i},$$
$$Q_i = g_1^{\tau \gamma_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}, Q_i' = Q_i Z_i^{t_i} g_1^s, Q_i'' = g_1^{t_i},$$
$$T_i = M \cdot E_i^{\tau \gamma_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)} \cdot E^s$$

It returns the ciphertext as

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

Finally, the user uploads the ciphertext $CT_\mathbf{x}$ obtained by the encryption algorithm to the cloud server. It is worth noting that when generating non-tracing functional ciphertext, there is always $(\bar{i}, \bar{j}) = (1, 1)$ by default.

D. File Access

If and only if the attribute set specified by the ciphertext can satisfy the access structure corresponding to the user key, the user can successfully decrypt to obtain the correct corresponding plaintext. This stage can be described as the decryption algorithm in the standard KP-ABE system.

$\mathbf{Decrypt}(\mathbf{pp}, \mathbf{CT}_\mathbf{x}, \mathbf{SK}_{(\mathbf{i},\mathbf{j}),\mathbb{A}}) \rightarrow \mathbf{M} \mid \perp$. If $\mathbf{x}$, the set of attributes from ciphertext, satisfies the access policy (A, ρ) from $SK_{(i,j),\mathbb{A}}$, the algorithm could compute constants $\{\mu_k\}_{k \in [\iota]}$ such that

$$\textstyle\sum_{\rho(k) \in \mathbf{x}} \mu_k (A_k \cdot \begin{pmatrix} \alpha \\ \bar{\boldsymbol{u}} \end{pmatrix}) = \alpha.$$

And then, it could compute

$$D_p = \Pi_{\rho(k) \in \mathbf{x}} \frac{e(P_0, K_{2,k})^{\mu_k} \cdot e(P_{2,\rho(x)}, K_{4,k})^{\mu_k}}{e(P_{1,\rho(x)}, K_{3,k})^{\mu_k}} \tag{2}$$

$$D_I = \frac{e(K_0, Q_i) \cdot e(K_0', Q_i'')}{e(K_1, Q_i')} \cdot \frac{e^{\mathbf{2}}(R_i', C_j')}{e^{\mathbf{2}}(R_i, C_j)} \tag{3}$$

Finally, it could get M' by

$$M' = \tfrac{T_i}{D_p \cdot D_I}.$$

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 $\mathbb{A} = (A, \rho)$, and A = $\{A_1, \cdots A_n\}$ is a collection of all minimal forms. For a ciphertext associated with the attribute set $\mathbf{x}$, only A_i ($i \in \{1, \cdots, n\}$) exists in A such that $\mathbf{x} \supseteq A_i$, 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 $\mathcal{O}$ 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 $\mathbb{A}_{\mathcal{O}}$. From this, our tracing algorithm is as follows:

$Trace^{\mathcal{O}}(pp, \mathbb{A}_{\mathcal{O}}, \epsilon) \rightarrow \mathbf{K} \subseteq \{1, \cdots, \mathcal{K}\}$: Express $\mathbb{A}_{\mathcal{O}}$ as its smallest form set $A_{\mathcal{O}} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{n_{\mathcal{O}}}\}$ (where $\mathbf{x}_*$ is an attribute set), then for $i \in \{1, \cdots, n_{\mathcal{O}}\}$, execute:

1. For $k \in \{1 \cdots, \mathcal{K}\}$, execute:
   (1) The algorithm repeats the following $2\lambda(2\mathcal{K}/\epsilon)^2$ times:
      a. Randomly selects a message M from plaintext space.
      b. Computes $CT_{TR} \leftarrow Encrypt_{Trace}(pp, M, \mathbf{x}_i, k)$.
      c. Sends CT_{TR} to oracle $\mathcal{O}$, and compares the output from $\mathcal{O}$ with M.
   (2) Let $p_{i,k}$ be the proportion of times that the ciphertext correctly outputted by the oracle $\mathcal{O}$.
2. Let $\mathbf{K}_i$ be the set of all k values that make the inequality $p_{i,k} - p_{i,k+1} \geq \epsilon/4\mathcal{K}$ true.

Output $\mathbf{K} = \bigcup_{1 \leq i \leq n_{\mathcal{O}}} \mathbf{K}_i$ as the tracing result, that is, the set of malicious users' indices.

V. Security Analysis

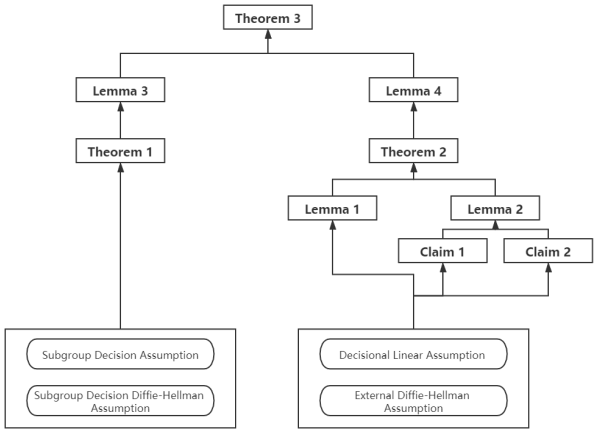


Fig. 2. Sketch of security proof.

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 $\mathbf{Game_{MH}}$ 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 $\mathcal{A}$ that can break $\sum_{TR}$ with a non-negligible advantage in $\mathbf{Game_{MH}}$, we can construct a polynomial-time algorithm $\mathcal{B}$ to break $\sum_{KP}$ with the same advantage.

Setup. $\mathcal{B}$ receives the public parameter

$$PK_{\sum_{KP}} = ((\widetilde{N}, G_{\widetilde{N}}, H_{\widetilde{N}}, \widetilde{G}_T, e), g_1, g_1^{\omega}, g_1^{\omega_0}, g_1^{\omega_1}, e(g_1, h_{\widetilde{N}})^{\widetilde{\alpha}}),$$

from the challenger, where $g_1 \in G_{p_1}$, $h_{\widetilde{N}} \in H_{\widetilde{N}}$ are the generators of subgroups G_{p_1} and $H_{\widetilde{N}}$ respectively, and $\widetilde{\alpha}, \omega, \omega_0, \omega_1 \in \mathbb{Z}_N$ are random exponents. $\mathcal{B}$ randomly choose $\{\alpha_i, r_i, z_i \in \mathbb{Z}_N\}_{(i \in [m])}$, $\{c_j \in \mathbb{Z}_N\}_{j \in [m]}$, a prime number p with $N = \widetilde{N} \cdot p$ and a generator $h_p \in G_p$ of subgroup G_p. And then $\mathcal{B}$ gives $\mathcal{A}$ the public parameter pp:

$$pp_{\sum_{TR}} = ((N, G, H, G_T, e), g_1, g_1^{\omega}, g_1^{\omega_0}, g_1^{\omega_1}, h_p,$$
$$E = e(h_{\widetilde{N}}, g_1)^{\widetilde{\alpha}} e(h_p, g_1)^{\alpha_p}, \{D_j = h_p^{c_j}\}_{j \in [m]},$$
$$\{E_i = (h_p, g_1)^{\alpha_i}, G_i = g_1^{r_i}, Z_i = g_1^{z_i}\}_{(i \in [m])}).$$

$\mathcal{B}$ implicitly chooses α such that $\widetilde{\alpha} \equiv \alpha \bmod \widetilde{N}$, $\alpha_p \equiv \alpha \bmod p$.

Phase 1. In this phase, $\mathcal{A}$ adaptively submits $(\mathbb{A}_i, (i, j))$ to $\mathcal{B}$, and $\mathcal{B}$ submits $\mathbb{A}$ to challenger to get a private key

$$SK^{\sum_{KP}}_{((i,j),\mathbb{A}_i)} = \{\widetilde{K}_{0,k} = h_{\widetilde{N}}^{A_k \begin{pmatrix} \widetilde{\alpha} \\ \bar{u} \end{pmatrix}} h_1^{\xi_k \omega}, \widetilde{K}_{1,k} = h_1^{\xi_k}, \widetilde{K}_{2,k} = h_1^{\xi_k(\omega_0 + \omega_1)}\}_{k \in [\iota]}$$

where $\widetilde{\alpha}, \omega, \omega_0, \omega_1, \xi_1, ..., \xi_k$ are randomly chosen and unknown to $\mathcal{B}$. For the first submitted query, $\mathcal{B}$ randomly chooses an exponent $\eta_{i,j} \in \mathbb{Z}_N$, two $\iota - 1$ dimensional vectors $\bar{u_1}$,$\bar{u_2}$ $\in \mathbb{Z}_N^{\iota - 1}$. $\mathcal{A}$ will receive response with $SK^{TR}_{A,\rho} = (k, K, K_0^{'})$, where

$$SK^{\sum_{TR}}_{((i,j),\mathbb{A}_i)} = (K_0 = h_p^{r_i c_j + \alpha_i + \eta_{i,j}}, K_1 = h_p^{\eta_{i,j}},$$
$$\{K_{2,k} = \widetilde{K}_{0,k} \cdot h_p^{A_k \begin{pmatrix} \alpha_p \\ \bar{u_1} \end{pmatrix}} \cdot h_p^{A_k \begin{pmatrix} \eta_{i,j} \\ \bar{u_1} \end{pmatrix}} \cdot h_1^{\xi_k \omega}, K_{3,k} =$$
$$\widetilde{K}_{1,k}, K_{4,k} = \widetilde{K}_{2,k}\}_{k \in [\iota]}, K_0^{'} = Z_i^{\eta_{i,j}})$$

The distribution of the private key is the same as that of the real scheme where $\bar{u_1}$ is implicitly chosen such that $\bar{u_1} \equiv u \bmod p$.

Challenge. $\mathcal{A}$ submits an access policy $\mathbb{A} = (\rho, \mathbf{A})$ and two equal length messages M_0, M_1 to $\mathcal{B}$. $\mathcal{B}$ summits $(\mathbb{A}, M_0, M_1)$ to the challenger to get the challenge ciphertext in the form of $CT_{\mathbf{x}}^{\sum_{KP}} =$

$$(\widetilde{C}_0 = g_1^s, \{\widetilde{C}_{1,k} = g_1^{s\omega + s_k(\omega_0 + k \cdot \omega_1)}, \widetilde{C}_{2,k} = g_1^{s_k}\}_{k: x_k = 1}, \widetilde{C} = e(g_1, h_{\widetilde{N}})^{\widetilde{\alpha} s} \cdot M_b)$$

where s is randomly chosen and unknown to $\mathcal{B}$. And then, $\mathcal{B}$ randomly chooses exponents

$$\kappa, \tau, \gamma_1, ..., \gamma_m, t_1, ..., t_m \in \mathbb{Z}_N$$

$$\boldsymbol{v}_1, \boldsymbol{v}_c, \boldsymbol{d}_1,...,\boldsymbol{d}_m \in \mathbb{Z}_N^2$$

and chooses $\boldsymbol{v}_2 \in \mathbb{Z}_N^2$ which makes $\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = 0$ true. Let $\boldsymbol{v}_c' := \boldsymbol{v}_c + v_N \cdot \boldsymbol{v}_2$ where $v_N \in \mathbb{Z}_N$, then $\boldsymbol{v}_c' \cdot \boldsymbol{v}_1 = \boldsymbol{v}_c \cdot \boldsymbol{v}_1$.

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

- $j < \bar{j}$: It sets:
$$C_j = D_j^{\tau \boldsymbol{v}_c'} \cdot h_p^{\kappa \boldsymbol{d}_j},\ C_j' = h_p^{\boldsymbol{d}_j}.$$
- $j \geq \bar{j}$: It sets:
$$C_j = D_j^{\tau \boldsymbol{v}_c} \cdot h_p^{\kappa \boldsymbol{d}_j},\ C_j' = h_p^{\boldsymbol{d}_j}.$$

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

- $i < \bar{i}$: It randomly chooses $\gamma_i' \in \mathbb{Z}_p$, $\boldsymbol{v}_i \in \mathbb{Z}_N^2$ and sets:
$$R_i = g_1^{\boldsymbol{v}_i},\ R_i' = g_1^{\kappa \boldsymbol{v}_i},$$
$$Q_i = g_1^{\gamma_i},\ Q_i' = Q_i Z_i^{t_i} g_1^s,\ Q_i'' = g_1^{t_i},$$
$$T_i = E_i^{\gamma_i'}$$
- $i = \bar{i}$: It randomly chooses $\boldsymbol{v}_i \in \mathbb{Z}_N^2$ which makes $\boldsymbol{v}_i \cdot \boldsymbol{v}_c' \neq \boldsymbol{v}_i \cdot \boldsymbol{v}_c$ true and sets:
$$R_i = G_i^{\gamma_i \boldsymbol{v}_i},\ R_i' = G_i^{\kappa \gamma_i \boldsymbol{v}_i},$$
$$Q_i = g_1^{\tau \gamma_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\ Q_i' = Q_i Z_i^{t_i} g_1^s,\ Q_i'' = g_1^{t_i},$$
$$T_i = \widetilde{C} \cdot e(h_p, \widetilde{C}_0)^{\alpha_p} \cdot E_i^{\tau \gamma_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$$
- $i > \bar{i}$: It randomly chooses $v_N' \in \mathbb{Z}_N$. Let $\boldsymbol{v}_i := v_N' \cdot \boldsymbol{v}_1$, then $\boldsymbol{v}_i \cdot \boldsymbol{v}_c' = \boldsymbol{v}_i \cdot \boldsymbol{v}_c$. And it computes:
$$R_i = G_i^{\gamma_i \boldsymbol{v}_i},\ R_i' = G_i^{\kappa \gamma_i \boldsymbol{v}_i},$$
$$Q_i = g_1^{\tau \gamma_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)},\ Q_i' = Q_i Z_i^{t_i} g_1^s,\ Q_i'' = g_1^{t_i},$$
$$T_i = \widetilde{C} \cdot e(h_p, \widetilde{C}_0)^{\alpha_p} \cdot E_i^{\tau \gamma_i (\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$$

And $\mathcal{B}$ sets

$$P = (P_0 = \widetilde{C}_0, \{P_{1,\rho(x)} = \widetilde{C}_{1,k}, P_{2,\rho(x)} = \widetilde{C}_{2,k}\}_{k:x_k=1}).$$

Finally, $\mathcal{B}$ sends

$$CT_{\mathbf{x}}^{\sum_{TR}} = (\mathbf{x}, P, \{R_i, R_i', Q_i, Q_i', Q_i'', T_i\}_{i\in[m]}, \{C_j, C_j'\}_{j\in[m]})$$

to $\mathcal{A}$.

Phase 2. As same as Phase 1.

Guess. $\mathcal{A}$ submits a $b^{'}$ to $\mathcal{B}$. And $\mathcal{B}$ submits $b^{'}$ to challenger.

All the distributions of the public parameters, private keys, and challenge ciphertexts in the game $\mathcal{B}$ gives $\mathcal{A}$ are as same as the real scheme, so we have $\mathbf{Adv}_{\mathcal{B}}^{\sum_{KP}} = \mathbf{Adv}_{\mathcal{MH}}$ where $\mathbf{Adv}_{\mathcal{B}}^{\sum_{KP}}$ is the advantage of $\mathcal{B}$ breaking $\sum_{KP}$.

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

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

$Proof$. If there is polynomial-time adversary $\mathcal{A}$ who can win the game $\mathbf{Game_{IH}}$, then we can construct an algorithm $\mathcal{B}$ to solve the XDH problem with the same advantage.

Initialize. $\mathcal{B}$ gets an input of the XDH problem:

$$(h_p, h_p^b, h_p^c, T)$$

This input is given on the p-order subgroup H_p of the N-order bilinear group H, where $N = pp_1p_2p_3$. In addition, $\mathcal{B}$ also obtains the values of prime factors p, p_1, p_2, p_3. $\mathcal{B}$ can select the elements in subgroup H_{p_1} and group G according to its own needs. $\mathcal{A}$ submits to $\mathcal{B}$ the set of attributes $\mathbf{x}^*$ to be challenged.

Setup. $\mathcal{B}$ randomly chooses exponents α, ω, ω_0, $\omega_1 \in \mathbb{Z}_p$, exponents $\{\alpha_i, r_i, z_i \in \mathbb{Z}_N\}_{i\in[m]}$, $\{c_j \in \mathbb{Z}_N\}_{j\in[m]}$, generators g_1 of cyclic groups G_{p_1}, and element $h \in H_{p_1p_2p_3}$. $\mathcal{B}$ reveals to $\mathcal{A}$ with:

$pp = ((N, G, H, G_T, e), g_1, g_1^{\omega}, g_1^{\omega_0}, g_1^{\omega_1}, h_p, E = e(h, g_1)^{\alpha},$
$\{E_i = e(h_p, g_1)^{\alpha_i}, Z_i = g_1^{z_i}\}_{i\in[m]}, \{D_j = h_p^{c_j}\}_{j\in[m]},$
$\{G_i = g_1^{r_i}\}_{i\in[m]\setminus\{\bar{i}\}}, G_{\bar{i}} = B^{r_{\bar{i}}},$
$\{D_j = h_p^{c_j}\}_{j\in[m]\setminus\{\bar{j}\}}, D_{\bar{j}} = C^{\bar{j}})$

Queries. For responding $\mathcal{A}$'s query with $((i,j), \mathbb{A})$, $\mathcal{B}$ randomly chooses $\eta_{i,j}$, $\xi_1, \cdots, \xi_\iota \in \mathbb{Z}_N$, $\bar{\boldsymbol{u}} \in \mathbb{Z}_N^{n-1}$ sets:

$$K_0 = \begin{cases} = h_p^{\alpha_i} h_p^{r_i c_j} h_p^{\eta_{i,j}}, : i \neq \bar{i}, j \neq \bar{j} \\ = h_p^{\alpha_i} B^{r_i c_j} h_p^{\eta_{i,j}}, : i = \bar{i}, j \neq \bar{j} \\ = h_p^{\alpha_i} C^{r_i c_j} h_p^{\eta_{i,j}}, : i \neq \bar{i}, j = \bar{j} \\ = h_p^{\alpha_i} h_p^{\eta_{i,j}}, : i = \bar{i}, j = \bar{j} \end{cases}, K_0' = (h_p^{z_i})^{\eta_{i,j}},$$

$K_1 = h_p^{\eta_{i,j}} \cdot h_p^{A_k\begin{pmatrix}\alpha \\ \bar{\boldsymbol{u}}\end{pmatrix}}, \{K_{2,k} = h^{A_k\begin{pmatrix}\eta_{i,j} \\ \bar{\boldsymbol{u}}\end{pmatrix}} h_1^{\xi_k \omega}, K_{3,k} = h_1^{\xi_k}, K_{4,k} = h_1^{\xi_k(\omega_0 + k\omega_1)}\}_{k\in[\iota]}$.

And then, $\mathcal{B}$ sends

$$SK_{(i,j),\mathbb{A}} = (K_0, K_0', K_1, \{K_{2,k}, K_{3,k}, K_{4,k}\}_{k:x_k=1})$$

to $\mathcal{A}$.

Challenge. $\mathcal{B}$ randomly chooses exponents

$$\kappa,\tau,\gamma_1,...,\gamma_m,t_1,...,t_m \in \mathbb{Z}_N$$
$$\boldsymbol{v}_1, \boldsymbol{v}_c, \boldsymbol{d}_1,...,\boldsymbol{d}_m \in \mathbb{Z}_N^2$$

and chooses $\boldsymbol{v}_2 \in \mathbb{Z}_N^2$ which makes $\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = 0$ true. Let $\boldsymbol{v}_c' := \boldsymbol{v}_c + v_N \cdot \boldsymbol{v}_2$ where $v_N \in \mathbb{Z}_N$, then $\boldsymbol{v}_c' \cdot \boldsymbol{v}_1 = \boldsymbol{v}_c \cdot \boldsymbol{v}_1$.

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

- $j < \bar{j}$: It sets:
$$C_j = h_p^{c_j \tau \boldsymbol{v}_c'} \cdot h_p^{\kappa \boldsymbol{d}_j},\ C_j' = h_p^{\boldsymbol{d}_j}.$$
- $j = \bar{j}$: It sets:
$$C_j = T^{\tau \boldsymbol{v}_c} \cdot h_p^{\kappa \boldsymbol{d}_j},\ C_j' = h_p^{\boldsymbol{d}_j}.$$
- $j \geq \bar{j}$: It sets:
$$C_j = B^{c_j \tau \boldsymbol{v}_c} \cdot h_p^{\kappa \boldsymbol{d}_j},\ C_j' = h_p^{\boldsymbol{d}_j}.$$

The rest is exactly the same as the settings in IV-C. Finally, $\mathcal{B}$ sends

$$CT_{\mathbf{x}^*} = (\mathbf{x}*, P, \{R_i, R_i', Q_i, Q_i', Q_i'', T_i\}_{i\in[m]}, \{C_j, C_j'\}_{j\in[m]}).$$

to $\mathcal{A}$.

It should be noted here that when $T = h_p^{bc}$, $CT_{\mathbf{x}^*}$ 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. $\mathcal{A}$ gives $\mathcal{B}$ a b'. $\mathcal{B}$ outputs this b' as the solution to the XDH problem.

The above, $\mathcal{B}$ gives $\mathcal{A}$ the same distribution of public parameters, private keys, and challenge ciphertext as the real solition, so $\mathcal{B}$'s advantage in solving the XDH problem is the

same as $\mathcal{A}$'s advantage in game $\mathbf{Game_{IH}}$.

Lemma 2. If the $XDH\ assumption$ and the $decisional\ linear\ assumption$ hold, then no polynomial-time adversary can distinguish between an encryptions of $(\bar{i}, m)$ and $(\bar{i}+1, 1)$ in $\mathbf{Game_{IH}}$ with non-negligible advantage.

$Proof$. To prove this lemma, we define three hybrid games:

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

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

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.

$Proof$: The proof of Proposition 1 is the same as the proof of Lemma 1.

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.

$Proof$: 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 $\mathbf{Game_{IH}}$ to the security of the scheme in [27].

For simplicity, here we describe the scheme in [27] by $\sum_{IBE}$, 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 $\mathbf{Game_{IH}}$, we can construct a polynomial-time algorithm $\mathcal{B}$ to break $\sum_{IBE}$ with the same advantage.

Setup. $\mathcal{B}$ receives public parameters

$$pp_{\sum_{IBE}} = (h_p, g_1, \{G_i = g_1^{r_i}, E_i = (h_p, g_1)^{\alpha_i}, u_i\}_{i\in[m]}, \{D_j = h_p^{c_j}\}_{j\in[m]}).$$

Since $(\bar{i}, m+1) \notin \{(i,j) \mid 1 \le i,j \le m\}$, $\mathcal{B}$ can get all private keys of $\sum_{IBE}$

$$SK_{(i,j)}^{\sum_{IBE}} = (\widetilde{K}_0, \widetilde{K}_1, \{\widetilde{K}'_{\bar{j}}\}_{1\le\bar{j}\le m, \bar{j}\neq j}) = (h_p^{\alpha_i + r_i c_j} u_j^{\eta_{i,j}}, h_p^{\eta_{i,j}}, \{u_{\bar{j}}^{\eta_{i,j}}\}_{1\le\bar{j}\le m, \bar{j}\neq j})$$

$\mathcal{B}$ randomly chooses exponents ω, ω_0, ω_1, α, $\{z_i\}_{i\in[m]} \in \mathbb{Z}_N$, then sends

$$pp_{\sum_{TR}} = (g_1, g_1^{\omega}, g_1^{\omega_0}, g_1^{\omega_1}, h_p, E = e(h_p, g_1)^{\alpha}, \{E_i, G_i, Z_i = g_1^{z_i}\}_{i\in[m]}, \{D_j\}_{j\in[m]})$$

Phase 1. For responding $\mathcal{A}$'s query with $((i,j), \mathbb{A})$, $\mathcal{B}$ randomly chooses ξ_1,...,$\xi_\iota \in \mathbb{Z}_N$, $\vec{\boldsymbol{u}} \in \mathbb{Z}_N^{n-1}$ sets

$$K = (K_0 = \widetilde{K}_0 \cdot \prod_{1\le\bar{j}\le m, \bar{j}\neq j} \widetilde{K}'^{z_i}_{\bar{j}}, K'_0 = \widetilde{K}_1^{z_i}, K_1 = \widetilde{K}_1,$$
$$\{K_{2,k} = h^{A_k\begin{pmatrix}\alpha\\ \vec{\boldsymbol{u}}\end{pmatrix}} h_p^{A_k\begin{pmatrix}\eta_{i,j}\\ \vec{\boldsymbol{u}}\end{pmatrix}} h_1^{\xi_k\omega}, K_{3,k} = h_1^{\xi_k}, K_{4,k} = h_1^{\xi_k(\omega_0 + k\omega_1)}\}_{k\in[\iota]}).$$

And then, it sends $SK_{(i,j),\mathbb{A}}^{\sum_{TR}} = ((i,j), \mathbb{A}, K)$ to $\mathcal{A}$ as response.

Challenge. For responding the challenge with $(M, \mathbf{x}^*)$, $\mathcal{B}$ lets Y = $\{(i,j) | 1 \le i,j \le m\}$ and submit $(M, \mathbf{x}^*, Y)$ to $\sum_{IBE}$ to get $CT_{\mathbf{x}}^{\sum_{IBE}} = (\{\widetilde{R_i}, \widetilde{R'_i}, \widetilde{Q_i}, \widetilde{Q'_i}, \widetilde{Q''_i}, \widetilde{T_i}\}_{(i\in[m])}, \{\widetilde{C_j}, \widetilde{C'_j}\}_{j\in[m]})$ which is in the form of:

- For every row $i \in [m]$:
  - $i < \bar{i}$:
    $\widetilde{R_i} = g_1^{\boldsymbol{v}_i}$, $\widetilde{R'_i} = g_1^{\kappa \boldsymbol{v}_i}$,
    $\widetilde{Q_i} = g_i^{\gamma_i}$, $\widetilde{Q'_i} = (\prod_{j\in Y_i} u_j)^{\gamma_i}$,
    $\widetilde{T_i} = E_i^{\gamma'_i}$
  - $i \le \bar{i}$:
    $\widetilde{R_i} = g_1^{r_i s_i \boldsymbol{v}_i}$, $\widetilde{R'_i} = g_1^{\kappa r_i s_i \boldsymbol{v}_i}$,
    $\widetilde{Q_i} = g_i^{\tau\gamma^i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$, $\widetilde{Q'_i} = (\prod_{j\in Y_i} u_j)^{\tau\gamma_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$,
    $\widetilde{T_i} = E_i^{\tau\gamma_i(\boldsymbol{v}_i \cdot \boldsymbol{v}_c)}$
- For every column $j \in [m]$:
  - $j < \bar{j}$:
    $C_j = D_j^{\tau \boldsymbol{v}'_c} \cdot h_p^{\kappa \boldsymbol{d}_j}$, $C'_j = h_p^{\boldsymbol{d}_j}$.
  - $j \le \bar{j}$:
    $C_j = D_j^{\tau \boldsymbol{v}_c} \cdot h_p^{\kappa \boldsymbol{d}_j}$, $C'_j = h_p^{\boldsymbol{d}_j}$.

For a vector of attributes represented by $\mathbf{x}$:= $(x_1, ..., x_n) \in \{0,1\}^n$, $\mathcal{B}$ randomly chooses s, $\{s_k\}_{k\in\iota} \in \mathbb{Z}_N$ and computes: $P = (P_0 = g_1^s, \{P_{1,\rho(x)} = g_1^{s\omega} g_1^{s_k(\omega_0 + k\cdot\omega_1)}, P_{2,\rho(x)} = g_1^{s_k}\}_{k:x_k=1})$

And $\mathcal{B}$ sets:

- For every row $i \in [m]$:
  - $i < \bar{i}$:
    $R_i = \widetilde{R_i}$, $R'_i = \widetilde{R'_i}$, $Q_i = \widetilde{Q_i}$, $Q'_i = \widetilde{Q'_i} Z_i^{t_i} g_1^s g_1^{\delta}$, $Q''_i = g_1^{t_i}$, $T_i = \widetilde{T_i}$.
  - $i \le \bar{i}$
    $R_i = \widetilde{R_i}$, $R'_i = \widetilde{R'_i}$, $Q_i = \widetilde{Q_i}$, $Q'_i = \widetilde{Q'_i} Z_i^{t_i} g_1^s g_1^{\delta}$, $Q''_i = g_1^{t_i}$, $T_i = \widetilde{T_i} \cdot E^s$.
- For every column $j \in [m]$: $C_j = \widetilde{C_j}$, $C'_j = \widetilde{C'_j}$.

$\mathcal{B}$ implicitly chooses δ such that $\prod_{j\in Y_i} u_j \equiv g_1^{p_1 - \delta}$. Finally, $\mathcal{B}$ sends

$CT_{\mathbf{x}} = (\mathbf{x}, P, \{R_i, R'_i, Q_i, Q'_i, Q''_i, T_i\}_{i\in[m]}, \{C_j, C'_j\}_{j\in[m]})$ to $\mathcal{A}$.

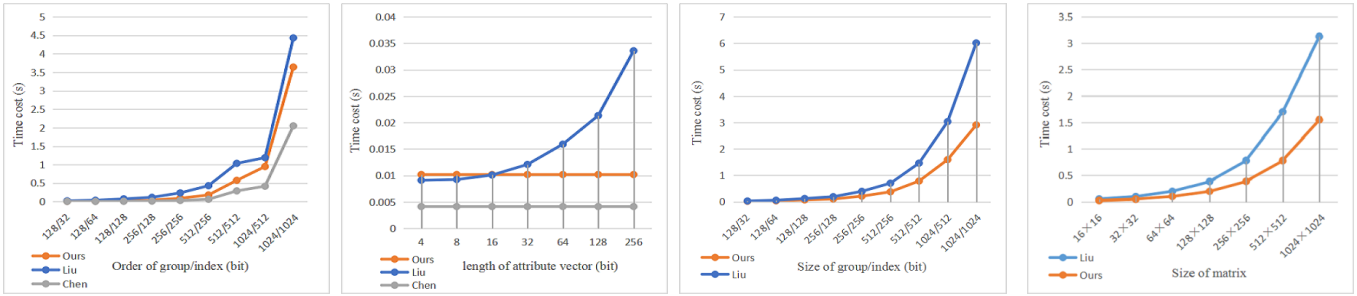
Phase 2. As same as Phase 1.

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

3) Proof of Theorem 3: $Proof$. **Theorem 3** follows from following **Lemma 3** and **Lemma 4**.

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

$Proof$. 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 $\mathbf{Game_{MH}}$, so the advantage of adversaries breaking through ordinary ciphertext



(a) (b) (c) (d)

Fig. 3. Experimental Result.
(a)On the basis that the length and width of the matrix are all 10 bits, the time cost by different group/index pairs in the initialization phase.
(b)On the basis of group/index = 128/32, the time cost of different matrix sizes in the initialization phase.
(c)On the basis of the length and width of the matrix are 10 bits, the time cost of generating additional ciphertext parts for different group/index pairs.
(d)On the basis of group/index = 128/32, the time cost of generating additional ciphertext parts for different matrix sizes.

is the same as winning the game $\mathbf{Game_{MH}}$. 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, $\mathbb{A}_{\mathcal{O}}$ is expressed as its smallest form set $A_{\mathcal{O}} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{n_{\mathcal{O}}}\}$. We define

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

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

[1] Z. Cao, "New directions of modern cryptography," *New Directions of Modern Cryptography*, p. 73, 2012.

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT*, R. Cramer, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 457–473.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," *CCS*, pp. 89–98, 2006. [Online]. Available: http://doi.acm.org/10.1145/1180405.1180418

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *S&P*, May 2007, pp. 321–334.

[5] J. Chen, J. Gong, L. Kowalczyk, and H. Wee, "Unbounded abe via bilinear entropy expansion, revisited," in *EUROCRYPT*, J. B. Nielsen and V. Rijmen, Eds. Cham: Springer International Publishing, 2018, pp. 503–534.

[6] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *EUROCRYPT*, H. Gilbert, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 62–91.

[7] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *CRYPTO*, R. Safavi-Naini and R. Canetti, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 180–198.

[8] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE TIFS*, vol. 8, no. 1, pp. 76–88, Jan 2013.

[9] Z. Liu, Z. Cao, and D. S. Wong, "Blackbox traceable cp-abe: How to catch people leaking their keys by selling decryption devices on ebay," in *CCS*, ser. CCS '13. New York, NY, USA: ACM, 2013, pp. 475–486. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516683

[10] Z. Liu, S. Duan, P. Zhou, and B. Wang, "Traceable-then-revocable ciphertext-policy attribute-based encryption scheme," *Future Generation Computer Systems*, vol. 93, pp. 903 – 913, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17320964

[11] A. Lewko and B. Waters, "Unbounded hibe and attribute-based encryption," in *EUROCRYPT*, K. G. Paterson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 547–567.

[12] T. Okamoto and K. Takashima, "Fully secure unbounded inner-product and attribute-based encryption," in *ASIACRYPT*, X. Wang and K. Sako, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 349–366.

[13] F. Z. Zehong Chen, Peng Zhang and J. Huang, "Ciphertext policy attribute-based encryption supporting unbounded attribute space from r-lwe," *ITIIS*, 2017.

[14] Z. Brakerski and V. Vaikuntanathan, "Circuit-abe from lwe: Unbounded attributes and semi-adaptive security," in *CRYPTO*, M. Robshaw and J. Katz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 363–384.

[15] C. Wang, J. Fang, and J. Xie, "Fully secure unbounded revocable key-policy attribute-based encryption scheme," in *SpaCCS*, G. Wang, I. Ray, J. M. Alcaraz Calero, and S. M. Thampi, Eds. Cham: Springer International Publishing, 2016, pp. 251–264.

[16] J. Ning, Z. Cao, X. Dong, and L. Wei, "White-box traceable cp-abe for cloud storage service: How to catch people leaking their access credentials effectively," *IEEE TDSC*, vol. 15, no. 5, pp. 883–897, Sep. 2018.

[17] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE TIFS*, vol. 10, no. 6, pp. 1274–1288, June 2015.

[18] J. Ning, Z. Cao, X. Dong, L. Wei, and X. Lin, "Large universe ciphertext-policy attribute-based encryption with white-box traceability," in *ESORICS*, M. Kutyłowski and J. Vaidya, Eds. Cham: Springer International Publishing, 2014, pp. 55–72.

[19] 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 *Computer Security – ESORICS 2015*, G. Pernul, P. Y A Ryan, and E. Weippl, Eds. Cham: Springer International Publishing, 2015, pp. 270–289.

[20] X. Li, K. Liang, Z. Liu, and D. S. Wong, "Attribute based encryption: Traitor tracing, revocation and fully security on prime order groups," in *CLOSER 2017*, 2017, pp. 281–292.

[21] Z. Liu and D. S. Wong, "Practical Attribute-Based Encryption: Traitor Tracing, Revocation and Large Universe," *The Computer Journal*, vol. 59, no. 7, pp. 983–1004, 07 2016. [Online]. Available: https://doi.org/10.1093/comjnl/bxv101

[22] J. Ning, Z. Cao, X. Dong, and L. Wei, "Traceable and revocable cp-abe with shorter ciphertexts," *Science China Information Sciences*, vol. 59, no. 11, p. 119102, Sep 2016. [Online]. Available: https://doi.org/10.1007/s11432-016-0062-7

[23] Y. Zhu, G. Gan, R. Guo, and D. Huang, "Phe: An efficient traitor tracing and revocation for encrypted file syncing-and-sharing in cloud," *IEEE Trans. Cloud Computing*, vol. 6, no. 4, pp. 1110–1124, Oct 2018.

[24] X. Fu, X. Nie, and F. Li, "Black box traceable ciphertext policy attribute-based encryption scheme," *Information*, vol. 6, no. 3, pp. 481–493, 2015.

[25] S. Xu, G. Yang, Y. Mu, and X. Liu, "Efficient attribute-based encryption with blackbox traceability," in *ProvSec*, J. Baek, W. Susilo, and J. Kim, Eds. Cham: Springer International Publishing, 2018, pp. 182–200.

[26] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisited," *J. ACM*, vol. 51, no. 4, pp. 557–594, Jul. 2004. [Online]. Available: https://doi.org/10.1145/1008731.1008734

[27] S. Garg, A. Kumarasubramanian, A. Sahai, and B. Waters, "Building efficient fully collusion-resilient traitor tracing and revocation schemes," in *CCS*, ser. CCS '10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 121–130. [Online]. Available: https://doi.org/10.1145/1866307.1866322

[28] A. Miyaji, M. Nakabayashi, and S. Takano, "Characterization of elliptic curve traces under fr-reduction," in *ICISC*, D. Won, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 90–108.

[29] A. Beimel, "Secure schemes for secret sharing and key distribution," 01 1996.

[30] M. Karchmer and A. Wigderson, "On span programs," in *CCC*, May 1993, pp. 102–111.

[31] D. Boneh, A. Sahai, and B. Waters, "Fully collusion resistant traitor tracing with short ciphertexts and private keys," in *EUROCRYPT*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 573–592.