Unbounded Key-Policy Attribute-Based Encryption With Black-Box Traceability

yunxiu ye (fdw6gw@163.com)
East China Normal University
Zhenfu Cao
East China Normal University
Jiachen Shen
East China Normal University

Research

Keywords: Attribute-based encryption, Key-policy, Traceability, Unbounded

Posted Date: October 14th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-90899/v1

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

Yunxiu Ye[*1], Zhenfu Cao[1,2,3], Jiachen Shen[1]

1. Shanghai Key Laboratory of Trustworthy Computing, East China Normal University
2. Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen
3. Shanghai Institute of Intelligent Science and Technology, Tongji University

Email: fdw6gw@163.com

Abstract：Attribute-based encryption received widespread attention as soon as it proposes. However, due to its specific characteristics, the attribute-based access control method is 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 distinguish. Once a malicious user makes illicit gains by their decryption authorities, it is difficult to trace specific users. This paper follows the practical demand to propose a more flexible key-policy attribute-based encryption scheme with black-box traceability. The scheme has a constant number of constant 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.

Keywords：Attribute-based encryption, Key-policy, Traceability, Unbounded

1. Introduction

The development of cloud computing has enabled the cloud service platform to penetrate all walks of life. An increasing number of industries and related companies use cloud service platforms to achieve business expansion to meet the changing needs of customers better. The emergence of public clouds has made data provisioning services more flexible and also has brought many security issues. For a series of security issues that follow, different aspects of security requirements are also raised, including, but not limited to, confidentiality, completeness, availability, and so on of data. The traditional public-key cryptosystem can no longer perfectly meet the needs of each specific cloud service. The coarse-grained data access mode of the single key it provides can not provide the functionality of fine-grained data sharing services in a big data environment.

For removing the limitations of the traditional public-key cryptography system and achieving fine-grained access control in the big data environment, some work proposed the corresponding

solutions, which called attribute-based encryption (ABE) [25]. ABE enables a "many-to-many" public-key cryptosystem. In an ABE system, the decryption capability of keys or the access threshold of ciphertexts is bound up with a set of attributes. Therefore, ABE technology can better meet the needs of a cloud service platform that has a large number of different roles and needs to provide big data sharing services.

Currently, ABE systems are mainly divided into two categories: key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). In an ABE system, the decryption authority of keys owned by the users and the access threshold of ciphertexts is all associated with combinations of attributes, and a structure describing the requirement of such an attribute combination is called access structure. ABE systems are classified according to partial settings of access structures. With KP-ABE, the system constructs access structures corresponding to the attributes owned by users and embeds them into private keys that are distributed to users, whereas the system with CP-ABE binds such access structures to ciphertexts. Moreover, researches based on different needs have been proposed one after another in both types.

However, ABE has brought new problems while meeting new demands. First of all, ABE systems are designed to better adapt to some changes, but there are some inherent limitations in the current structure. The number of the public parameters of most current ABE systems increases linearly with the maximum depth of the hierarchy, which does not apply to actual need. For this problem, a concept of unbounded ABE is proposed. 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 not be able to catch the traitor effectively. For protecting data privacy and interests of users, the traitor tracing mechanism has become indispensable.

2. Related work

Sahai and Waters first proposed ABE in [6], which solved the problem of fine-grained access control. Since then, Goyal et al. [7] proposed the first KP-ABE, as well as Bethencourt et al. proposed the first CP-ABE in [8], and both of them support any monotonic access tree. At present, there is a series of work on both KP-ABE and CP-ABE [4,9,10,12,17,18] according to different need to obtain better performance and achieve a higher security level.

2.1 Unbounded ABE

For some inherent limitations of ABE system design, Lewko and Waters first proposed the concept of *Unbounded* ABE in [5] and gave their solutions. Since then, Tatsuaki and Katsuyuki

have proposed the first unbounded inner-product encryption (IPE) scheme in [11]. 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 [19,20,21] that have been explored in depth. The most recent work from this perspective comes from [4]. This scheme is not only unbounded but also implements selective security, relying on simple difficulties.

2.2 Black-box Traceability

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, users with the same properties will be unable to be distinguished. If a malicious user leaks his (her) key in exchange for interest, it will be hard to trace a specific user.

To solve above problem, Liu et al. first proposed their scheme in [12] of implementing white-box tracing to solve 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 [18] to solve the same problem without obtaining any details related to users' private keys, which is more in line with the actual scene. Since then, Ning et al. have further proposed more competitive white-box tracing schemes in [13,14,15,16]. There are also a number of researches proposed like [1,17,22,23,24] aiming at various needs. [26,27] are recent results of further research on black-box tracing functionality.

3. Preliminaries

Before delving into the details of our scheme, we first review some the existing definitions and concepts.

3.1. Unbounded Key-Policy Attribute-Based Encryption

According to different sources of access structure used in attribute pair authentication, ABE has divided into ciphertext 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 $(\mathrm{Setup}, \mathrm{KeyGen}, \mathrm{Encrypt}, \mathrm{Decrypt})$:

$\mathrm{Setup}\left(\lambda, \mathrm{S}\right) \rightarrow (\mathrm{pp}, \mathrm{MSK})$: The system establishment algorithm includes two input parameters, namely λ, the system security parameter, and $\mathbf{S}$, the set of all attributes contained in the system. After running the system establishment algorithm, a public parameter $\mathbf{pp}$ and a system master key $\mathbf{MSK}$ will be output.

$\mathrm{KeyGen}(\mathrm{pp}, \mathrm{MSK}, \mathrm{A}) \rightarrow \mathrm{SK_A}$: The function of the key generation algorithm is to generate private keys for users. It takes the system public parameter $\mathbf{pp}$, the system master key $\mathbf{MSK}$ and an access policy $\mathbf{A}$ corresponding to the attributes owned by the user as input, and then outputs the private key $\mathbf{SK_A}$.

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

$\mathrm{Decrypt}(\mathrm{pp}, \mathbf{CT_x}, \mathbf{SK_A}) \rightarrow (\mathbf{M} \mid \perp)$: The decryption algorithm takes the system public parameters $\mathbf{pp}$, a ciphertext $\mathbf{CT_x}$ and a private key $\mathbf{SK_A}$ 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 $(\mathbf{pp}, \mathbf{MSK}) \leftarrow \mathbf{Setup}(\lambda, \mathbf{S})$, all$SK_A \leftarrow \mathbf{KeyGen}(\mathbf{pp}, \mathbf{MSK}, \mathbf{A})$ and all $CT_{\mathbf{x}} \leftarrow \mathbf{Encrypt}(\mathbf{pp}, \mathbf{x}, \mathbf{M})$,

$$\Pr[\mathrm{Decrypt}(\mathrm{pp}, \mathrm{CT_x}, \mathrm{SK_A}) = \mathrm{M}] = 1,$$

if the $\mathbf{x}$ in $CT_{\mathbf{x}}$ satisfies the access structure $\mathbf{A}$ in SK_A.

Unbounded [4]. An attribute-based encryption scheme is unbounded if the running time of Setup only depends on λ, otherwise, is bounded.

3.2. Bilinear Group of Composite Order

Bilinear group of composite order is firstly proposed in [28] 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 $(\mathrm{p}, \mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3, \mathrm{G}, \mathrm{H}, \mathrm{G_T}, \mathrm{e})$ as output in which $\mathrm{p}, \mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3$ are four different prime numbers determined by security parameter, $\mathrm{G}, \mathrm{H}, \mathrm{G_T}$ are three cyclic groups of order $\mathrm{N} = \mathrm{pp}_1\mathrm{p}_2\mathrm{p}_3$ and $\mathrm{e}: \mathrm{G} \times \mathrm{H} \rightarrow \mathrm{G_T}$ is a mapping that satisfies the following conditions:

- Bilinear: For $\forall g \in G, h \in H$ and $a, b \in Z_N$, $e(g^a, h^b) = e(g, h)^{ab}$;
- 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 $\mathrm{G_T}$ as well the bilinear map e are computable in deterministic polynomial-time respect to λ.

Let $\mathrm{G_p}, \mathrm{G_{p_1}}, \mathrm{G_{p_2}}, \mathrm{G_{p_3}}$ be subgroups of order $\mathrm{p}, \mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3$ in G respectively, and $H_p, H_{p_1}, H_{p_2}, H_{p_3}$ be subgroups of order $p, \mathrm{p}_1, \mathrm{p}_2, \mathrm{p}_3$ in H respectively. It is easy to know that these three subgroups are "orthogonal" to each other $(\forall \mathrm{g_i} \in \mathrm{G_{p_i}}, \mathrm{h_j} \in \mathrm{H_{p_j}}, \mathrm{i} \neq \mathrm{j}, \mathrm{e}(\mathrm{g_i}, \mathrm{h_j}) = 1)$. Further, for any element $\mathrm{T} \in \mathrm{G}$, 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.

3.2.1 Computational Assumptions

The scheme proposed in this paper will be based on four assumptions in the composite-order group, used e.g. in [4,29].

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

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

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

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

$G_{p_1 \rightarrow p_1p_2}$-*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 $\mathcal{A}$,

$$\mathbf{Adv}^{\mathcal{A}}_{(p_1 \rightarrow p_1p_2)}(\lambda)$$

is a negligible function of λ.

By exchanging the roles of G and H and/or permuting the indices for subgroups, one can define $G_{p_1 \rightarrow p_1p_2}$-*subgroup decision assumption*, $G_{3 \rightarrow p_3p_2}$-*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:

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

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

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

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_2-*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:

$$\mathrm{I} \coloneqq (p, G, G_T, e: G \times G \to G_T) \leftarrow_R \mathcal{G},$$

$$\mathrm{g} \leftarrow_{\mathrm{R}} \mathrm{G},$$

$$\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{x}, \mathrm{y} \leftarrow_{\mathrm{R}} \mathrm{Z_p},$$

$$\mathrm{D} = \left(\mathrm{g}, \mathrm{g}^{\mathrm{a}}, \mathrm{g}^{\mathrm{b}}, \mathrm{g}^{\mathrm{c}}, \mathrm{g}^{\mathrm{ax}}, \mathrm{g}^{\mathrm{by}}\right),$$

$$\mathrm{T}_1 = \mathrm{g}^{\mathrm{c(x+y)}}, \mathrm{T}_2 \leftarrow_{\mathrm{R}} \mathrm{G_T}.$$

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

$$\mathrm{Adv}_{\mathrm{p}_1}^{\mathrm{A}}\left(\lambda\right) = |\Pr[\mathcal{A}(\mathrm{I}, \mathrm{D}, \mathrm{T}_1) = 1] - \Pr[\mathcal{A}(\mathrm{I}, \mathrm{D}, \mathrm{T}_2) = 1].$$

External Diffie-Hellman Assumption. For an asymmetrical bilinear mapping $\mathrm{e}: \mathrm{G} \times \mathrm{H} \to \mathrm{G_T}$, 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 [30].

3.3. Access Control

According to the definition of the access structure in [2], in attribute-based encryption, the attribute 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 $\mathrm{P}_1, \ldots, \mathrm{P}_\mathrm{n}$, we define $\mathcal{A}$ including all the access structures for the attribute set, which has

$$2^{\{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_\mathrm{n}\}} = \{\mathcal{A} \mid \mathcal{A} \in \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_\mathrm{n}\}\}.$$

If a collection $L \in 2^{\{P_1, P_2, \ldots, P_n\}}$ has $\forall \mathcal{R}, \mathcal{Q} \in \{P_1, P_2, \ldots, P_n\}$, $\mathcal{R} \in L \wedge \mathcal{R} \subseteq \mathcal{Q} \to \mathcal{Q} \in L$, we say L is monotone. For the collection $L \subseteq 2^{P_1, P_2, \ldots, P_n} \setminus \{\emptyset\}$, we describe the sets in it as authorized set, and the unauthorized set identifies those not in L.

Monotone Span Programs [3]. A (monotone) span program for attribute universe $[\mathrm{n}]$ is a pair (A, ρ) where A is a $\mathrm{l} \times \mathrm{l}'$ matrix over $\mathrm{Z_p}$ and $\rho: \left[\mathrm{l}\right] \to [\mathrm{n}]$. Given $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, we say that

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

Here, $\mathbf{1} \coloneqq (\mathbf{1}, \mathbf{0}, \ldots, \mathbf{0})^{\top} \in Z^{1 \times \mathrm{l}'}$ is a row vector; $A_{\mathbf{x}}$ denotes the collection of vectors $\{A_{\mathbf{j}}: x_{\rho(j)} = 1\}$ where $A_{\mathbf{j}}$ denotes the j'th row of A; and span refers to linear span of collection of (row) vector over Z_p.

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

Observe that the constants $\{\omega_j\}$ can be computed in polynomial time in the size of the matrix A via Gaussian elimination. Like in [4], 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 [4]) For any x that does not satisfy A, the distributions

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

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

4. Problem Statement

4.1 System Model

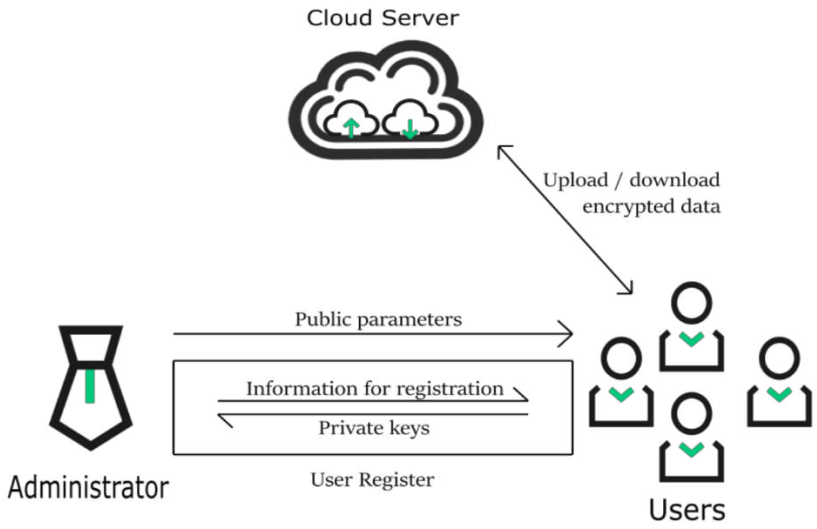


Figure 1. System Model

We use a specific example to describe our system architecture. As showing in Figure 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.

4.2 Malicious User Tracing with Black-Box

Above, we have mentioned that attribute-based encryption, due to its inherent characteristics, has some unavoidable disadvantages while implementing fine-grained access function. Unlike identity-based encryption, in an attribute-based encryption 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 [18] 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.

5. Methods

5.1 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 $k \le (i-1) \times m + y$ 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 $g^v = (g^{v_1}, g^{v_2}, \cdots, g^{v_n})$ and $g^{v'} = (g^{v'_1}, g^{v'_2}, \cdots, g^{v'_n})$, there is $g^v \cdot g^{v'} = (g^{v_1+v'_1}, g^{v_2+v'_2}, \cdots, g^{v_n+v'_n})$. Similarly, e is a biinear mapping, and $e_n(g^v, g^{v'}) = \prod_{i\in[n]} e(g^{v_i}, g^{v'_i})$.

5.2 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:

$\mathrm{Setup}(\lambda, m) \to (pp, MSK)$. 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 $\mathcal{G}(\lambda) \to (N = pp_1p_2p_3, G, H, G_T, e)$. Then, the algorithm randomly choose exponents $\alpha, \omega, \omega_0, \omega_1 \in Z_p$, exponents $\{\alpha_i, r_i, z_i \in Z_N\}_{i\in[m]}$, $\{c_j \in Z_N\}_{j\in[m]}$, 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:

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

5.3 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:

$\mathrm{KeyGen}(\mathrm{pp}, \mathrm{MSK}, \mathrm{A} = (\mathrm{A}, \rho)) \rightarrow \mathrm{SK}_{(i,j),A}$. A is a monotone span program submitted by the user where $\mathrm{A} \in \mathrm{Z}_N^{\iota \times n}$ is a matrix. ρ is a mapping which maps each row of A to an attribute. Then, it randomly chooses exponents $\eta_{i,j}, \xi_1, \ldots, \xi_\iota \in \mathrm{Z}_N, \bar{u} \in \mathrm{Z}_N^{n-1}$ and computes:

$$\mathrm{K} = (\mathrm{K}_0, = h_p^{r_i c_j + \alpha_i + \eta_{i,j}}, \mathrm{K}_0' = \left(h_p^{z_i}\right)^{\eta_{i,j}}, \mathrm{K}_1 = h_p^{\eta_{i,j}}, \{\mathrm{K}_{2,k} = h^{A_k\left(\frac{\alpha}{\bar{u}}\right)} h_p^{A_k\left(\frac{\eta_{i,j}}{\bar{u}}\right)} h_1^{\xi_k \omega}, \mathrm{K}_{3,k} = h_1^{\xi_k}, \mathrm{K}_{4,k} = h_1^{\xi_k(\omega_0 + k\omega_1)}\}_{k \in [\iota]}).$$

Finally, it outputs

$$\mathrm{SK}_{(i,j),A} = \left((i, j), A, K\right)$$

and sends to the user.

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

5.4 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.

$\mathrm{Encrypt}(\mathrm{pp}, \mathrm{M}, \mathrm{x}, (\bar{\imath}, \bar{\jmath})) \rightarrow \mathrm{CT}_x$. For a vector of attributes represented by $\mathrm{x} \coloneqq (x_1, \ldots, x_n) \in \{0,1\}^n$, the algorithm randomly chooses $s, \{s_k\}_{k \in \iota} \in \mathrm{Z}_N$ and computes:

$$\mathrm{P} = (\mathrm{P}_0 = g_1^s, \{\mathrm{P}_{1,\rho(x)} = g_1^{s\omega} g_1^{s_k(\omega_0 + k \cdot \omega_1)}, \mathrm{P}_{2,\rho(x)} = g_1^{s_k}\}_{k : x_k = 1})$$

And then, it randomly chooses exponents

$$\kappa, \tau, \gamma_1, \ldots, \gamma_m, t_1, \ldots, t_m \in Z_N$$

$$v_1, v_c, d_1, \ldots, d_m \in Z_N^2$$

and $v_2 \in Z_N^2$ which makes $v_1 \cdot v_2 = 0$ true. Let $v_c' \coloneqq v_c + v_N \cdot v_2$ where $v_N \in Z_N$, then $v_c' \cdot v_1 = v_c \cdot v_1$.

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

-$j < \bar{j}$: It sets:

$$C_j = D_j^{\tau v_c'} \cdot h_p^{\kappa d_j}, C_j' = h_p^{d_j},$$

-$j \geq \bar{j}$: It sets:

$$C_j = D_j^{\tau v_c} \cdot h_p^{\kappa d_j}, C_j' = h_p^{d_j}.$$

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

-$i < \bar{\imath}$: It randomly chooses $\gamma_i' \in Z_p, v_i \in Z_N^2$ and sets:

$$R_i = g_1^{v_i}, R_i' = g_1^{\kappa v_i},$$

$$Q_i = g_1^{\gamma_i}, Q_i' = Q_i Z_i^{t_i} g_1^s, Q_i^{'} = g_1^{t_i},$$

$$T_i = E_i^{\gamma_i'}$$

-$i = \bar{\imath}$: It randomly chooses $v_i \in Z_N^2$ which makes $v_i \cdot v_c' \neq v_i \cdot v_c$ true and sets:

$$R_i = G_i^{\gamma_i v_i}, R_i' = G_i^{\kappa \gamma_i v_i},$$

$$Q_i = g_1^{\tau \gamma_i (v_i \cdot v_c)}, Q_i^{'} = Q_i Z_i^{t_i} g_1^s, Q_i^{''} = g_1^{t_i},$$

$$T_i = M \cdot E_i^{\tau \gamma_i (v_i \cdot v_c)} \cdot E^s$$

- $i > \bar{\imath}$: It randomly chooses $v_N' \in Z_N$. Let $v_i \coloneqq v_N' \cdot v_1$, then $v_i \cdot v_c' = v_i \cdot v_c$. And it computes:

$$R_i = G_i^{\gamma_i v_i}, R_i^{'} = G_i^{\kappa \gamma_i v_i},$$

$$Q_i = g_1^{\tau \gamma_i (v_i \cdot v_c)}, Q_i' = Q_i Z_i^{t_i} g_1^s, Q_i^{''} = g_1^{t_i},$$

$$T_i = M \cdot E_i^{\tau \gamma_i (v_i \cdot v_c)} \cdot E^s$$

It returns the ciphertext as

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

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

5.5 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.

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

$$\sum_{\rho(k)\in x} \mu_k \left(\mathrm{A_k} \cdot \left(\frac{\alpha}{\vec{u}}\right)\right) = \alpha.$$

And then, it could compute

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

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

Finally, it could get $\mathrm{M'}$ by

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

It can be easily verified that $\mathrm{M'} = \mathrm{M}$ 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.

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

$\mathrm{Trace}^{\mathcal{O}}(pp, A_{\mathcal{O}}, \epsilon) \rightarrow K \subseteq \{1, \cdots, \mathcal{K}\}$: Express $A_{\mathcal{O}}$ as its smallest form set $A_{\mathcal{O}} = \{x_1, \cdots, x_{n_{\mathcal{O}}}\}$ (where 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 $CT_{TR} \leftarrow \mathrm{Encrypt}_{\mathrm{Trace}}(pp, M, x_i, k)$.

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 outputed by the oracle $\mathcal{O}$.

2. Let 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 $K = \bigcup_{1 \leq i \leq n_{\mathcal{O}}} K_i$ as the tracing result, that is, the set of malicious users' indeces.

6 Security Analysis

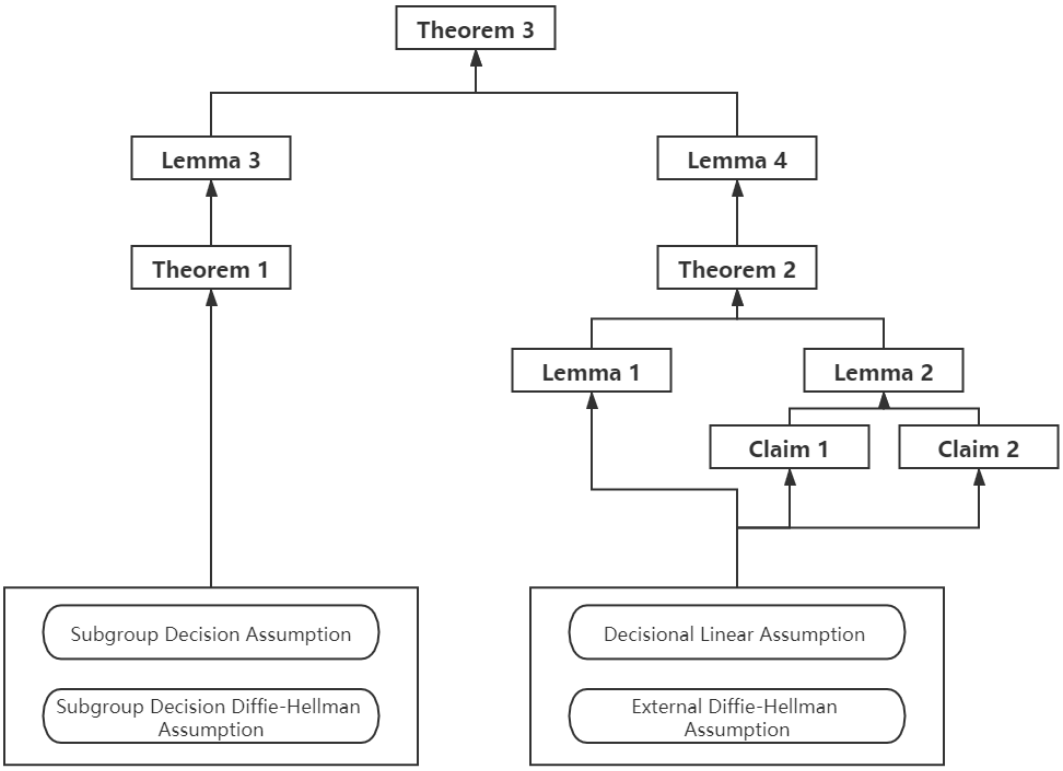


Figure 2. Sketch of security proof.

The sketch of our security proof is shown in Figure 2. This system 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.

6.1 Security Model

We define the security of the scheme 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 $\mathrm{Setup}(\lambda)$ and gives the public parameter pp to $\mathcal{A}$.

Phase1. For $\mathrm{k} = 1$ to q, $\mathcal{A}$ adaptively submits $\mathrm{A_k} = (\rho, \mathrm{A})$, and the challenger responds with $\mathrm{SK_{k,A_k}}$.

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

Phase2. For $\mathrm{k} = \mathrm{q} + 1$ to $\mathcal{K}'(\mathcal{K}' \leq \mathcal{K})$, $\mathcal{A}$ adaptively submits $\mathrm{A_k} = (\rho, \mathrm{A})$, and the challenger responds with $\mathrm{SK_{k,A_k}}$.

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

$\mathrm{Game_{MH}}$: In the Challenge phase the challenger sends $\mathrm{CT} \leftarrow \mathrm{Encryt}(\mathrm{pp}, \mathrm{M_b}, \mathrm{x^*})$ to $\mathcal{A}$. $\mathcal{A}$ wins the game if $\mathrm{b'} = \mathrm{b}$ under the restriction that $\mathrm{x^*}$ cannot be satisfied by any of the queried combinations of attributes $\mathrm{A_1}, \ldots, \mathrm{A_{\mathcal{K}'}}$. The advantage of $\mathcal{A}$ is defined as $\mathrm{Adv_{MH}} = |\Pr[\mathrm{b'} = \mathrm{b}] - \frac{1}{2}$. A scheme is message-hiding if for all polynomial-time adversaries $\mathcal{A}$ the advantage $\mathrm{Adv_{MH}}$ 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 $\mathrm{Game_{MH}}$ with non-negligible advantage.

We describe tracing capability through the next security game called $\mathrm{Game_{IH}}$. 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 $\mathrm{Encrypt}(\mathrm{pp}, \mathrm{M}, \mathrm{x}, \mathrm{k})$ and $\mathrm{Encrypt}(\mathrm{pp}, \mathrm{M}, \mathrm{x}, \mathrm{k} + 1)$.

2. Even if the adversary holds the key $\mathrm{SK_{k,A}}$, when x does not satisfy the access structure A, it should not be able to determine whether the index k or $\mathrm{k} + 1$ 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 $\mathcal{A}$.

Phase1. For $\mathrm{k} = 1$ to q, $\mathcal{A}$ adaptively submits an access policy $\mathrm{A_k} = (\rho, \mathrm{A})$ to challenger to get $\mathrm{SK_{k,A_k}}$.

Challenge. $\mathcal{A}$ 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 $\mathrm{Encrypt}(pp, M, x^*, \bar{k} + b)$ to $\mathcal{A}$.

Phase2. For $k = q + 1$ to $\mathcal{K}'$ $(\mathcal{K}' \leq \mathcal{K})$, $\mathcal{A}$ adaptively submits an access policy $A_i = (A, \rho)$ to challenger to get $SK_{k,A}$.

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

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

6.2 Proof Process

6.2.1 Proof of Theorem 1

Proof. The structure of the key-policy attribute-base encryption part is similar to the scheme in [4], 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 [4]. The proof details as following:

For simplicity, here we describe the KP-ABE scheme in [4] by $\sum \mathrm{KP}$, and describe our scheme by $\sum \mathrm{TR}$. Thus, if there is a polynomial-time adversary $\mathcal{A}$ that can break $\sum \mathrm{TR}$ with a non-negligible advantage in Game_{MH}, we can construct a polynomial-time algorithm $\mathcal{B}$ to break $\sum \mathrm{KP}$ 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 $\widetilde{\alpha}, \omega, \omega_0, \omega_1 \in Z_N$ are random exponents. $\mathcal{B}$ randomly choose $\{\alpha_i, r_i, z_i \in Z_N\}_{(i\in[m])}$, $\{c_j \in Z_N\}_{j\in[m]}$, 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 $\widetilde{\alpha} \equiv \alpha \bmod \widetilde{N}$, $\alpha_p \equiv \alpha \bmod p$.

Phase1. In this phase, $\mathcal{A}$ adaptively submits $(A_i, (i, j))$ to $\mathcal{B}$, and $\mathcal{B}$ submits A to challenger to get a private key

$$K_{((i,j),A_i)}^{\sum KP} = \{\widetilde{K_{0,k}} = h_{\widetilde{N}}^{A_k\left(\frac{\widetilde{\alpha}}{\overline{u}}\right)} h_1^{\xi_k \omega}, \widetilde{K_{1,k}} = h_1^{\xi_k}, \widetilde{K_{2,k}} = h_1^{\xi_k(\omega_0+\omega_1)}\}_{k\in[\iota]}$$

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

$$SK_{((i,j),A_i)}^{\sum TR} = (K_0 = h_p^{r_i c_j + \alpha_i + \eta_{i,j}}, K_1 = h_p^{\eta_{i,j}}, \{K_{2,k} = \widetilde{K_{0,k}} \cdot h_p^{A_k\left(\frac{\alpha_p}{\overline{u_1}}\right)} h_p^{A_k\left(\frac{\eta_{i,j}}{\overline{u_1}}\right)} h_1^{\xi_k \omega}, K_{3,k} = \widetilde{K_{1,k}}, K_{4,k} = \widetilde{K_{2,k}}\}_{k\in[\iota]}, K_0^{'} = Z_i^{\eta_{i,j}})$$

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

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

$$CT_x^{\sum KP} = (\widetilde{C_0} = g_1^s, \{\widetilde{C_{1,k}} = g_1^{s\omega + s_k(\omega_0 + k\cdot\omega_1)}, \widetilde{C_{2,k}} = g_1^{s_k}\}_{k:x_k=1}, \tilde{C} = e(g_1, h_{\widetilde{N}})^{\widetilde{\alpha} s} \cdot M_b)$$

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

$$\kappa, \tau, \gamma_1, \ldots, \gamma_m, t_1, \ldots, t_m \in Z_N$$
$$v_1, v_c, d_1, \ldots, d_m \in Z_N^2$$

and chooses $v_2 \in Z_N^2$ which makes $v_1 \cdot v_2 = 0$ true. Let $v_c' := v_c + v_N \cdot v_2$ where $v_N \in Z_N$, then $v_c' \cdot v_1 = v_c \cdot v_1$.

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

- $j < \bar{j}$: It sets:

$$C_j = D_j^{\tau v_c'} \cdot h_p^{\kappa d_j}, C_j' = h_p^{d_j}.$$

- $j \geq \bar{j}$: It sets:

$$C_j = D_j^{\tau v_c} \cdot h_p^{\kappa d_j}, C_j' = h_p^{d_j}.$$

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

- $i < \bar{\imath}$: It randomly chooses $\gamma_i' \in Z_p$, $v_i \in Z_N^2$ and sets:

$$R_i = g_1^{v_i}, R_i' = g_1^{\kappa v_i},$$
$$Q_i = g_1^{\gamma_i}, Q_i' = Q_i Z_i^{t_i} g_1^{s}, Q_i'' = g_1^{t_i},$$
$$T_i = E_i^{\gamma_i'}$$

- $i = \bar{\imath}$: It randomly chooses $v_i \in Z_N^{\mathbb{2}}$ which makes $v_i \cdot v_c' \neq v_i \cdot v_c$ true and sets:

$$R_i = G_i^{\gamma_i v_i}, R_i' = G_i^{\kappa \gamma_i v_i},$$

$$Q_i = g_1^{\tau \gamma_i (v_i \cdot v_c)}, Q_i' = Q_i Z_i^{t_i} g_1^{s}, Q_i'' = g_1^{t_i},$$

$$T_i = \tilde{C} \cdot e(h_p, \widetilde{C_0})^{\alpha_p} \cdot E_i^{\tau \gamma_i (v_i \cdot v_c)}$$

- $i > \bar{\imath}$: It randomly chooses $v_N' \in Z_N$. Let $v_i := v_N' \cdot v_1$, then $v_i \cdot v_c' = v_i \cdot v_c$. And it computes:

$$R_i = G_i^{\gamma_i v_i}, R_i' = G_i^{\kappa \gamma_i v_i},$$
$$Q_i = g_1^{\tau \gamma_i (v_i \cdot v_c)}, Q_i' = Q_i Z_i^{t_i} g_1^{s}, Q_i'' = g_1^{t_i},$$

$$T_i = \tilde{C} \cdot e(h_p, \widetilde{C_0})^{\alpha_p} \cdot E_i^{\tau \gamma_i (v_i \cdot v_c)}$$

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

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

to $\mathcal{A}$.

Phase2. 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 $Adv_B^{\sum KP} = Adv_{MH}$ where $Adv_B^{\sum KP}$ is the advantage of $\mathcal{B}$ breaking $\sum KP$.

6.2.2 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 $\mathrm{Game}_{\mathrm{IH}}$, 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:

$$\left(h_p, h_p^b, h_p^c, T\right)$$

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 x^* to be challenged.

Setup. $\mathcal{B}$ randomly chooses exponents $\alpha, \omega, \omega_0, \omega_1 \in Z_p$, exponents $\{\alpha_i, r_i, z_i \in Z_N\}_{i\in[m]}$, $\{c_j \in Z_N\}_{j\in[m]}$, 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 $((i, j), A)$, $\mathcal{B}$ randomly chooses $\eta_{i,j}, \xi_1, \cdots, \xi_\iota \in Z_N, \bar{u} \in Z_N^{n-1}$ sets:

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

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

And then, $\mathcal{B}$ sends

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

to $\mathcal{A}$.

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

$$\kappa, \tau, \gamma_1, \ldots, \gamma_m, t_1, \ldots, t_m \in Z_N$$

$$v_1, v_c, d_1, \ldots, d_m \in Z_N^2$$

and chooses $v_2 \in Z_N^2$ which makes $v_1 \cdot v_2 = 0$ true. Let $v_c' \coloneqq v_c + v_N \cdot v_2$ where $v_N \in Z_N$, then $v_c' \cdot v_1 = v_c \cdot v_1$.

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

- $j < \bar{j}$: It sets:

$$C_j = h_p^{c_j \tau v_c'} \cdot h_p^{\kappa d_j}, C_j' = h_p^{d_j}.$$

- $j = \bar{j}$: It sets:

$$C_j = T^{\tau v_c} \cdot h_p^{\kappa d_j}, C_j' = h_p^{d_j}.$$

- $j \geq \bar{j}$: It sets:

$$C_j = B^{c_j \tau v_c} \cdot h_p^{\kappa d_j}, C_j' = h_p^{d_j}.$$

The rest is exactly the same as the settings in normal system. Finally, $\mathcal{B}$ sends

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

to $\mathcal{A}$.

It should be noted here that when $T = h_p^{bc}$, 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. $\mathcal{A}$ 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 solution, so $\mathcal{B}$'s advantage in solving the XDH problem is the same as $\mathcal{A}$'s advantage in game $\mathrm{Game}_{\mathrm{IH}}$.

Lemma 2. If the *XDH assumption* and the *decisional linear assumption* hold, then no polynomial-time adversary can distinguish between an encryption of $(\bar{\imath}, m)$ and another of $(\bar{\imath}+1, 1)$ in $\mathrm{Game}_{\mathrm{IH}}$ with non-negligible advantage.

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

- H1: Encrypt with $(\bar{\imath}, \bar{\jmath} = m)$,

- H2: Encrypt with $(\bar{\imath}, \bar{\jmath} = m + 1)$,

- H3: Encrypt with $(\bar{\imath} + 1, 1)$.

From the following **Claim 1** and **Claim 2**, we can see that **Lemma 2** 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**, **Claim 5.6** and **Claim 5.7** in [29]. Thus, we prove the theorem by reducing the message-hiding property of our scheme in $\mathrm{Game}_{\mathrm{IH}}$ to the security of the scheme in [29].

For simplicity, here we describe the scheme in [29] 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 $\mathrm{Game}_{\mathrm{IH}}$, we can construct a polynomial-time algorithm $\mathcal{B}$ to break $\sum$IBE with the same advantage.

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

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

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

$$\mathrm{SK}_{(i,j)}^{\sum \mathrm{IBE}} = (\widetilde{K_0}, \widetilde{K_1}, \{\widetilde{K_{\bar{\jmath}}'}\}_{1\le\bar{\jmath}\le m,\bar{\jmath}\neq j}) = (h_p^{\alpha_i + r_i c_j} u_j^{\eta_{i,j}}, h_p^{\eta_{i,j}}, \{u_{\bar{\jmath}}^{\eta_{i,j}}\}_{1\le\bar{\jmath}\le m,\bar{\jmath}\neq j})$$

$\mathcal{B}$ randomly chooses exponents $\omega, \omega_0, \omega_1, \alpha, \{z_i\}_{i\in[m]} \in Z_N$, then sends

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

Phase1. For responding $\mathcal{A}$'s query with $\big((i,j), A\big)$, $\mathcal{B}$ randomly chooses $\xi_1, \ldots, \xi_\iota \in Z_N$, $\bar{u} \in Z_N^{n-1}$ sets

$$K = (K_0 = \widetilde{K_0} \cdot \prod_{1\le\bar{\jmath}\le m,\bar{\jmath}\neq j} \widetilde{K_{\bar{\jmath}}'}, K_0' = \widetilde{K_1^{z_i}}, K_1 = \widetilde{K_1},$$

$$\{K_{2,k} = h^{A_k\left(\frac{\alpha}{\bar{u}}\right)} h_p^{A_k\left(\frac{\eta_{i,j}}{\bar{u}}\right)} h_1^{\xi_k \omega}, K_{3,k} = h_1^{\xi_k}, K_{4,k} = h_1^{\xi_k(\omega_0 + k\omega_1)}\}_{k\in[\iota]}).$$

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

Challenge. For responding the challenge with (M, x^*), $\mathcal{B}$ lets $Y = \{\,(i,j) \mid 1 \le i,j \le m\,\}$ and submit (M, x^*, Y) to $\sum IBE$ to get $CT_x^{\sum IBE} = \big(\{\widetilde{R_\iota}, \widetilde{R'_\iota}, \widetilde{Q_\iota}, \widetilde{Q'_\iota}, \widetilde{Q''_\iota}, \widetilde{T_\iota}\}_{(i\in[m])}, \{\widetilde{C_j}, \widetilde{C'_j}\}_{j\in[m]}\big)$ which is in the form of:

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

-$i < \bar{\imath}$:

$$\widetilde{R_\iota} = g_1^{v_i}, \widetilde{R'_\iota} = g_1^{\kappa v_i},$$

$$\widetilde{Q_\iota} = g_i^{\gamma^i}, \widetilde{Q'_\iota} = \left(\prod_{j\in Y_i} u_j\right)^{\gamma_i},$$

$$\widetilde{T_\iota} = E_i^{\gamma'_i}$$

- $i \le \bar{\imath}$

$$\widetilde{R_\iota} = g_1^{r_i s_i v_i}, \widetilde{R'_\iota} = g_1^{\kappa r_i s_i v_i},$$

$$\widetilde{Q_\iota} = g_i^{\tau\gamma^i(v_i\cdot v_c)}, \widetilde{Q'_\iota} = \left(\prod_{j\in Y_i} u_j\right)^{\tau\gamma_i(v_i\cdot v_c)},$$

$$\widetilde{T_\iota} = E_i^{\tau\gamma_i(v_i\cdot v_c)}$$

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

- $j < \bar{\jmath}$:

$$C_j = D_j^{\tau v'_c} \cdot h_p^{\kappa d_j}, C'_j = h_p^{d_j}.$$

- $j \le \bar{\jmath}$:

$$C_j = D_j^{\tau v_c} \cdot h_p^{\kappa d_j}, C'_j = h_p^{d_j}.$$

For a vector of attributes represented by $x := (x_1, \dots, x_n) \in \{0,1\}^n$, $\mathcal{B}$ randomly chooses $s, \{s_k\}_{k\in\iota} \in Z_N$ and computes:

$$P = \left(P_0 = g_1^s, \{P_{1,\rho(x)} = g_1^{s\omega} g_1^{s_k(\omega_0 + k\cdot\omega_1)}, P_{2,\rho(x)} = g_1^{s_k}\}_{k:x_k=1}\right).$$

And $\mathcal{B}$ sets:

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

- $i < \bar{\imath}$:

$$R_i = \widetilde{R_i}, R_i' = \widetilde{R_i'},$$

$$Q_i = \widetilde{Q_i}, Q_i' = \widetilde{Q_i'} Z_i^{t_i} g_1^s g_1^\delta, Q_i'' = g_1^{t_i},$$

$$T_i = \widetilde{T_i}$$

- $i \leq \bar{\imath}$:

$$R_i = \widetilde{R_i}, R_i' = \widetilde{R_i'},$$

$$Q_i = \widetilde{Q_i}, Q_i' = \widetilde{Q_i'} Z_i^{t_i} g_1^s g_1^\delta, Q_i'' = g_1^{t_i},$$

$$T_i = \widetilde{T_i} \cdot E^s$$

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

$$CT_x = \left(x, P, \{R_i, R_i', Q_i, Q_i', Q_i^{'}, T_i\}_{i\in[m]}, \{C_j, C_j'\}_{j\in[m]}\right)$$

to $\mathcal{A}$.

Phase2. As same as Phase 1.

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

6.2.3 Proof of Theorem 3

Proof. **Theorem 3** follows from following **Lemma 3** and **Lemma 4**.

Lemma 3. If the scheme proposed in this paper 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 $\mathrm{Game}_{\mathrm{MH}}$, so the advantage of

adversaries breaking through ordinary ciphertext is the same as winning the game $\mathrm{Game}_{\mathrm{MH}}$. That is, if our scheme is message-hiding, then it is secure.

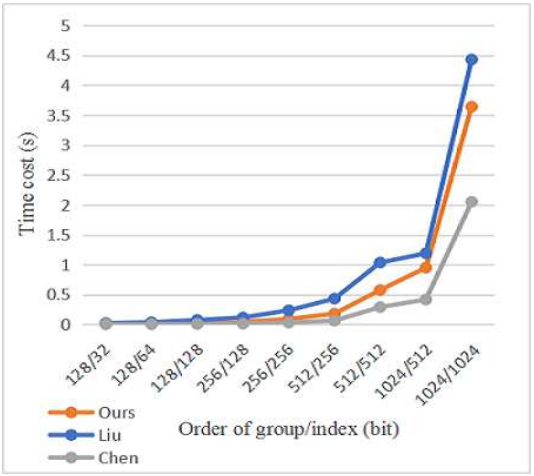
Lemma 4. If the scheme proposed in this paper is index-hiding and message-hiding, then it is traceable.

Proof. The proof is similar to that in [18,29,31]. As in the tracing algorithm, $\mathrm{A}_{\mathcal{O}}$ is expressed as its smallest form set $\mathrm{A}_{\mathcal{O}} = \{\mathrm{x}_1, \cdots, \mathrm{x}_{\mathrm{n}_{\mathcal{O}}}\}$. We define

$$\widehat{\mathrm{p}_{\mathrm{i,k}}} = \Pr\left[\mathcal{O}\left(\mathrm{Encrypt}(\mathrm{pp}, \mathrm{M}, \mathrm{x}, \mathrm{k})\right) = \mathrm{M}\right].$$

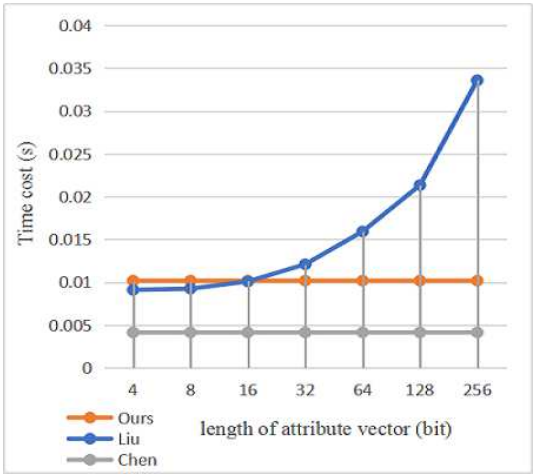
When $\mathcal{O}$ is a valid decryption device and $\mathrm{S}_{\mathcal{O}}$ satisfies $\mathrm{A}_{\mathcal{O}}$, $\mathrm{p}_{\mathrm{i},1} \geq \epsilon$. Because the ciphertext encrypted with the serial number $\mathcal{K}+1$ (that is,$(\mathrm{m}+1,1)$) does not contain any information related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

7. Experiment



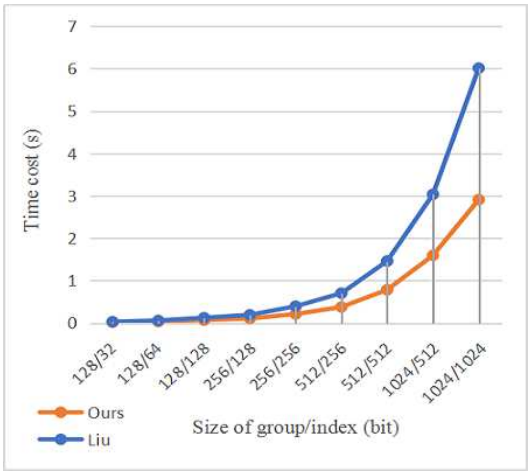
On the basis that the length and width of the matrix are all 10 bits, the time cost by different group/index pairs in the initialization phase.

Figure 3.



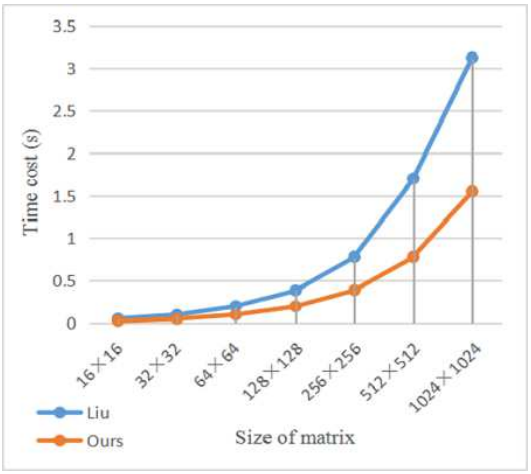
On the basis of group/index = 128/32, the time cost of different matrix sizes in the initialization phase.

Figure 4.



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.

Figure 5.



On the basis of group/index = 128/32,
the time cost of generating additional ciphertext parts for different matrix sizes.

Figure 6.

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 [4] and [18] 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 Figure 3 and Figure 4.

In Figure 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 [4], and lower than the CP-ABE scheme with the same type of tracing function from [18].

The result of experiments described in Figure 4 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 [18] 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 Figure 5 and Figure 6.

Our results also 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 Figure 5, as well as Figure 6 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.

8. Results and Discussion

Reference	Black-box Traceability	Unbounded
[4]	x	√
[18]	√	x
[26]	√	x
[27]	√	x
Ours	√	√

Table 1. Functional Comparison

In this paper, we put forward an unbounded attribute-based encryption system 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 above. 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 functionalities and efficiency. From the perspective of functionality, we compared black-box traceability, and dynamic attribute addition for five schemes in Table 1. For three of these schemes with black-box traceability and similar structure, we compared their efficiency by analyzing their data sizes in Table 2 .

Reference	Public Parameter	User's Private Key	Cipher Text
[18]	$\mathcal{O}(\lvert S\rvert + \sqrt{n})$	$\mathcal{O}(\lvert x\rvert)$	$\mathcal{O}(\sqrt{n} + \iota)$
[27]	$\mathcal{O}(\lvert S\rvert + n)$	$\mathcal{O}(\iota)$	$\mathcal{O}(\lvert x\rvert)$
Ours	$\mathcal{O}(n)$	$\mathcal{O}(\iota)$	$\mathcal{O}(\lvert x\rvert + \sqrt{n})$

$|S|$ be the size of the attribute universe;
ι be the size of an policy;
n Be the max number of users in system;
x be the size of attribute set of a ciphertest.

Table 2. Efficiency Comparison

Availability of data and materials

Data sharing not applicable to this article as no datasets are generated or analyzed during the current study.

Abbreviations

(ABE)	Central Processing Unit
(CP)	Cipher-Policy
(KP)	Key-Policy
(XDH)	External Diffie-Hellman Assumption
(DDH)	Decisional Diffie-Hellman Assumption
(IPE)	Inner-product Encryption
(CPU)	Central Processing Unit

REFERENCES

[1] Xiaoyi Li, Kaitai Liang, Zhen Liu, and Duncan S. Wong, " Attribute based Encryption: Traitor Tracing, Revocation and Fully Security on Prime Order Groups," *CLOSER,* 2017.

[2] Beimel, A, " Secure Schemes for Secret Sharing and Key Distribution," 1996.

[3] M. Karchmer and A. Wigderson, " On span programs," *CCC,* pp. 102-111, 1993.

[4] Jie Chen, Junqing Gong,Lucas Kowalczyk, and Hoeteck Wee, " Unbounded ABE via Bilinear Entropy Expansion, Revisited," *EUROCRYPT,* pp. 503-534, 2018.

[5] Lewko Allison and Waters Brent, " Unbounded HIBE and Attribute-Based Encryption," *EUROCRYPT,* 2011.

[6] Sahai Amit and Waters Brent, " Fuzzy Identity-Based Encryption," *EUROCRYPT,* pp. 457-473, 2005.

[7] Goyal Vipul, Pandey Omkant, Sahai Amit and Waters Brent, " Attribute-based Encryption for Fine-grained Access Control of Encrypted Data," *CCS,* pp. 89-98, 2006.

[8] J. Bethencourt, A. Sahai and B. Waters, " Ciphertext-Policy Attribute-Based Encryption," *S&P,* pp. 321-334, 2007.

[9] Lewko Allison, Okamoto Tatsuaki, Sahai Amit, Takashima Katsuyuki, and Waters Brent, " Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption," *EUROCRYPT,* 2010.

[10] Lewko Allison and Waters Brent, " New Proof Methods for Attribute-Based Encryption: Achieving Full Security through Selective Techniques," *CRYPTO,* 2012.

[11] Okamoto Tatsuaki and Takashima Katsuyuki, " Fully Secure Unbounded Inner-Product and Attribute-Based Encryption," *ASIACRYPT,* 2012.

[12] 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, 2013.

[13] J. Ning, Z. Cao, X. Dong, L. Wei, and X. Lin, " Large Universe Ciphertext-Policy Attribute-Based Encryption with White-Box Traceability," *ESORICS,* 2014.

[14] 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," *ESORICS,* 2015.

[15] 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, 2015.

[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, 2018.

[17] Zhenhua Liu, Shuhong Duan, Peilin Zhou, and Baocang Wang, " Traceable-then-revocable ciphertext-policy attribute-based encryption scheme," *Future*

Generation Computer Systems, vol. 93, pp. 903-913, 2019.

[18] Z. Liu, Z. Cao, and D. Wong, " Blackbox Traceable CP-ABE: How to Catch People Leaking Their Keys by Selling Decryption Devices on Ebay," *CCS,* 2013.

[19] Zehong Chen, Peng Zhang, Fangguo Zhang and Jiwu Huang, " Ciphertext policy attribute-based encryption supporting unbounded attribute space from R-LWE," *ITIIS,* 2017.

[20] Brakerski Zvika and Vaikuntanathan Vinod, " Circuit-ABE from LWE: Unbounded Attributes and Semi-adaptive Security," *CRYPTO,* 2016.

[21] Changji Wang, Jian Fang, and Jianguo Xie, " Fully Secure Unbounded Revocable Key-Policy Attribute-Based Encryption Scheme," *SpaCCS,* 2016.

[22] 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, 2018.

[23] 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, pp. 1869-1919, 2016.

[24] Zhen Liu and Duncan S. Wong, " Practical Attribute-Based Encryption: Traitor Tracing, Revocation and Large Universe," *The Computer Journal,* 2016.

[25] Zhenfu Cao, " New directions of modern cryptography," pp. 73, 2012.

[26] Xingbing Fu, Xuyun Nie, and Fagen Li, " Black Box Traceable Ciphertext Policy Attribute-Based Encryption Scheme," *Information,* vol. 6, no. 3, pp. p.481-493, 2015.

[27] Shengmin Xu, Guomin Yang, Yi Mu, and Ximeng Liu, " Efficient Attribute-Based Encryption with Blackbox Traceability," *ProvSec,* 2018.

[28] Canetti Ran, Goldreich Oded, and Halevi Shai, " The Random Oracle Methodology, Revisited," *J. ACM,* vol. 51, no. 4, pp.557-594, 2004.

[29] Sanjam Garg, Abishek Kumarasubramanian, Amit Sahai, and Brent Waters, " Building Efficient Fully Collusion-Resilient Traitor Tracing and Revocation Schemes," *CCS,*2010.

[30] Miyaji Atsuko, Nakabayashi Masaki, and Takano Shunzo, " Characterization of Elliptic Curve Traces Under FR-Reduction," *ICISC,* 2000.

[31] Dan Boneh, Amit Sahai, and Brent Waters, " Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys," *EUROCRYPT,* 2006.

Acknowledgements

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".

Author information

Affiliations

Shanghai Key Laboratory of Trustworthy Computing, East China Normal University
Yunxiu Ye

Contributions

Yunxiu Ye wrote the entire article.

Corresponding author

Correspondence to Yunxiu Ye

Ethics declarations

Competing interests

The authors declare that they have no competing interests.

Figures

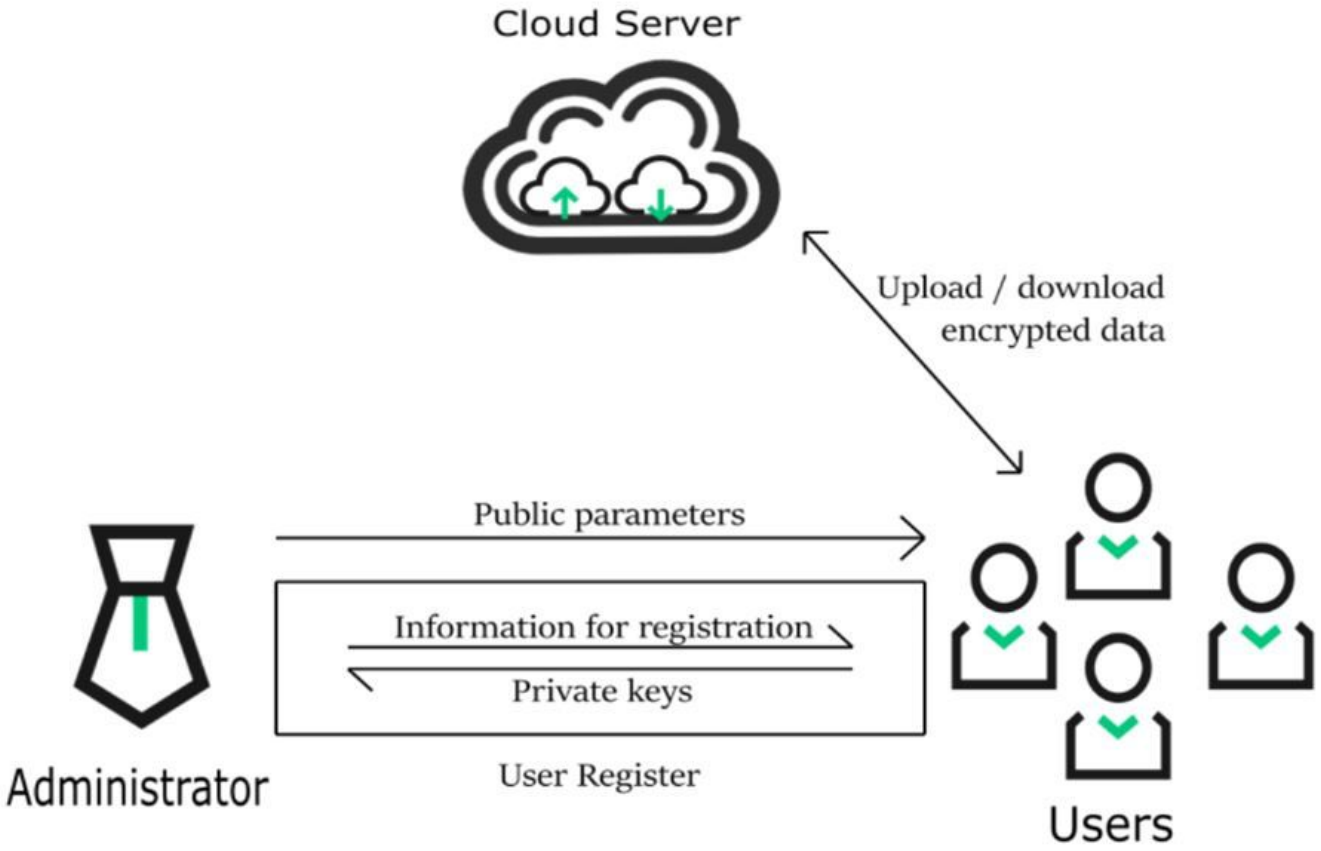


Figure 1

System Model.

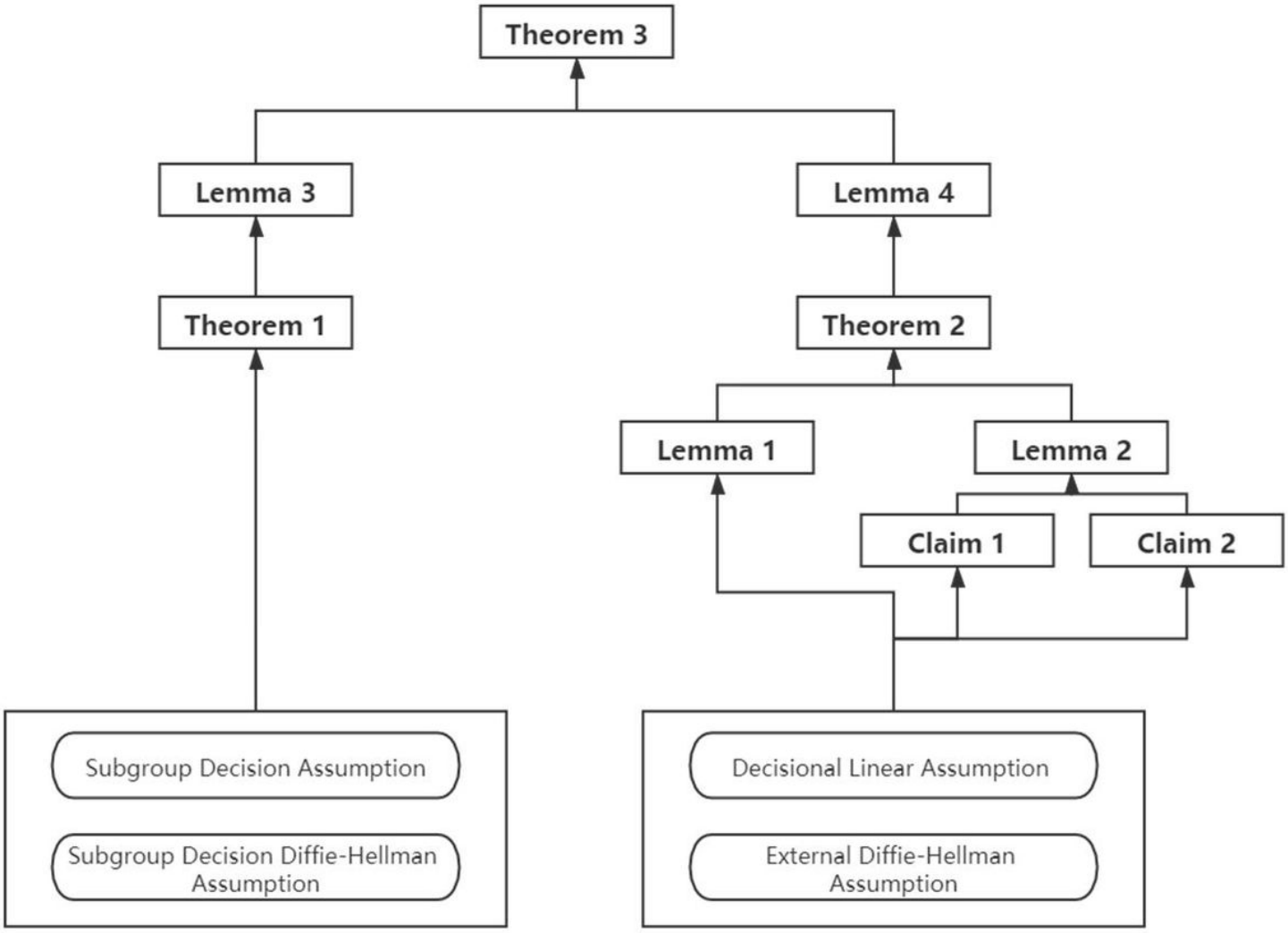


Figure 2

Sketch of security proof.

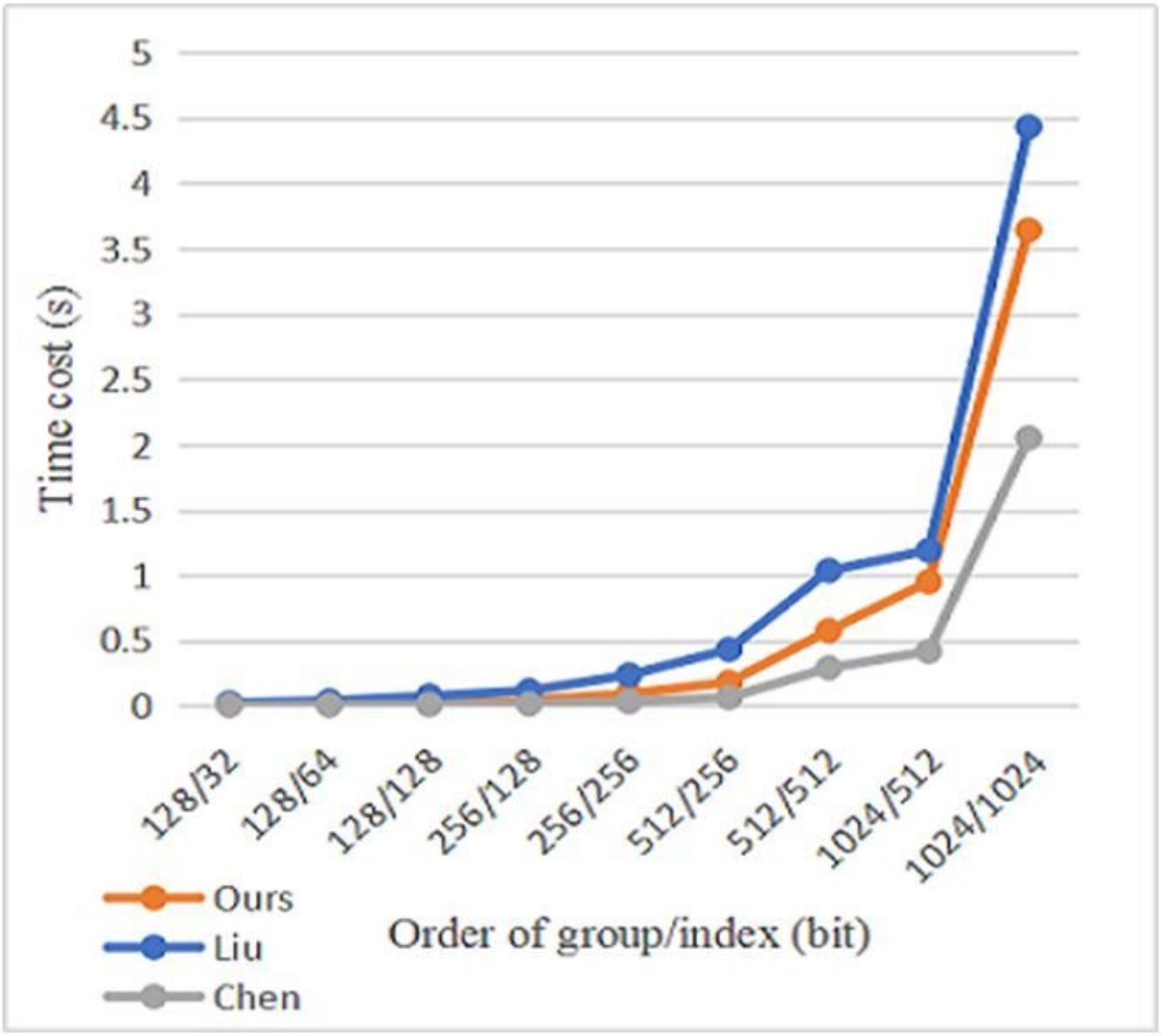


Figure 3

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.

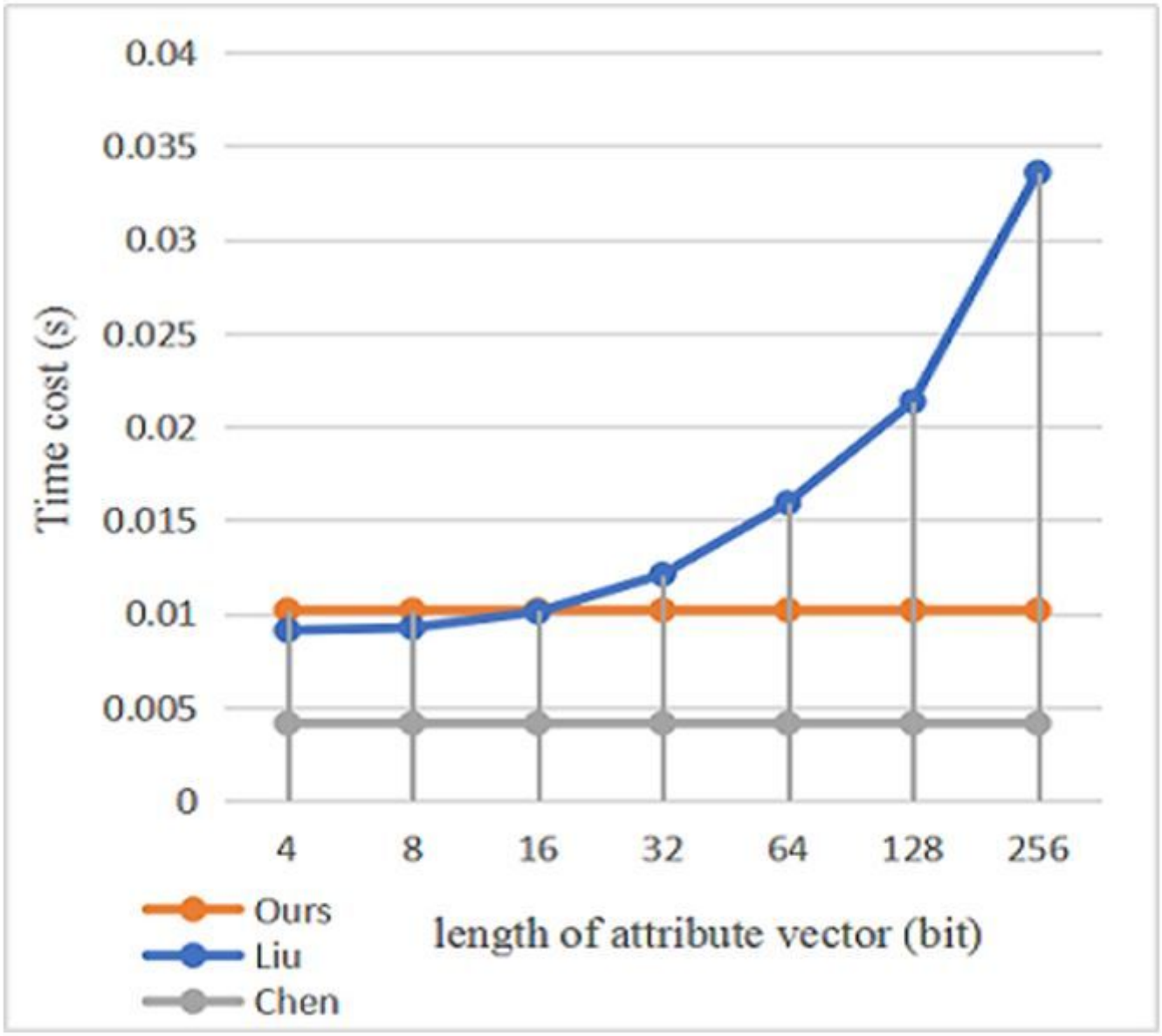


Figure 4

On the basis of group/index = 128/32, the time cost of different matrix sizes in the initialization phase.

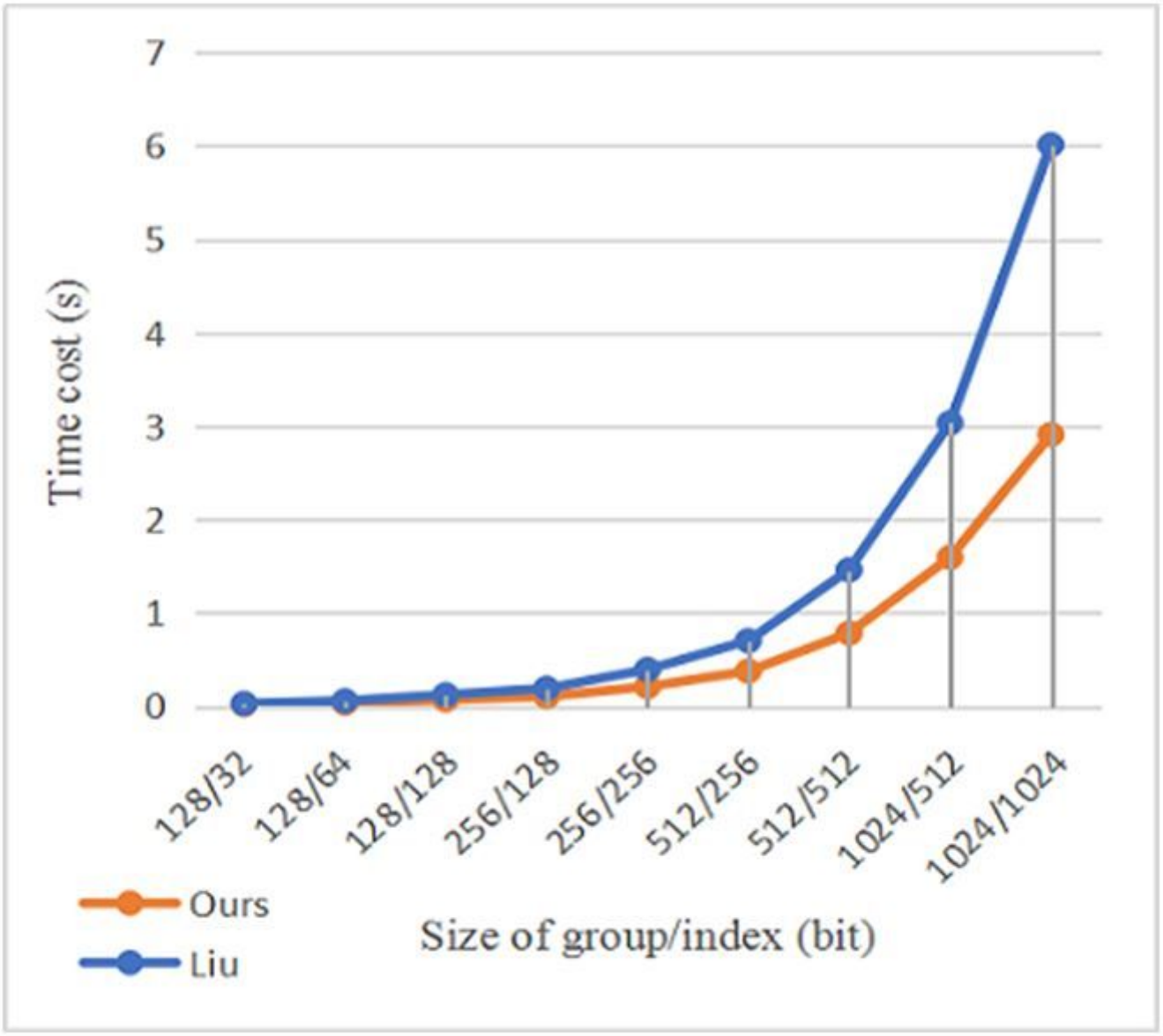


Figure 5

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.

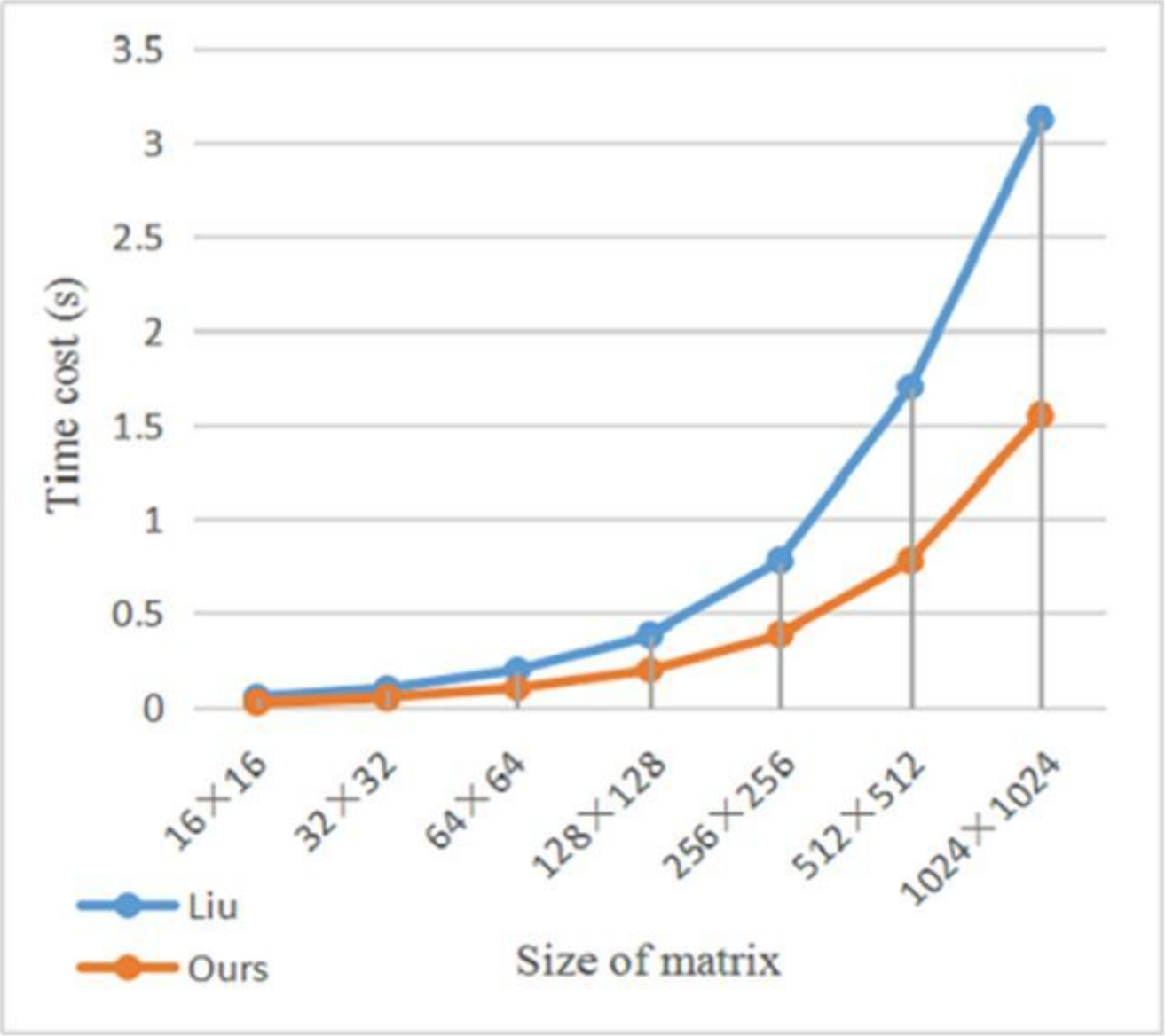


Figure 6

On the basis of group/index = 128/32, the time cost of generating additional ciphertext parts for different matrix sizes.