

12-2016

Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment

Lixian LIU
Jinan University - China


Junzuo LAI
Jinan University - China

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Yingjiu LI
Singapore Management University, yjli@smu.edu.sg

DOI: <https://doi.org/10.1002/sec.1663>

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the [Information Security Commons](#), and the [Medicine and Health Sciences Commons](#)

Citation

LIU, Lixian; LAI, Junzuo; DENG, Robert H.; and LI, Yingjiu. Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment. (2016). *Security and Communication Networks*. 9, (18), 4897-4913. Research Collection School Of Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/3592

This Journal Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

RESEARCH ARTICLE

Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment

Lixian Liu¹, Junzuo Lai^{1*}, Robert H. Deng² and Yingjiu Li²¹ Department of Computer Science, Jinan University, Guangzhou 510632, China² School of Information Systems, Singapore Management University, 178902, Singapore

ABSTRACT

With the development of cloud computing, more and more sensitive data are uploaded to cloud by companies or individuals, which brings forth new challenges for outsourced data security and privacy. Ciphertext-policy attribute-based encryption (CP-ABE) provides fine-grained access control of encrypted data in the cloud; in a CP-ABE scheme, an access structure, also referred to as ciphertext-policy, is sent along with a ciphertext explicitly, and anyone who obtains a ciphertext can know the access structure associated with the ciphertext. In certain applications, access structures contain very sensitive information and must be protected from everyone except the users whose private key attributes satisfy the access structures. In this paper, we propose a new model for CP-ABE with partially hidden access structure (See Figure 2). In our model, each attribute consists of two parts: an attribute name and its value; if the private key attributes of a user do not satisfy the access structure associated with a ciphertext, the specific attribute values of the access structure are hidden, while other information about the access structure is public. Based on the CP-ABE scheme proposed by Lewko and Waters [1] recently, we then present a concrete construction of CP-ABE with partially hidden access structure and prove that it is fully secure in the standard model. In addition, we discuss how our new model can be employed to construct a privacy-preserving electronic medical record system in the cloud environment. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS

cloud computing; ciphertext-policy; hidden access structure; privacy-preserving electronic medical record

*Correspondence

Junzuo Lai, Department of Computer Science, Jinan University, Huangpu Avenue West 601, Tianhe District, Guangzhou 510632, China.

E-mail: laijunzuo@gmail.com

1. INTRODUCTION

Cloud storage services enable users to upload and store their data remotely in the cloud environment because of the great potential of providing various services to the society at significantly reduced cost. Many distributed applications require complex access control mechanisms where a user is able to access sensitive data in the cloud only if the user possesses a certain set of credentials or attributes. A healthcare information system is required to restrict access of medical records to eligible doctors or researchers. A customer relation management system may allow access of customer data by marketing and sales executives of a company only. In these systems, access control of data is either required by legislation (e.g., Health Insurance Portability and Accountability Act) or company regulations.

Traditionally, access controls are enforced by employing trusted servers to store the data and mediate access control. However, services are increasingly storing data across many servers shared with other data owners. Because software systems are not guaranteed to be bug-free, and the hardware platforms are not under the direct control of the data owners in such distributed systems, security risks are abundant, which may allow access of sensitive information by unauthorized users, other applications, and other data owners. To mitigate users' concern about their data, a common solution is to store sensitive data in encrypted form so that it will remain private even if a data server is not trusted or compromised. The encrypted data, however, must be amenable to sharing and access control. Sahai

and Waters [2] addressed this problem by introducing the notion of attribute-based encryption (ABE). ABE enables public key based one-to-many encryption and is envisioned as a promising cryptographic primitive for realizing scalable and fine-grained access control to encrypted data. There are two kinds of ABE schemes [3], key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) schemes. In this paper, our concern is on the latter.

In a CP-ABE scheme [4], every ciphertext is associated with an access structure or access formula on attributes, and every user's secret key is associated with a set of attributes. A user is able to decrypt a ciphertext only if the set of attributes associated with the user's private key satisfies the access structure associated with the ciphertext. CP-ABE is similar to the traditional access control model where data is protected with access structures, and users with credentials satisfying the structures are allowed access to the data. However, in contrast to the traditional access control model where data is stored in cleartext and access control is enforced by trusted servers, CP-ABE stores encrypted data on untrusted servers, and access control is performed via matching of a user's privacy key attributes to access structure of a ciphertext. In the standard CP-ABE schemes [4–7], a cleartext access structure is attached to a ciphertext; therefore, anyone who obtains the ciphertext is able to know its corresponding access structure. Unfortunately, cleartext access structures may leak extremely sensitive information about the encrypted data in certain applications (such as electronic medical record (EMR) systems), as we will demonstrate in Section 2.

To hide access structures in CP-ABE, one can construct CP-ABE with hidden access structure from an attribute-hiding inner-product predicate encryption (IPE) scheme [8]. Predicate Encryption (PE) was proposed by Katz, Sahai and Waters [8] as a generalized fine-grained notion of encryption that covers CP-ABE. In a PE scheme, secret keys that correspond to predicates and ciphertexts are associated with sets of attributes; a secret key SK_f corresponding to a predicate f can be used to decrypt a ciphertext associated with an attribute set I if and only if $f(I) = 1$. Katz, Sahai, and Waters [8] also introduced the idea of *attribute-hiding*, a security notion for PE that is stronger than the basic security requirement of *payload-hiding*. Roughly speaking, attribute-hiding requires that a ciphertext conceal the associated attributes as well as the plaintext, while payload-hiding only requires that a ciphertext conceal the plaintext. The special case of inner-product predicates is obtained by having each attribute correspond to a vector \vec{x} , and each predicate $f_{\vec{v}}$ corresponds to a vector \vec{v} , where $f_{\vec{v}}(\vec{x}) = 1$ if $\vec{x} \cdot \vec{v} = 0$. ($\vec{x} \cdot \vec{v}$ denotes the standard inner-product.)

As mentioned in [6], in order to use inner-product predicates for CP-ABE, access structures must be written in conjunctive normal form or disjunctive normal form, which can cause a *superpolynomial* blowup in size for arbitrary access structures. Because it is extremely inefficient to implement CP-ABE schemes with fully hidden access structure derived from attribute-hiding IPE, we investigate

how to trade-off fully hidden access structure for the efficiency of CP-ABE.

1.1. Our Contributions

In many applications, specific attribute values carry much more sensitive information than the generic attribute names (see Section 2 for a detailed discussion). This observation motivates us to consider a new model of CP-ABE with partially hidden access structure. In this model, each attribute consists of two parts: an attribute name and its value; if the set of attributes associated with a user's private key does not satisfy the access structure associated with a ciphertext, attribute values in the access structure are hidden from the user, while other information, such as attribute names, about the access structure is public.

In the preliminary version of this paper [9], based on the CP-ABE scheme proposed by Lewko *et al.* [6], we present a concrete construction of CP-ABE with partially hidden access structure, which is proven fully secure in the standard model using the dual system encryption methodology in [10]. However, similar to the scheme in [6], our scheme in [9] has the restriction that each attribute can only be used once in an access formula, which is called one-use CP-ABE. This can be extended to a system which allows reuse of attributes by setting a fixed bound N on the maximum number of times an attribute may be used and having separate parameters for each use. This expands the size of the public parameters as well as the size of secret keys by a factor of N and hence incurs a considerable loss in efficiency.

Recently, Lewko and Waters [1] developed a new methodology for utilizing the prior techniques to prove selective security for functional encryption systems as a direct ingredient in devising proofs of full security. Based on their work, we propose a new construction of CP-ABE with partially hidden access structure, which eliminates the aforementioned efficiency loss and allows unrestricted use of attributes while still achieving full security in the standard model. Note that in a CP-ABE scheme, if the access structure associated with a ciphertext is fully hidden, a user is not able to know which attribute set satisfies the access structure, and this makes decryption difficult. In the proposed CP-ABE with partially hidden access structure, we avoid the problem by adding some redundant components to a ciphertext, so that if the private key attributes of a user indeed satisfy the access structure associated with the ciphertext, the user knows which attribute set to use in decrypting the ciphertext. Our scheme handles any access structure that can be expressed as a Linear Secret-Sharing Scheme (LSSS), and its ciphertext size scales *linearly* with the complexity of the access structure.

1.2. Related Work

In this section, we summarize the major related work in the areas of ABE, KP-ABE, PE, CP-ABE, CP-ABE with partially hidden access structure, and dual system encryption.

Attribute-Based Encryption. The notion of ABE was first introduced by Sahai and Waters as an application of their fuzzy identity-based encryption (IBE) scheme [2], where both ciphertexts and secret keys are associated with sets of attributes. The decryption of a ciphertext is enabled if and only if the attribute set for the ciphertext and the attribute set for the secret key overlap by at least a fixed threshold value d .

KP-ABE. Goyal *et al.* [3] formulated two complementary forms of ABE: KP-ABE and CP-ABE. In a CP-ABE scheme, decryption keys are associated with sets of attributes, and ciphertexts are associated with access structures. In a KP-ABE scheme, the situation is reversed: decryption keys are associated with access structures while ciphertexts are associated with sets of attributes. There exists a general method to transform KP-ABE to CP-ABE [11]. In terms of the expressive power of access structures, Goyal *et al.* [3] presented the first KP-ABE supporting monotonic access structures. To enable more flexible access control structures, Ostrovsky *et al.* [12] presented a KP-ABE system that supports the expression of non-monotone formulas in key policies. The problem of building KP-ABE systems with multiple authorities was investigated in [13–15]. Recently, Lewko and Waters [16] proposed a KP-ABE scheme which is ‘unbounded’ in the sense that the public parameters do not impose additional limitations on the functionality of the scheme.

Predicate Encryption. We briefly discuss the work on PE because CP-ABE can be derived from inner-product PE. The notion of PE was introduced by Katz *et al.* [8]. They also proposed the first inner-product PE. Shi and Waters [17] presented a delegation mechanism for a class of PE, in which the admissible predicates of the system are more limited than inner-product predicates. Okamoto and Takashima [18] presented a (hierarchical) delegation mechanism for an inner-product PE scheme. Shen *et al.* [19] introduced a new security notion of PE called predicate privacy and proposed a symmetric-key inner-product PE, which achieves both plaintext privacy and predicate privacy. These schemes were only proven selectively secure. Lewko *et al.* [6] proposed the first fully secure inner-product PE. Okamoto and Takashima [20] presented a fully secure PE for a wide class of admissible predicates, which are specified by non-monotone access structures combined with inner-product predicates.

Ciphertext-policy Attribute-based Encryption. The first CP-ABE construction was proposed by Bethencourt *et al.* [4] and was proven secure under the generic group model. Later, Cheung and Newport [5] proposed a CP-ABE scheme that is secure under the standard model; however, access structures in this scheme are restricted to AND of different attributes. Recently, secure and expressive CP-ABE schemes [6,7] were proposed. CP-ABE schemes with multiple authorities were also studied in [21,22].

Ciphertext-policy Attribute-based Encryption with Partially Hidden Access Structure. Nishide *et al.* [23] considered CP-ABE schemes which hide encryptor-specified access structures associated with ciphertexts. The admissible access structures in [23] can be expressed as AND gates on multi-valued attributes with wildcards, and the CP-ABE scheme in [23] can be considered as a special case of CP-ABE with partially hidden access structure. Li *et al.* [24] followed their work and studied the problem of user accountability. All these schemes are proven to be selectively secure only, which is a weak security model analogous to the selective-ID model [25,26] in IBE schemes. Recently, Lai *et al.* [27] proposed a fully secure (cf. selectively secure) CP-ABE scheme with partially hidden access structure; however, their scheme only supports restricted access structures as in [23,24]. Moving one step forward, we proposed a fully secure CP-ABE scheme with partially hidden access structure expressed as LSSS, which is more flexible and expressive than previous works [23,24,27].

Dual System Encryption. The dual system encryption methodology, introduced by Waters in [10], will be used in the security proofs of our construction. This methodology has been leveraged to obtain constructions of fully secure (H)IBE from simple assumptions [10], fully secure (H)IBE with short ciphertexts [28], fully secure (H)IBE and ABE with leakage resilience [29], fully secure ABE, and inner-product PE [6,20].

1.3. Organization

The rest of the paper is organized as follows. In Section 2, we show how CP-ABE with partially hidden access structure can be used to construct a privacy-preserving EMR system. In Section 3, we review some standard notations and cryptographic definitions. In Section 4, we describe the security model for CP-ABE with partially hidden access structure and propose a concrete construction. We state our conclusion in Section 5.

2. PRIVACY-PRESERVING ELECTRONIC MEDICAL RECORD SYSTEM

An EMR is a collection of patients’ health related information to allow efficient, consistent, and universal sharing of health information. Because of the sensitivity of health related information, providing secure storage and flexible access to EMR is the main challenge in today’s EMR systems. In health care, it must meet the requirements of Health Insurance Portability and Accountability Act (HIPAA) for any use or disclosure of protected healthcare information. On the other hand, with the emergence of cloud computing, it is attractive for EMR service providers to shift their EMR applications and storage into the cloud, in order to enjoy the elastic resources and reduce the

operational cost. However, a cloud environment introduces an even greater risk to security and privacy of sensitive data. Data stored in cloud may reside on computers that are located in dispersed geographic locations and can be seen by many in transit and in their stored form.

Recently, the use of attribute-based cryptography to provide secure cloud storage for EMRs was considered in [30–33]. To enable fine-grained and scalable access control to encrypted data, the solutions in [30–33] leveraged the standard CP-ABE schemes to encrypt data. However, the standard CP-ABE schemes do not hide access structures associated with the ciphertexts, and this property is not appropriate for protecting confidential data because the cloud service provider may infer sensitive information from the access structures about the encrypted data as well as about the users who are granted access to the data. For example, the nature of a patient’s health problem is pretty clear if a cardiologist or a psychiatrist has access to his or her record. The treating medical staff may also have an interest in hiding the access control structures, for example, to avoid being approached by the press when treating public figures.

Figure 1 depicts the system architecture of a cloud-based privacy-preserving EMR system. Suppose that a healthcare provider intends to submit an EMR to the cloud and specifies that it can only be accessed by a cardiologist in University Hospital or by the patient with social security

number 123-45-6789. The healthcare provider encrypts the EMR using a CP-ABE scheme in order to keep it confidential from the cloud service provider and other unauthorized parties. If the healthcare provider uses a standard CP-ABE scheme for encryption, everyone including the cloud service provider is able to know the access structure associated with the encrypted EMR, and can infer that someone with social security number 123-45-6789 suffers from a heart problem. This is clearly not acceptable and shows the necessity of hiding the access structures from prying eyes in a privacy-preserving EMR system.

We observe that, in the access structure shown in Figure 1, ‘Cardiologist’ and ‘123-45-6789’ leak more sensitive information than ‘Occupation’ and ‘SS#’, respectively. In other words, specific attribute values carry much more sensitive information than the generic attribute names. This observation motivates the notion and design of CP-ABE with partially hidden access structure, and we believe that this new notion is more appropriate to use in designing privacy-preserving EMR systems than the standard CP-ABE schemes used in [30–33]. In the aforementioned example, if the healthcare provider uses a CP-ABE scheme with partially hidden access structure to encrypt EMRs, anyone obtaining the ciphertexts only knows the following information about the access structure:

SS# : * OR (Affiliation : * AND Occupation : *),

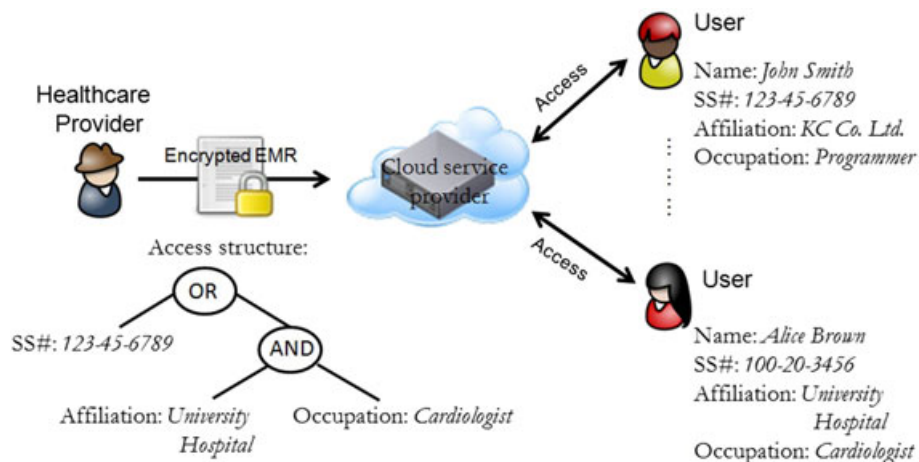


Figure 1. Architecture of a cloud-based privacy-preserving electronic medical record (EMR) system.

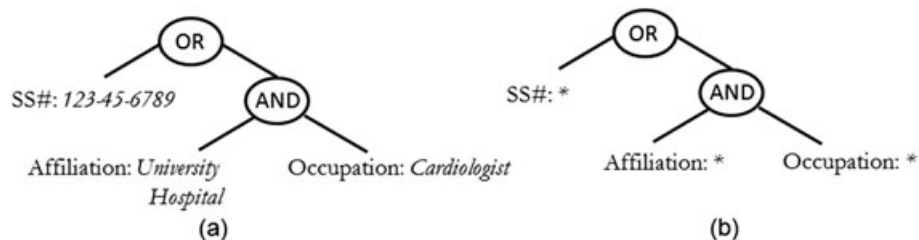


Figure 2. (a) An access structure and (b) the corresponding partially hidden access structure.

while the sensitive attribute values, ‘123-45-6789’, ‘University Hospital’, and ‘Cardiologist’, are hidden from the public. Figure 2 shows graphically this example of partially hidden access structure. We will formally introduce the notion of CP-ABE with partially hidden access structure in Section 4.

3. NOTATIONS AND ASSUMPTIONS

If S is a set, then $s \xleftarrow{\$} S$ denotes the operation of picking an element s uniformly at random from S . If A is a finite set, $|A|$ denotes the cardinality of the set A . Let \mathbb{N} denote the set of natural numbers. If $\lambda \in \mathbb{N}$ then 1^λ denotes the string of λ ones. Let $z \leftarrow \mathbf{A}(x, y, \dots)$ denote the operation of running an algorithm \mathbf{A} with inputs (x, y, \dots) and output z . A function $f(\lambda)$ is negligible, if for every $c > 0$ there exists a λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$.

3.1. Access structures

Definition 1 (Access structure [34]). *Let $\{P_1, \dots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, \dots, P_n\}}$ is monotone if $\forall B, C : \text{if } B \in \mathbb{A} \text{ and } B \subseteq C, \text{ then } C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) \mathbb{A} of non-empty subsets of $\{P_1, \dots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, \dots, P_n\}} \setminus \{\emptyset\}$. The sets in \mathbb{A} are called authorized sets, and the sets not in \mathbb{A} are called unauthorized sets.*

In our context, attributes play the role of parties, and we restrict our attention to monotone access structures. It is possible to (inefficiently) realize general access structures using our techniques by treating the negation of an attribute as a separate attribute.

3.2. Linear secret-sharing schemes

Our construction will employ LSSS. We use the definition adapted from [34]:

Definition 2 (Linear secret-sharing schemes). *A secret sharing scheme Π over a set of parties \mathcal{P} is called linear (over \mathbb{Z}_p) if*

- (1) *The shares for each party form a vector over \mathbb{Z}_p .*
- (2) *There exists a matrix \mathbf{A} with ℓ rows and n columns called the share-generating matrix for Π . For all $i = 1, \dots, \ell$, the i^{th} row of \mathbf{A} is labeled by a party $\rho(i)$ (ρ is a function from $\{1, \dots, \ell\}$ to \mathcal{P}). When we consider the column vector $v = (s, r_2, \dots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \dots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $\mathbf{A}v$ is the vector of ℓ shares of the secret s according to Π . The share $(\mathbf{A}v)_i$ belongs to party $\rho(i)$.*

It is shown in [34] that every LSSS according to the aforementioned definition also enjoys the linear recon-

struction property, defined as follows. Suppose that Π is an LSSS for the access structure \mathbb{A} . Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, \dots, \ell\}$ be defined as $I = \{i | \rho(i) \in S\}$. Then, there exists constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π , then $\sum_{i \in I} \omega_i \lambda_i = s$. Let A_i denotes the i^{th} row of \mathbf{A} , we have $\sum_{i \in I} \omega_i A_i = (1, 0, \dots, 0)$. These constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generation matrix \mathbf{A} [34]. Note that, for unauthorized sets, no such constants $\{\omega_i\}$ exist.

Boolean Formulas Access structures might also be described in terms of monotonic boolean formulas. Using standard techniques [34] one can convert any monotonic boolean formula into an LSSS representation. We can represent the boolean formula as an access tree. An access tree of ℓ nodes will result in an LSSS matrix of ℓ rows. We refer the reader to the appendix of [22] for a discussion on how to perform this conversion.

3.3. Ciphertext-policy attribute-based encryption

A CP-ABE scheme consists of the following four algorithms:

Setup($1^\lambda, U$) takes as input a security parameter λ and the attribute universe description U . It outputs the public parameters PK and a master secret key MSK .

KeyGen(PK, MSK, S) takes as input the public parameters PK , the master secret key MSK , and a set of attributes S . It outputs a secret key SK_S .

Encrypt(PK, M, \mathbb{A}) takes as input the public parameters PK , a message M , and an access structure \mathbb{A} . It outputs a ciphertext C .

Decrypt($\text{PK}, \text{SK}_S, C$) takes as input the public parameters PK , a secret key SK_S and a ciphertext C . It outputs a message M .

Let $(\text{PK}, \text{MSK}) \leftarrow \text{Setup}(1^\lambda, U)$, $\text{SK}_S \leftarrow \text{KeyGen}(\text{PK}, \text{MSK}, S)$, and $C \leftarrow \text{Encrypt}(\text{PK}, M, \mathbb{A})$. For correctness, we require the following to hold:

- (1) If the set S of attributes satisfies the access structure \mathbb{A} , then $M \leftarrow \text{Decrypt}(\text{PK}, \text{SK}_S, C)$;
- (2) Otherwise, with overwhelming probability, $\text{Decrypt}(\text{PK}, \text{SK}_S, C)$ outputs a random message.

3.4. Composite order bilinear groups

We will construct our scheme in composite order bilinear groups whose order is the product of four distinct primes. Composite order bilinear groups were first introduced in [35].

Let \mathcal{G} be an algorithm that takes as input a security parameter 1^λ and outputs a tuple $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$, where p_1, p_2, p_3 , and p_4 are distinct primes, \mathbb{G} and \mathbb{G}_T are cyclic groups of order $N = p_1 p_2 p_3 p_4$, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a map such that

- (1) (Bilinear) $\forall g, h \in \mathbb{G}, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$;
 (2) (Non-degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in \mathbb{G}_T .

We further require that multiplication in \mathbb{G} and \mathbb{G}_T , as well as the bilinear map e , are computable in time polynomial in λ . We use $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$, and \mathbb{G}_{p_4} to denote the subgroups of \mathbb{G} having order p_1, p_2, p_3 , and p_4 , respectively. Observe that $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$. Note also that if $g_1 \in \mathbb{G}_{p_1}$ and $g_2 \in \mathbb{G}_{p_2}$ then $e(g_1, g_2) = 1$. A similar rule holds whenever e is applied to elements in distinct subgroups.

We now state the complexity assumptions we use. Assumptions 1 and 2 are some instantiations of the general subgroup decision assumption defined in [36]. Assumptions 3 and 4 are the three part Diffie–Hellman assumption and the source group q -parallel Bilinear Diffie–Hellman Exponent (BDHE) assumption used in [1], respectively, and we use them in the group whose order is a product of four primes. Assumptions 5 and 6 are essentially the same as Assumption 1 in [1].

Assumption 1. Let \mathcal{G} be as mentioned previously. We define the following distribution:

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), N = p_1 p_2 p_3 p_4$$

$$g \xleftarrow{\$} \mathbb{G}_{p_1}, X_3 \xleftarrow{\$} \mathbb{G}_{p_3}, X_4 \xleftarrow{\$} \mathbb{G}_{p_4}$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_3, X_4)$$

$$T_1 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}, T_2 \xleftarrow{\$} \mathbb{G}_{p_1}$$

The advantage of an algorithm \mathcal{A} in breaking Assumption 1 is defined as

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

Definition 3. We say \mathcal{G} satisfies Assumption 1 if for any polynomial time algorithm \mathcal{A} , $\text{Adv}_{\mathcal{A}}^1$ is negligible.

Assumption 2. Let \mathcal{G} be as mentioned previously. We define the following distribution:

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), N = p_1 p_2 p_3 p_4$$

$$g, X_1 \xleftarrow{\$} \mathbb{G}_{p_1}, X_2, Y_2 \xleftarrow{\$} \mathbb{G}_{p_2}$$

$$X_3, Y_3 \xleftarrow{\$} \mathbb{G}_{p_3}, X_4 \xleftarrow{\$} \mathbb{G}_{p_4}$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_1 X_2, Y_2 Y_3, X_3, X_4)$$

$$T_1 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}, T_2 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$$

The advantage of an algorithm \mathcal{A} in breaking Assumption 2 is defined as

$$\text{Adv}_{\mathcal{A}}^2 = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 4. We say \mathcal{G} satisfies Assumption 2 if for any polynomial time algorithm \mathcal{A} , $\text{Adv}_{\mathcal{A}}^2$ is negligible.

Assumption 3. Let \mathcal{G} be as mentioned previously. We define the following distribution:

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), N = p_1 p_2 p_3 p_4$$

$$g \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, X_3 \xleftarrow{\$} \mathbb{G}_{p_3}, X_4 \xleftarrow{\$} \mathbb{G}_{p_4}$$

$$x, y, z \xleftarrow{\$} \mathbb{Z}_N,$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, g_2^x, g_2^y, g_2^z, X_3, X_4)$$

$$T_1 = g_2^{xyz}, T_2 \xleftarrow{\$} \mathbb{G}_{p_2}$$

The advantage of an algorithm \mathcal{A} in breaking Assumption 3 is defined as

$$\text{Adv}_{\mathcal{A}}^3 = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 5. We say \mathcal{G} satisfies Assumption 3 if for any polynomial time algorithm \mathcal{A} , $\text{Adv}_{\mathcal{A}}^3$ is negligible.

Assumption 4. Let \mathcal{G} be as mentioned previously. We define the following distribution:

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), N = p_1 p_2 p_3 p_4$$

$$g \xleftarrow{\$} \mathbb{G}_{p_1}, g_2 \xleftarrow{\$} \mathbb{G}_{p_2}, X_3 \xleftarrow{\$} \mathbb{G}_{p_3}, X_4 \xleftarrow{\$} \mathbb{G}_{p_4}$$

$$x, y, f, z_1, \dots, z_q \xleftarrow{\$} \mathbb{Z}_N$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_3, X_4$$

$$g_2, g_2^f, g_2^{xf}, g_2^{y^i} \forall i \in [2q] \setminus \{q+1\}$$

$$g_2^{y^j/z_j} \forall i \in [2q] \setminus \{q+1\}, j \in [q]$$

$$g_2^{x^j/z_j} \forall j \in [q]$$

$$g_2^{x^j y^i z_j / z_j} \forall i \in [q], j, j' \in [q] \text{ s.t. } j \neq j')$$

$$T_1 = g_2^{xyz}, T_2 \xleftarrow{\$} \mathbb{G}_{p_2}$$

The advantage of an algorithm \mathcal{A} in breaking Assumption 4 is defined as

$$\text{Adv}_{\mathcal{A}}^4 = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 6. We say \mathcal{G} satisfies Assumption 4 if for any polynomial time algorithm \mathcal{A} , $\text{Adv}_{\mathcal{A}}^4$ is negligible.

Assumption 5. Let \mathcal{G} be as mentioned previously. We define the following distribution:

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), N = p_1 p_2 p_3 p_4$$

$$\alpha, s \xleftarrow{\$} \mathbb{Z}_N, g \xleftarrow{\$} \mathbb{G}_{p_1}$$

$$g_2, X_2, Y_2 \xleftarrow{\$} \mathbb{G}_{p_2}, X_3 \xleftarrow{\$} \mathbb{G}_{p_3}, X_4 \xleftarrow{\$} \mathbb{G}_{p_4}$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, g_2^\alpha X_2, g_2^s Y_2, X_3, X_4)$$

$$T_1 = e(g, g)^{\alpha^s}, T_2 \xleftarrow{\$} \mathbb{G}_T$$

The advantage of an algorithm \mathcal{A} in breaking Assumption 5 is defined as

$$\text{Adv}_{\mathcal{A}}^5 = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 7. We say \mathcal{G} satisfies Assumption 5 if for any polynomial time algorithm \mathcal{A} , $\text{Adv}_{\mathcal{A}}^5$ is negligible.

Assumption 6. Let \mathcal{G} be as mentioned previously. We define the following distribution:

$$(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^\lambda), N = p_1 p_2 p_3 p_4$$

$$a, s \xleftarrow{\$} \mathbb{Z}_N, g \xleftarrow{\$} \mathbb{G}_{p_1}, g_2, X_2, Y_2, D_2 \xleftarrow{\$} \mathbb{G}_{p_2}$$

$$X_3 \xleftarrow{\$} \mathbb{G}_{p_3}, X_4, Z', Y_4, D_4 \xleftarrow{\$} \mathbb{G}_{p_4}$$

$$D = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, g^a X_2, g^a Z', g^s Y_2 Y_4, X_3, X_4)$$

$$T_1 = g^{as} D_2 D_4, T_2 \xleftarrow{\$} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$$

The advantage of an algorithm \mathcal{A} in breaking Assumption 6 is defined as

$$\text{Adv}_{\mathcal{A}}^6 = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|.$$

Definition 8. We say \mathcal{G} satisfies Assumption 6 if for any polynomial time algorithm \mathcal{A} , $\text{Adv}_{\mathcal{A}}^6$ is negligible.

4. CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION WITH PARTIALLY HIDDEN ACCESS STRUCTURE

In this section, we first describe the security model for CP-ABE with partially hidden access structure. Then, based on the CP-ABE scheme given by Lewko and Waters [1], we propose a new CP-ABE scheme, which satisfies the security definition of CP-ABE with partially hidden access structure.

Our construction supports arbitrary monotone access formulas or structures. In the following, we will use the terms access formula and access structure interchangeably. As in [1], we express access structures by an LSSS matrix \mathbf{A} over the attributes in the system but with a significant difference. In our construction, each attribute includes two parts: attribute name and its value. Without loss of generality, we assume that there are n categories of attributes and every user has n attributes with each attribute belonging to a different category. For notational purposes, let i denote the attribute name of the i th category attribute. A user's attribute set \mathcal{S} is parsed as (s_1, \dots, s_n) , where $s_i \in \mathbb{Z}_N$ is the value of attribute i . We express an access structure by $(\mathbf{A}, \rho, \mathcal{T})$, where \mathbf{A} is $\ell \times m$ share-generating matrix, ρ is a map from each row of \mathbf{A} to an attribute name (i.e., ρ is

a function from $\{1, \dots, \ell\}$ to $\{1, \dots, n\}$), \mathcal{T} can be parsed as $(t_{\rho(1)}, \dots, t_{\rho(\ell)})$, and $t_{\rho(i)}$ is the value of attribute $\rho(i)$ specified by the access formula.

Using our notations, a user's attribute set $\mathcal{S} = (s_1, \dots, s_n)$ satisfies an access formula $(\mathbf{A}, \rho, \mathcal{T})$ if and only if there exists $\mathcal{I} \subseteq \{1, \dots, \ell\}$ and constants $\{\omega_i\}_{i \in \mathcal{I}}$ such that

$$\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \dots, 0) \text{ and } s_{\rho(i)} = t_{\rho(i)} \text{ for } \forall i \in \mathcal{I}$$

where A_i denotes the i th row of \mathbf{A} . We also say that $\mathcal{I} \subseteq \{1, \dots, \ell\}$ satisfies (\mathbf{A}, ρ) if there exist constants $\{\omega_i\}_{i \in \mathcal{I}}$ such that $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \dots, 0)$. We define $\mathbf{I}_{\mathbf{A}, \rho}$ as the set of minimum subsets of $\{1, \dots, \ell\}$ that satisfies (\mathbf{A}, ρ) . By 'minimum', we mean the subset cannot become smaller while still satisfying (\mathbf{A}, ρ) .

Note that, in our construction to be presented in the next sections, the specific attribute values (i.e., \mathcal{T}) of an access structure $(\mathbf{A}, \rho, \mathcal{T})$ is hidden, while other information about the access structure (i.e., (\mathbf{A}, ρ)) is sent along with the ciphertext explicitly).

4.1. Security model of ciphertext-policy attribute-based encryption with partially hidden access structure

We now give the security model of CP-ABE with partially hidden access structure, described as a security game between a challenger and an adversary \mathcal{A} . The game proceeds as follows:

Setup The challenger runs $\text{Setup}(1^\lambda, U)$ to obtain the public parameters PK and a master secret key MSK . It gives the public parameters PK to the adversary \mathcal{A} and keeps MSK to itself.

Query phase 1 The adversary \mathcal{A} adaptively queries the challenger for secret keys corresponding to sets of attributes $\mathcal{S}_1, \dots, \mathcal{S}_q$. In response, the challenger runs $\text{SK}_{\mathcal{S}_i} \leftarrow \text{KeyGen}(\text{PK}, \text{MSK}, \mathcal{S}_i)$ and gives the secret key $\text{SK}_{\mathcal{S}_i}$ to \mathcal{A} , for $1 \leq i \leq q$.

Challenge The adversary \mathcal{A} submits two (equal length) messages M_0 and M_1 , and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0)$ and $(\mathbf{A}, \rho, \mathcal{T}_1)$, subject to the restriction that $(\mathbf{A}, \rho, \mathcal{T}_0)$ and $(\mathbf{A}, \rho, \mathcal{T}_1)$ cannot be satisfied by any of the queried attribute sets. The challenger selects a random bit $\beta \in \{0, 1\}$, sets $C = \text{Encrypt}(\text{PK}, M_\beta, (\mathbf{A}, \rho, \mathcal{T}_\beta))$ and sends C to the adversary as its challenge ciphertext. Note that, the LSSS matrix \mathbf{A} and ρ are the same in the two access structures provided by the adversary. In a CP-ABE scheme with partially hidden access structure, one can distinguish the ciphertexts if the associated access structures have different (\mathbf{A}, ρ) , because (\mathbf{A}, ρ) is sent along with the ciphertext explicitly.

Query phase 2 The adversary continues to adaptively query the challenger for secret keys corresponding to sets of attributes with the added restriction that none of them satisfies $(\mathbf{A}, \rho, \mathcal{T}_0)$ and $(\mathbf{A}, \rho, \mathcal{T}_1)$.

Guess The adversary \mathcal{A} outputs its guess $\beta' \in \{0, 1\}$ for β and wins the game if $\beta = \beta'$. The advantage of the adversary in this game is defined as $|\Pr[\beta = \beta'] - \frac{1}{2}|$ where the probability is taken over the random bits used by the challenger and the adversary.

Definition 9. An access structure in a ciphertext-policy attribute-based encryption scheme is partially hidden if all polynomial time adversaries have at most a negligible advantage in this security game.

4.2. Our Proposed Construction

The proposed CP-ABE scheme with partially hidden access structure consists of the following algorithms:

Setup($1^\lambda, U$) The setup algorithm first runs $\mathcal{G}(1^\lambda)$ to obtain $(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)$ with $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$, where \mathbb{G} and \mathbb{G}_T are cyclic groups of order $N = p_1 p_2 p_3 p_4$. Let the attribute universe description $U = \mathbb{Z}_N$. Next, it chooses $g, u, h_1, \dots, h_n \in \mathbb{G}_{p_1}, X_3 \in \mathbb{G}_{p_3}, X_4, Z, Z', Z'', Z_0, Z_1, \dots, Z_n \in \mathbb{G}_{p_4}$ and $\alpha, a, b \in \mathbb{Z}_N$ uniformly at random. The public parameters are published as $\text{PK} = (N, gZ, g^a Z', g^b Z'', e(g, g)^\alpha, V = uZ_0, \{H_i = h_i \cdot Z_i\}_{1 \leq i \leq n}, X_4)$. The master secret key is $\text{MSK} = (g, u, h_1, \dots, h_n, X_3, a, b, \alpha)$.

KeyGen($\text{PK}, \text{MSK}, S = (s_1, \dots, s_n)$) The key generation algorithm chooses $t, \bar{t} \in \mathbb{Z}_N$, and $R, R', R'', R_1, \dots, R_n \in \mathbb{G}_{p_3}$ uniformly at random. The secret key $\text{SK}_S = (S, K, K', K'', \{K_i\}_{1 \leq i \leq n})$ is computed as

$$K = g^\alpha g^{at} g^{b\bar{t}} R, K' = g^{\bar{t}} R'$$

$$K'' = g^t R'', K_i = (u^{s_i} h_i)^t R_i$$

Encrypt($\text{PK}, M \in \mathbb{G}_T, (\mathbf{A}, \rho, \mathcal{T})$) Let \mathbf{A} be an $\ell \times m$ matrix, ρ a map from each row A_j of \mathbf{A} to an attribute name, and $\mathcal{T} = (t_{\rho(1)}, \dots, t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$. The encryption algorithm chooses two random vectors v and $v' \in \mathbb{Z}_N^m$, denoted $v = (s, v_2, \dots, v_m)$ and $v' = (s', v'_2, \dots, v'_m)$. It also chooses $r_j, r'_j \in \mathbb{Z}_N$ and $\tilde{Z}'_1, \tilde{Z}''_1, \tilde{Z}'_2, \tilde{Z}''_2, \tilde{Z}'_{1,j}, \tilde{Z}''_{1,j}, \tilde{Z}'_{2,j}, \tilde{Z}''_{2,j} \in \mathbb{G}_{p_4}$ uniformly at random, for $1 \leq j \leq \ell$. The ciphertext is $C = ((\mathbf{A}, \rho), \tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$, where

$$\tilde{C}_1 = M \cdot e(g, g)^{\alpha s}, C'_1 = (g^b Z'')^s \cdot \tilde{Z}'_1 = g^{bs} Z'_1$$

$$C''_1 = (gZ)^s \cdot \tilde{Z}''_1 = g^s Z''_1$$

$$C_{1,j} = (g^a Z')^{A_j \cdot v} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r_j} \cdot \tilde{Z}'_{1,j} \\ = g^{aA_j \cdot v} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r_j} Z'_{1,j}$$

$$D_{1,j} = (gZ)^{r_j} \cdot \tilde{Z}'_{1,j} = g^{r_j} Z'_{1,j}$$

$$\tilde{C}_2 = e(g, g)^{\alpha s'}, C'_2 = (g^b Z'')^{s'} \cdot \tilde{Z}'_2 = g^{bs'} Z''_2$$

$$C''_2 = (gZ)^{s'} \cdot \tilde{Z}''_2 = g^{s'} Z''_2$$

$$C_{r2,j} = (g^a Z')^{A_j \cdot v'} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r'_j} \cdot \tilde{Z}'_{2,j} \\ = g^{aA_j \cdot v'} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r'_j} Z'_{2,j}$$

$$D_{2,j} = (gZ)^{r'_j} \cdot \tilde{Z}'_{2,j} = g^{r'_j} Z'_{2,j}$$

$$Z'_1 = Z''^s \tilde{Z}'_1, Z''_1 = Z^s \tilde{Z}''_1, Z_{1,j} = (Z')^{A_j \cdot v} \tilde{Z}'_{1,j}, Z'_{1,j} = Z^{r_j} \tilde{Z}'_{1,j}, \\ Z'_2 = Z''^{s'} \tilde{Z}'_2, Z''_2 = Z^{s'} \tilde{Z}''_2, Z_{2,j} = (Z')^{A_j \cdot v'} \tilde{Z}'_{2,j} \text{ and } Z'_{2,j} = Z^{r'_j} \tilde{Z}'_{2,j}$$

Decrypt($\text{PK}, \text{SK}_S, C$) Let $C = ((\mathbf{A}, \rho), \tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$, $\text{SK}_S = (S, K, K', K'', \{K_i\}_{1 \leq i \leq n})$ and $S = (s_1, \dots, s_n)$. The decryption algorithm first calculates $\mathbf{I}_{\mathbf{A}, \rho}$ from (\mathbf{A}, ρ) , where $\mathbf{I}_{\mathbf{A}, \rho}$ denotes the set of minimum subsets of $\{1, \dots, \ell\}$ that satisfies (\mathbf{A}, ρ) . It then checks if there exists an $\mathcal{I} \in \mathbf{I}_{\mathbf{A}, \rho}$ that satisfies

$$\tilde{C}_2 = e(C''_2, K) e(C'_2, K')^{-1} / \left(\prod_{i \in \mathcal{I}} (e(C_{2,i}, K'') \cdot e(D_{2,i}, K_{\rho(i)}))^{\omega_i} \right)$$

where $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \dots, 0)$. If no element in $\mathbf{I}_{\mathbf{A}, \rho}$ satisfies the aforementioned equation, it outputs \perp . Otherwise, it computes

$$e(C''_1, K) e(C'_1, K')^{-1} / \left(\prod_{i \in \mathcal{I}} (e(C_{1,i}, K'') \cdot e(D_{1,i}, K_{\rho(i)}))^{\omega_i} \right) \\ = e(g, g)^{\alpha s} e(g, g)^{\alpha s'} / \left(\prod_{i \in \mathcal{I}} e(g, g)^{aA_i \cdot v \cdot \omega_i} \right) \\ = e(g, g)^{\alpha s}$$

Then M can be recovered as $\tilde{C}_1 / e(g, g)^{\alpha s}$.

In our construction, a ciphertext includes two parts: $(\tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell})$ and $(\tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$. The first part is an encryption of the message M . The second part is redundant and can be viewed as an encryption of the identity element 1. If the private key attributes of a user satisfy the access structure associated with the ciphertext, the redundant second part will help the user decide which his attribute set satisfies the access structure; and if yes, the user then can use his private key to decrypt the first part of the ciphertext and recover the plaintext M .

Efficiency. The size of the public parameters, a user's private key and a ciphertext are $(n+5)|\mathbb{G}| + |\mathbb{G}_T|$, $(n+3)|\mathbb{G}|$, and $(4\ell+4)|\mathbb{G}| + 2|\mathbb{G}_T|$, respectively, where $|\mathbb{G}|$ and $|\mathbb{G}_T|$ are the lengths of the bit-representation of a group element in \mathbb{G} and \mathbb{G}_T respectively. For an access structure $(\mathbf{A}, \rho, \mathcal{T})$, let $\iota_1 = |\mathbf{I}_{\mathbf{A}, \rho}|$, $\mathbf{I}_{\mathbf{A}, \rho} = \{I_1, \dots, I_{\iota_1}\}$, $\iota_2 = |I_1| + \dots + |I_{\iota_1}|$

and $\iota_3 = \max\{\|\Pi_1\|, \dots, \|\Pi_{\ell_1}\|\}$. The computational costs of an encryption under $(PK, M, (\mathbf{A}, \rho, \mathcal{T}))$, and a decryption are $(4\ell + 4)t_{\mathbb{G}_{m_e}} + 2t_{\mathbb{G}_{T_e}}$ and $\leq (2\iota_1 + 2\iota_2 + 2\iota_3 + 2)t_p + (\iota_1 + 1)t_{\mathbb{G}_{T_m_e}}$, respectively, where $t_p, t_{\mathbb{G}_{T_e}}, t_{\mathbb{G}_{m_e}}$ and $t_{\mathbb{G}_{T_m_e}}$ are the computational costs of bilinear map, exponentiation in \mathbb{G}_T , multi exponentiation in \mathbb{G} or \mathbb{G}_T , respectively.

Security. The CP-ABE construction in [1] uses composite order bilinear groups whose order is the product of three distinct primes, while our construction uses groups whose order is the product of four distinct primes. Note that in our construction, the public parameters (except for $e(g, g)^\alpha$) and the ciphertext (except for $\tilde{C}_1 = Me(g, g)^{\alpha s}$ and $\tilde{C}_2 = e(g, g)^{\alpha s'}$) have an element from \mathbb{G}_{p_4} as a factor. This formation allows us to prove that an access structures in our CP-ABE scheme is partially hidden. At the same time however, the formation does not affect decryption operations, because none of the components in a private key has element in \mathbb{G}_{p_4} as a factor. We now state the security theorem of our CP-ABE scheme.

Theorem 1. *If Assumptions 1–6 hold, then the access structure in the proposed CP-ABE is partially hidden.*

Proof. Following the approach by Lewko and Waters [28], we define two additional data structures: *semi-functional* ciphertexts and *semi-functional* keys. These will not be used in the real system, but will be used in our proof. *Semi-functional Ciphertext* Let g_2 denote a generator of the subgroup \mathbb{G}_{p_2} . A semi-functional ciphertext is created as follows. We first use the encryption algorithm to form a normal ciphertext $C' = ((\mathbf{A}, \rho), \tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$. Then, we choose random exponents $a', b', c, c' \in \mathbb{Z}_N$ and two random vectors w and $w' \in \mathbb{Z}_N^m$. We also choose random values $\eta_i \in \mathbb{Z}_N$ associated with attributes, and random values $\gamma_j, \gamma'_j \in \mathbb{Z}_N$ associated with row j of the $\ell \times m$ matrix \mathbf{A} . The semi-functional ciphertext C is set as

$$\left((\mathbf{A}, \rho), \tilde{C}_1, C'_1 \cdot g_2^{b'c}, C''_1 \cdot g_2^c, \left\{ C_{1,j} \cdot g_2^{a' A_j w + \gamma_j \eta_{\rho(j)}}, D_{1,j} \cdot g_2^{-\gamma_j} \right\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2 \cdot g_2^{b'c'}, C''_2 \cdot g_2^{c'}, \left\{ C_{2,j} \cdot g_2^{a' A_j w' + \gamma'_j \eta_{\rho(j)}}, D_{2,j} \cdot g_2^{-\gamma'_j} \right\}_{1 \leq j \leq \ell} \right)$$

Observe that the structure of the elements in \mathbb{G}_{p_2} here is similar to the structure in \mathbb{G}_{p_1} but is unrelated to the public parameters. It should be noted that these values a', b' , and η_i are chosen randomly once and then fixed. These same values will also be involved in semi-functional keys which we will be defined in the next discussions.

Semi-functional Key A semi-functional key will take on one of three forms. To create a semi-functional key, we

first use the key generation algorithm to form a normal key $SK'_S = (S, K, K', K'', \{K_i\}_{1 \leq i \leq n})$. Then, we choose $d, d' \in \mathbb{Z}_N$, and $W \in \mathbb{G}_{p_2}$ uniformly at random. The semi-functional key of type 1 is set as

$$\left(S, K \cdot g_2^{d'+b'd'}, K' \cdot g_2^{d'}, K'' \cdot g_2^d, \{K_i \cdot g_2^{d\eta_i}\}_{1 \leq i \leq n} \right)$$

The semi-functional key of type 2 is set as

$$\left(S, K \cdot W, K' \cdot g_2^{d'}, K'' \cdot g_2^d, \{K_i \cdot g_2^{d\eta_i}\}_{1 \leq i \leq n} \right)$$

The semi-functional key of type 3 is set as

$$(S, K \cdot W, K', K'', \{K_i\}_{1 \leq i \leq n})$$

We will prove the security of our scheme from Assumptions 1–6 using a hybrid argument over a sequence of games. The first game, Game_{real} , is the real security game (the ciphertext and all the keys are normal). In the next game, Game_0 , all of the keys will be normal, but the challenge ciphertext will be semi-functional. We let Q denote the number of key queries made by the attacker. For k from 1 to Q , we define

$\text{Game}_{k,1}$ In this game, the challenge ciphertext is semi-functional, the first $k-1$ keys are semi-functional of type 3, the k th key is semi-functional of type 1, and the remaining keys are normal.

$\text{Game}_{k,2}$ In this game, the challenge ciphertext is semi-functional, the first $k-1$ keys are semi-functional of type 3, the k th key is semi-functional of type 2, and the remaining keys are normal.

$\text{Game}_{k,3}$ In this game, the challenge ciphertext is semi-functional, the first k keys are semi-functional of type 3, and the remaining keys are normal.

For notational purposes, we think of $\text{Game}_{0,3}$ as another way of denoting Game_0 . We note that in $\text{Game}_{Q,3}$, all of the keys are semi-functional of type 3. In the penultimate game, $\text{Game}_{\text{Final}_0}$, all the keys are semi-functional, and the ciphertext is a semi-functional encryption of a random message, independent of the messages M_0 and M_1 provided by the adversary. The final game, $\text{Game}_{\text{Final}_1}$, is the same as $\text{Game}_{\text{Final}_0}$, except that in the challenge ciphertext, $C_{1,x}$ and $C_{2,x}$ are chosen from $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$ at random (thus, the ciphertext is independent of \mathcal{T}_0 and \mathcal{T}_1 provided by the adversary). It is clear that in the final game, no adversary can have advantage greater than 0.

We prove that these games are indistinguishable in the following seven lemmas. Therefore, we conclude that the advantage of the adversary in Game_{real} (i.e., the real security game) is negligible. This completes the proof of Theorem 1.

It should be noted that the indistinguishability between $\text{Game}_{k,1}$ and $\text{Game}_{k,2}$ will require different computa-

tional assumptions for Phase 1 and Phase 2 key queries. We let Q_1 denote the number of Phase 1 queries, and we will address this indistinguishability separately for $k \leq Q_1$ and $k > Q_1$. Our handling of Phase 1 queries will closely resemble the selective security proof strategy for KP-ABE in [3], while our handling of Phase 2 queries will closely resemble the selective security proof strategy for CP-ABE in [7]. \square

Lemma 1. *Suppose that \mathcal{G} satisfies Assumption 1. Then, $\text{Game}_{\text{real}}$ and Game_0 are computationally indistinguishable.*

Proof. Suppose there exists an algorithm \mathcal{A} that distinguishes $\text{Game}_{\text{real}}$ and Game_0 . Then, we can build an algorithm \mathcal{B} with non-negligible advantage in breaking Assumption 1. \mathcal{B} is given g, X_3, X_4, T and will simulate $\text{Game}_{\text{real}}$ or Game_0 with \mathcal{A} . \mathcal{B} chooses $\alpha, a, b, a_0, a_1, \dots, a_n \in \mathbb{Z}_N$ and $Z, Z', Z'', Z_0, Z_1, \dots, Z_n \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $u = g^{a_0}, h_1 = g^{a_1}, \dots, h_n = g^{a_n}$, and sends \mathcal{A} the public parameters

$$\text{PK} = (N, gZ, g^a Z', g^b Z'', e(g, g)^\alpha, V = uZ_0, \{H_i = h_i \cdot Z_i\}, X_4)$$

It can generate normal keys in response to \mathcal{A} 's key requests by using the key generation algorithm, because it knows the $\text{MSK} = (g, u, h_1, \dots, h_n, X_3, a, b, \alpha)$.

At some point, \mathcal{A} sends \mathcal{B} two (equal length) messages M_0, M_1 , and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0), (\mathbf{A}, \rho, \mathcal{T}_1)$, where \mathbf{A} is an $\ell \times m$ matrix. \mathcal{B} chooses $\beta \in \{0, 1\}$ randomly and does the following:

- (1) \mathcal{B} chooses random values $\tilde{v}_2, \dots, \tilde{v}_m, \tilde{v}'_2, \dots, \tilde{v}'_m \in \mathbb{Z}_N$ and creates vectors $\tilde{v} = (1, \tilde{v}_2, \dots, \tilde{v}_m)$ and $\tilde{v}' = (1, \tilde{v}'_2, \dots, \tilde{v}'_m)$.
- (2) \mathcal{B} chooses random values $\tilde{r}_j, \tilde{r}'_j \in \mathbb{Z}_N$ and $Z'_1, Z''_1, Z'_2, Z''_2, \tilde{Z}_{1,j}, Z'_{1,j}, \tilde{Z}_{2,j}, Z'_{2,j} \in \mathbb{G}_{p_4}$ for $1 \leq j \leq \ell$.
- (3) Let $\mathcal{T}_\beta = (t_{\rho(1)}, \dots, t_{\rho(\ell)})$. \mathcal{B} chooses random exponent $\tilde{s} \in \mathbb{Z}_N$ and computes

$$\begin{aligned} \tilde{C}_1 &= M_\beta \cdot e(g^\alpha, T), \quad C'_1 = T^b \cdot Z'_1 \\ C''_1 &= T \cdot Z''_1 \\ C_{1,j} &= T^{aA_j \cdot \tilde{v}} \cdot T^{-(a_0 t_{\rho(j)} + a_{\rho(j)}) \tilde{r}_j} \cdot \tilde{Z}_{1,j} \\ D_{1,j} &= T^{\tilde{r}_j} \cdot Z'_{1,j} \\ \tilde{C}_2 &= e(g^\alpha, T^{\tilde{s}}), \quad C'_2 = T^{b\tilde{s}} \cdot Z'_2 \\ C''_2 &= T^{\tilde{s}} \cdot Z''_2 \\ C_{2,j} &= T^{\tilde{s} a A_j \cdot \tilde{v}'} \cdot T^{-(a_0 t_{\rho(j)} + a_{\rho(j)}) \tilde{r}'_j} \cdot \tilde{Z}_{2,j} \\ D_{2,j} &= T^{\tilde{r}'_j} \cdot Z'_{2,j} \end{aligned}$$

- (4) \mathcal{B} sets the challenge ciphertext as $C = ((\mathbf{A}, \rho), \tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$ and sends it to \mathcal{A} .

If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$, let $T = g^s g_2^c$, then

$$\begin{aligned} \tilde{C}_1 &= M_\beta e(g, g)^{\alpha s}, \quad C'_1 = g^{bs} Z'_1 \cdot g_2^{bc} \\ C''_1 &= g^s Z''_1 \cdot g_2^c \\ C_{1,j} &= g^{aA_j \cdot v} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r_j} Z_{1,j} \cdot g_2^{aA_j w + \gamma_j \eta_{\rho(j)}} \\ D_{1,j} &= g^{r_j} Z'_{1,j} \cdot g_2^{-\gamma_j} \\ \tilde{C}_2 &= e(g, g)^{\alpha s'}, \quad C'_2 = g^{bs'} Z'_2 \cdot g_2^{bc'} \\ C''_2 &= g^{s'} Z''_2 \cdot g_2^{c'} \\ C_{2,j} &= g^{aA_j \cdot v'} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r'_j} Z_{2,j} \cdot g_2^{aA_j w' + \gamma'_j \eta_{\rho(j)}} \\ D_{2,j} &= g^{r'_j} Z'_{2,j} \cdot g_2^{-\gamma'_j} \end{aligned}$$

where $s' = s\tilde{s}$, $c' = c\tilde{s}$, $v = (s, s\tilde{v}_2, \dots, s\tilde{v}_n)$, $v' = (s', s'\tilde{v}'_2, \dots, s'\tilde{v}'_n)$, $r_j = s\tilde{r}_j$, $r'_j = s\tilde{r}'_j$, $Z_{1,j} = \tilde{Z}_{1,j} (Z_0^{t_{\rho(j)}} Z_{\rho(j)})^{r_j}$, $Z_{2,j} = \tilde{Z}_{2,j} (Z_0^{t_{\rho(j)}} Z_{\rho(j)})^{r'_j}$, $w = c\tilde{v}$, $w' = c\tilde{v}'$, $\gamma_j = -c\tilde{r}_j$, $\gamma'_j = -c\tilde{r}'_j$, and $\eta_{\rho(j)} = a_0 t_{\rho(j)} + a_{\rho(j)}$. This is a semi-functional ciphertext and \mathcal{B} simulates Game_0 . We note that the values of $a, b, a_0, a_{\rho(j)}, t_{\rho(j)}, \tilde{s}, \tilde{v}_2, \dots, \tilde{v}_m, \tilde{v}'_2, \dots, \tilde{v}'_m, \tilde{r}_j, \tilde{r}'_j$ modulo p_1 are uncorrelated from their val-

ues modulo p_2 , so this is properly distributed. If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1}$, it is easy to observe that this is a normal ciphertext and \mathcal{B} simulates $\text{Game}_{\text{real}}$. Hence, \mathcal{B} can use the output of \mathcal{A} to distinguish between these possibilities for T . \square

Lemma 2. *Suppose that \mathcal{G} satisfies Assumption 2. Then, $\text{Game}_{k-1,3}$ and $\text{Game}_{k,1}$ are computationally indistinguishable.*

Proof. Suppose there exists an algorithm \mathcal{A} that distinguishes $\text{Game}_{k-1,3}$ and $\text{Game}_{k,1}$. Then, we can build an algorithm \mathcal{B} with non-negligible advantage in breaking Assumption 2. \mathcal{B} is given $g, X_1 X_2, Y_2 Y_3, X_3, X_4, T$ and will simulate $\text{Game}_{k-1,3}$ or $\text{Game}_{k,1}$ with \mathcal{A} . \mathcal{B} chooses $\alpha, a, b, a_0, a_1, \dots, a_n \in \mathbb{Z}_N$ and $Z, Z', Z'', Z_0, Z_1, \dots, Z_n \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $u = g^{a_0}, h_1 = g^{a_1}, \dots, h_n = g^{a_n}$ and sends \mathcal{A} the public parameters:

$$\text{PK} = (N, gZ, g^a Z', g^b Z'', e(g, g)^\alpha, V = uZ_0, \{H_i = h_i \cdot Z_i\}_{1 \leq i \leq n}, X_4)$$

Note that \mathcal{B} knows the master secret key $\text{MSK} = (g, u, h_1, \dots, h_n, X_3, a, b, \alpha)$ associated with PK. Let us now explain how \mathcal{B} answers the j -th key query for $S = (s_1, \dots, s_n)$.

For $j < k$, \mathcal{B} creates a semi-functional key of type 3 by choosing random exponents $t, \tilde{t}, \tilde{d} \in \mathbb{Z}_N$, random elements $R', R'', R_1, \dots, R_n \in \mathbb{G}_{p_3}$, and setting

$$\begin{aligned} K &= g^\alpha g^{at} g^{b\tilde{t}} (Y_2 Y_3)^{\tilde{d}}, \quad K' = g^{\tilde{t}} R', \quad K'' = g^{\tilde{t}} R'' \\ \{K_i &= (u^{s_i} h_i)^t R_i\}_{1 \leq i \leq n} \end{aligned}$$

We note that this is a properly distributed semi-functional key of type 3 because the value of \tilde{d} modulo p_2 is uncorrelated to its value modulo p_3 .

For $j > k$, \mathcal{B} creates a normal key by running the key generation algorithm because it knows the MSK.

To answer the k -th key quest for $\mathcal{S} = (s_1, \dots, s_n)$, \mathcal{B} chooses random exponent $\tilde{t} \in \mathbb{Z}_N$, random elements $\tilde{R}, \tilde{R}', \tilde{R}'', \tilde{R}_1, \dots, \tilde{R}_n \in \mathbb{G}_{p_3}$ and sets

$$K = g^\alpha T^a T^{b\tilde{t}} \tilde{R}, K' = T^{\tilde{t}} \cdot \tilde{R}', K'' = T \cdot \tilde{R}'' \\ \{K_i = T^{a_0 s_i + a_i} \cdot \tilde{R}'_i\}_{1 \leq i \leq n}$$

We have the following observations. If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then T can be written as $g^t g_2^d \tilde{R}$, and

$$K = g^\alpha g^{at} g^{b\tilde{t}} \tilde{R} \cdot g_2^{ad+bd'}, K' = g^{\tilde{t}} \tilde{R}' \cdot g_2^{d'} \\ K'' = g^t \tilde{R}'' \cdot g_2^d, \{K_i = (u^{s_i} h_i)^t \tilde{R}'_i \cdot g_2^{d\eta_i}\}_{1 \leq i \leq n}$$

where $\tilde{t} = \tilde{t}\tilde{t}, d' = \tilde{d}\tilde{t}, R = \tilde{R}^{a+b\tilde{t}} \tilde{R}, R' = \tilde{R}'^{\tilde{t}}, R'' = \tilde{R}''^{\tilde{t}}, R_i = \tilde{R}^{a_0 s_i + a_i} \tilde{R}'_i$, and $\eta_i = a_0 s_i + a_i$. This is a semi-function key of type 1. Note that the values of $\tilde{t}, a, b, a_0, a_i, s_i$ modulo p_1 are uncorrelated from their values modulo p_2 . If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, this is a properly distributed normal key.

At some point, \mathcal{A} sends \mathcal{B} two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, T_0), (\mathbf{A}, \rho, T_1)$, where \mathbf{A} is an $\ell \times m$ matrix. \mathcal{B} chooses $\beta \in \{0, 1\}$ randomly and does the following:

- (1) \mathcal{B} chooses random values $\tilde{v}_2, \dots, \tilde{v}_m, \tilde{v}'_2, \dots, \tilde{v}'_m \in \mathbb{Z}_N$ and creates vectors $\tilde{v} = (1, \tilde{v}_2, \dots, \tilde{v}_m)$ and $\tilde{v}' = (1, \tilde{v}'_2, \dots, \tilde{v}'_m)$.
- (2) \mathcal{B} chooses random values $\tilde{r}_j, \tilde{r}'_j \in \mathbb{Z}_N$ and $Z'_1, Z''_1, Z'_2, Z''_2, \tilde{Z}_{1,j}, Z'_{1,j}, \tilde{Z}_{2,j}, Z'_{2,j} \in \mathbb{G}_{p_4}$ for $1 \leq j \leq \ell$.
- (3) Let $\tilde{T}_\beta = (t_{\rho(1)}, \dots, t_{\rho(\ell)})$. \mathcal{B} chooses random exponent $\tilde{s} \in \mathbb{Z}_N$ and computes

$$\tilde{C}_1 = M_\beta \cdot e(g^\alpha, X_1 X_2) \\ C'_1 = (X_1 X_2)^b \cdot Z'_1, C''_1 = X_1 X_2 \cdot Z''_1 \\ C_{1,j} = (X_1 X_2)^{a_{A_j} \tilde{v}} \\ (X_1 X_2)^{-(a_0 t_{\rho(j)} + a_{\rho(j)}) \tilde{r}_j} \cdot \tilde{Z}_{1,j} \\ D_{1,j} = (X_1 X_2)^{\tilde{r}_j} \cdot Z'_{1,j} \\ \tilde{C}_2 = e(g^\alpha, (X_1 X_2)^{\tilde{s}}) \\ C'_2 = (X_1 X_2)^{b\tilde{s}} \cdot Z'_2, C''_2 = (X_1 X_2)^{\tilde{s}} \cdot Z''_2 \\ C_{2,j} = (X_1 X_2)^{\tilde{s} a_{A_j} \tilde{v}'} \\ (X_1 X_2)^{-(a_0 t_{\rho(j)} + a_{\rho(j)}) \tilde{r}'_j} \cdot \tilde{Z}_{2,j} \\ D_{2,j} = (X_1 X_2)^{\tilde{r}'_j} \cdot Z'_{2,j}$$

- (4) \mathcal{B} sets the challenge ciphertext as $C = ((\mathbf{A}, \rho), \tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$ and sends it to \mathcal{A} .

If we let $X_1 X_2 = g^s g_2^c$, then

$$\tilde{C}_1 = M_\beta \cdot e(g, g)^{\alpha s}, C'_1 = g^{bs} Z'_1 \cdot g_2^{bc} \\ C''_1 = g^s Z''_1 \cdot g_2^c \\ C_{1,j} = g^{a_{A_j} v} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r_j} Z_{1,j} \cdot g_2^{a_{A_j} w + \gamma_j \eta_{\rho(j)}} \\ D_{1,j} = g^{r_j} Z'_{1,j} \cdot g_2^{-\gamma_j} \\ \tilde{C}_2 = e(g, g)^{\alpha s'}, C'_2 = g^{bs'} Z'_2 \cdot g_2^{bc'}, C''_2 = g^{s'} Z''_2 \cdot g_2^{c'} \\ C_{2,j} = g^{a_{A_j} v'} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r'_j} Z_{2,j} \cdot g_2^{a_{A_j} w' + \gamma'_j \eta_{\rho(j)}} \\ D_{2,j} = g^{r'_j} Z'_{2,j} \cdot g_2^{-\gamma'_j}$$

where $s' = s\tilde{s}, c' = c\tilde{s}, v = (s, s\tilde{v}_2, \dots, s\tilde{v}_m), v' = (s', s'\tilde{v}'_2, \dots, s'\tilde{v}'_m), r_j = s\tilde{r}_j, r'_j = s'\tilde{r}'_j, Z_{1,j} = \tilde{Z}_{1,j} (Z_0^{t_{\rho(j)}} Z_{\rho(j)})^{r_j}, Z_{2,j} = \tilde{Z}_{2,j} (Z_0^{t_{\rho(j)}} Z_{\rho(j)})^{r'_j}, w = c\tilde{v}, w' = c\tilde{s}\tilde{v}', \gamma_j = -c\tilde{r}_j, \gamma'_j = -c'\tilde{r}'_j$, and $\eta_{\rho(j)} = a_0 t_{\rho(j)} + a_{\rho(j)}$. This is a semi-functional ciphertext. Note that the values of $a, b, a_0, a_{\rho(j)}, t_{\rho(j)}, \tilde{s}, \tilde{v}_2, \dots, \tilde{v}_m, \tilde{v}'_2, \dots, \tilde{v}'_m, \tilde{r}_j, \tilde{r}'_j$ modulo p_1 are uncorrelated from their values modulo p_2 .

We can conclude that, if $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then \mathcal{B} has properly simulated $\text{Game}_{k,1}$. If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then \mathcal{B} has properly simulated $\text{Game}_{k-1,3}$. Hence, \mathcal{B} can use the output of \mathcal{A} to distinguish between these possibilities for T . \square

Lemma 3. Suppose that \mathcal{G} satisfies Assumption 3. Then, $\text{Game}_{k,1}$ and $\text{Game}_{k,2}$ are computationally indistinguishable for a k from 1 to Q_1 (recall these are all the Phase 1 queries).

Proof. Suppose there exists an algorithm \mathcal{A} that distinguishes $\text{Game}_{k,1}$ and $\text{Game}_{k,2}$ for some k between 1 and Q_1 . Then, we can build an algorithm \mathcal{B} with non-negligible advantage in breaking Assumption 3. \mathcal{B} is given $g, g_2, g_2^x, g_2^y, g_2^z, X_3, X_4, T$ and will simulate $\text{Game}_{k,1}$ or $\text{Game}_{k,2}$ with \mathcal{A} . \mathcal{B} chooses $\alpha, a, b, a_0, a_1, \dots, a_n \in \mathbb{Z}_N$, and $Z, Z', Z'', Z_0, Z_1, \dots, Z_n \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $u = g^{a_0}, h_1 = g^{a_1}, \dots, h_n = g^{a_n}$, and sends \mathcal{A} the public parameters

$$\text{PK} = (N, g, Z, g^a Z', g^b Z'', e(g, g)^\alpha, V = uZ_0 \\ \{H_i = h_i \cdot Z_i\}_{1 \leq i \leq n}, X_4)$$

We note that \mathcal{B} knows the master secret key $\text{MSK} = (g, u, h_1, \dots, h_n, X_3, a, b, \alpha)$ associated with PK and hence can use the normal key generation algorithm to make normal keys in response to \mathcal{A} 's key requests from the $k + 1$ request and onward. To respond to \mathcal{A} 's first $k - 1$ key requests, \mathcal{B} can create semi-functional keys of type 3

because it knows the MSK and g_2 . Let us now explain how \mathcal{B} answers the k -th key query for $\mathcal{S} = (s_1, \dots, s_n)$.

Because we are assuming the k -th key query occurs in Phase 1, \mathcal{S} is declared before \mathcal{B} must produce the challenge ciphertext. This allows \mathcal{B} to define the values η_i modulo p_2 to be shared by the k -th key and the semi-functional challenge ciphertext after learning the set \mathcal{S} . To set these values, \mathcal{B} chooses random exponents $\eta_i \in \mathbb{Z}_N$ for each attribute belonging to \mathcal{S} . For each attribute not belonging to \mathcal{S} , it implicitly sets η_i modulo p_2 to be equal to $x\tilde{\eta}_i$ modulo p_2 , where random exponents $\tilde{\eta}_i \in \mathbb{Z}_N$ are chosen uniformly at random. It also implicitly sets a' equal to xy modulo p_2 .

To form the k -th key, \mathcal{B} first uses the normal key generation algorithm to produce a normal key K, K', K'', K_i . It then choose random exponents $b', d' \in \mathbb{Z}_N$ and implicitly sets d modulo p_2 equal to z modulo p_2 . It sets the key as

$$Kg_2^{b'd'}T, K'g_2^{d'}, K''g_2^z, K_i(g_2^z)^{\eta_i}$$

We observe that if $T = g_2^{xyz}$, this will be a properly distributed semi-functional key of type 1, and when T is a random element in \mathbb{G}_{p_2} , this will be a properly distributed semi-functional key of type 2.

\mathcal{A} sends \mathcal{B} two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0), (\mathbf{A}, \rho, \mathcal{T}_1)$, where \mathbf{A} is an $\ell \times m$ matrix. \mathcal{B} chooses $\beta \in \{0, 1\}$ randomly and does the following:

- (1) \mathcal{B} first runs the normal encryption algorithm to produce a normal ciphertext $(\tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$.
- (2) Let $\mathcal{T}_\beta = (t_{\rho(1)}, \dots, t_{\rho(\ell)})$. \mathcal{B} computes a vector $\tilde{w} \in \mathbb{Z}_N^m$ such that $\tilde{w} \cdot A_j = 0$ modulo N for all j such that $t_{\rho(j)} = s_{\rho(j)}$, and the first entry of \tilde{w} is nonzero modulo each prime dividing N . Note that, as shown in [34], because the attribute set \mathcal{S} cannot satisfy the access structure of $(\mathbf{A}, \rho, \mathcal{T}_\beta)$, the vector \tilde{w} can be efficiently found by \mathcal{B} .
- (3) \mathcal{B} also chooses two random vectors w_1 and $w_2 \in \mathbb{Z}_N^m$. It will implicitly set the sharing vectors w, w' modulo p_2 so that $a'w = xy\tilde{w} + w_1$ and $a'w' = xy\tilde{w} + w_2$. \mathcal{B} also chooses random values $\gamma_j, \gamma'_j \in \mathbb{Z}_N$ for each j such that $t_{\rho(j)} = s_{\rho(j)}$, and random values $\tilde{\gamma}_j, \tilde{\gamma}'_j \in \mathbb{Z}_N$ for each j such that $t_{\rho(j)} \neq s_{\rho(j)}$. For these j 's such that $t_{\rho(j)} \neq s_{\rho(j)}$, it will implicitly set $\gamma_j = -y\tilde{\eta}_{\rho(j)}^{-1}A_j \cdot \tilde{w} + \tilde{\gamma}_j$ and $\gamma'_j = -y\tilde{\eta}_{\rho(j)}^{-1}A_j \cdot \tilde{w} + \tilde{\gamma}'_j$.
- (4) It chooses random exponents $c, c' \in \mathbb{Z}_N$ and forms the semi-functional challenge ciphertext as

$$\begin{aligned} &\tilde{C}_1, C'_1 \cdot g_2^{b'c}, C''_1 \cdot g_2^c \\ &C_{1,j} \cdot g_2^{A_j w_1 + \gamma_j \eta_{\rho(j)}}, D_{1,j} \cdot g_2^{-\gamma_j} \\ &\quad \forall j \text{ s.t. } t_{\rho(j)} = s_{\rho(j)} \\ &C_{1,j} \cdot g_2^{A_j w_1} (g_2^x)^{\tilde{\gamma}_j \tilde{\eta}_{\rho(j)}} \\ &D_{1,j} \cdot (g_2^y)^{\tilde{\eta}_{\rho(j)}^{-1} A_j \cdot \tilde{w}} g_2^{-\tilde{\gamma}_j} \\ &\quad \forall j \text{ s.t. } t_{\rho(j)} \neq s_{\rho(j)} \end{aligned}$$

$$\begin{aligned} &\tilde{C}_2, C'_2 \cdot g_2^{b'c'}, C''_2 \cdot g_2^{c'} \\ &C_{2,j} \cdot g_2^{A_j w_2 + \gamma'_j \eta_{\rho(j)}}, D_{2,j} \cdot g_2^{-\gamma'_j} \\ &\quad \forall j \text{ s.t. } t_{\rho(j)} = s_{\rho(j)} \\ &C_{2,j} \cdot g_2^{A_j w_2} (g_2^x)^{\tilde{\gamma}'_j \tilde{\eta}_{\rho(j)}} \\ &D_{2,j} \cdot (g_2^y)^{\tilde{\eta}_{\rho(j)}^{-1} A_j \cdot \tilde{w}} g_2^{-\tilde{\gamma}'_j} \\ &\quad \forall j \text{ s.t. } t_{\rho(j)} \neq s_{\rho(j)} \end{aligned}$$

Observe that this is a properly formed semi-functional ciphertext.

We can thus conclude that, if $T = g_2^{xyz}$, then \mathcal{B} has properly simulated $\text{Game}_{k,1}$. If T is a random element in \mathbb{G}_{p_2} , then \mathcal{B} has properly simulated $\text{Game}_{k,2}$. Hence, \mathcal{B} can use the output of \mathcal{A} to distinguish between these possibilities for T . \square

Lemma 4. Suppose that \mathcal{G} satisfies Assumption 4. Then $\text{Game}_{k,1}$ and $\text{Game}_{k,2}$ are computationally indistinguishable for a $k > Q_1$ using an access matrix (\mathbf{A}, ρ) of size $\ell \times m$ where $\ell, n \leq q$.

Proof. Suppose there exists an algorithm \mathcal{A} that distinguishes $\text{Game}_{k,1}$ and $\text{Game}_{k,2}$ for some k such that $Q_1 < k \leq Q$ using an access matrix with dimensions $\leq q$. Then, we can build an algorithm \mathcal{B} with non-negligible advantage in breaking Assumption 4. \mathcal{B} is given $g, g_2, g_2^f, g_2^{x^f}, g_2^{y^f} \forall i \in [2q] \setminus \{q+1\}, g_2^{y^i/z_j} \forall i \in [2q] \setminus \{q+1\}, j \in [q], g_2^{x^f z_j} \forall j \in [q], g_2^{x^f y^i z_j / z_j} \forall i \in [q], j, j' \in [q]$ such that $j \neq j', X_3, X_4, T$ and will simulate $\text{Game}_{k,1}$ or $\text{Game}_{k,2}$ with \mathcal{A} . \mathcal{B} chooses $\alpha, a, b, a_0, a_1, \dots, a_n \in \mathbb{Z}_N$ and $Z, Z', Z'', Z_0, Z_1, \dots, Z_n \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $u = g^{a_0}, h_1 = g^{a_1}, \dots, h_n = g^{a_n}$ and sends \mathcal{A} the public parameters

$$\text{PK} = (N, gZ, g^a Z', g^b Z'', e(g, g)^\alpha, V = uZ_0, \{H_i = h_i \cdot Z_i\}_{1 \leq i \leq n}, X_4)$$

We note that \mathcal{B} knows the master secret key $\text{MSK} = (g, u, h_1, \dots, h_n, X_3, a, b, \alpha)$ associated with PK and hence can use the normal key generation algorithm to make normal keys in response to \mathcal{A} 's key requests from the $k+1$ request and onward. To respond to \mathcal{A} 's first $k-1$ key requests, \mathcal{B} can create semi-functional keys of type 3 because it knows the MSK and g_2 . Because we are assuming the k -th key query occurs in Phase 2, \mathcal{A} will request the challenge ciphertext before requesting the k -th key.

\mathcal{A} sends \mathcal{B} two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0), (\mathbf{A}, \rho, \mathcal{T}_1)$ for a challenge ciphertext, where \mathbf{A} is an $\ell \times m$ matrix. \mathcal{B} first chooses $\beta \in \{0, 1\}$ randomly and runs the normal encryption algorithm for $(\mathbf{A}, \rho, \mathcal{T}_\beta)$ to produce a normal ciphertext $(\tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$. \mathcal{B} chooses random values $\tilde{b}, \tilde{\eta}_j, \tilde{\gamma}_j, \tilde{\gamma}'_j \in \mathbb{Z}_N$. It will implicitly set $a' = xy$ modulo p_2 , $b' = x + b$ modulo p_2 , $\gamma_j = xz_j + \tilde{\gamma}_j$, and $\gamma'_j = xz_j + \tilde{\gamma}'_j$ for

each j from 1 to ℓ . Let $\bar{\mathcal{T}}_\beta = (t_{\rho(1)}, \dots, t_{\rho(\ell)})$. For each attribute i , we let J_i denote the set of indices j such that $t_{\rho(j)} = i$. \mathcal{B} define $g_2^{\eta_i}$ as

$$g_2^{\eta_i} = g_2^{\tilde{\eta}_i} \prod_{j \in J_i} \left(g_2^{y/z_j}\right)^{A_{j,1}} \cdot \left(g_2^{y^2/z_j}\right)^{A_{j,2}} \cdots \left(g_2^{y^m/z_j}\right)^{A_{j,m}}$$

It also chooses random values $\tilde{c}, v_1, \dots, v_n, v'_1, \dots, v'_n \in \mathbb{Z}_N$ and implicitly sets $c = f$ modulo p_2 , $c' = \tilde{c}f$ modulo p_2 , $w = (f + v_1(a')^{-1}, fy + v_2(a')^{-1}, \dots, fy^{m-1} + v_n(a')^{-1})$, and $w' = (f + v'_1(a')^{-1}, fy + v'_2(a')^{-1}, \dots, fy^{m-1} + v'_n(a')^{-1})$. Then, \mathcal{B} computes the semi-functional challenge ciphertext as

$$\left((\mathbf{A}, \rho), \tilde{C}_1, C'_1 \cdot g_2^{b'c}, C''_1 \cdot g_2^c \right. \\ \left. \left\{ C_{1,j} \cdot g_2^{a' A_j w + \gamma_j \eta_{\rho(j)}}, D_{1,j} \cdot g_2^{-\gamma_j} \right\}_{1 \leq j \leq \ell} \right. \\ \tilde{C}_2, C'_2 \cdot g_2^{b'c'}, C''_2 \cdot g_2^{c'}, \\ \left. \left\{ C_{2,j} \cdot g_2^{a' A_j w' + \gamma'_j \eta_{\rho(j)}}, D_{2,j} \cdot g_2^{-\gamma'_j} \right\}_{1 \leq j \leq \ell} \right)$$

Note that the semi-functional components (the parts in \mathbb{G}_{p_2}) in the challenge ciphertext can be computed by \mathcal{B} using the \mathbb{G}_{p_2} elements in the assumption.

Let us now explain how \mathcal{B} answers the k -th key query for $\mathcal{S} = (s_1, \dots, s_n)$. To form the k -th key, \mathcal{B} first uses the normal key generation algorithm to produce a normal key K, K', K'', K_i . Because the attribute set \mathcal{S} cannot satisfy the access structure of $(\mathbf{A}, \rho, \bar{\mathcal{T}}_\beta)$, \mathcal{B} can efficiently find a vector $\theta = (\theta_1, \dots, \theta_m) \in \mathbb{Z}_N^m$ such that $\theta \cdot A_j = 0$ modulo N for all j such that $t_{\rho(j)} = s_{\rho(j)}$, and the first entry of θ is nonzero modulo each prime dividing N . \mathcal{B} chooses random exponent $\tilde{d} \in \mathbb{Z}_N$ and implicitly sets

$$d' = -\theta_2 y^q - \theta_3 y^{q-1} - \dots - \theta_m y^{q-m+2} + f \tilde{d} \\ d = \theta_1 y^q + \theta_2 y^{q-1} + \dots + \theta_m y^{q-m+1}$$

It sets the k -th key as

$$K \cdot T^{\theta_1} \left(g_2^{y^q}\right)^{-\tilde{b}\theta_2} \cdots \left(g_2^{y^{q-m+2}}\right)^{-\tilde{b}\theta_m} \left(g_2^{xf}\right)^{\tilde{d}} \left(g_2^f\right)^{\tilde{b}\tilde{d}} \\ K' g_2^{d'}, K'' g_2^d, K_i \left(g_2^d\right)^{\eta_i}$$

Note that the parts in \mathbb{G}_{p_2} in the k -th key can be computed by \mathcal{B} using the \mathbb{G}_{p_2} elements in the assumption. We observe that if $T = g_2^{xy^{q+1}}$, this will be a properly distributed semi-functional key of type 1, and when T is a random element in \mathbb{G}_{p_2} , this will be a properly distributed semi-functional key of type 2.

We can thus conclude that, if $T = g_2^{xy^{q+1}}$, then \mathcal{B} has properly simulated $\text{Game}_{k,1}$. If T is a random element in \mathbb{G}_{p_2} , then \mathcal{B} has properly simulated $\text{Game}_{k,2}$. Hence, \mathcal{B} can use the output of \mathcal{A} to distinguish between these possibilities for T . \square

Lemma 5. Suppose that \mathcal{G} satisfies Assumption 2. Then, $\text{Game}_{k,2}$ and $\text{Game}_{k,3}$ are computationally indistinguishable.

Proof. Suppose there exists an algorithm \mathcal{A} that distinguishes $\text{Game}_{k,2}$ and $\text{Game}_{k,3}$. Then, we can build an algorithm \mathcal{B} with non-negligible advantage in breaking Assumption 2. \mathcal{B} is given $g, X_1 X_2, Y_2 Y_3, X_3, X_4, T$ and will simulate $\text{Game}_{k,2}$ or $\text{Game}_{k,3}$ with \mathcal{A} . \mathcal{B} chooses $\alpha, a, b, a_0, a_1, \dots, a_n \in \mathbb{Z}_N$ and $Z, Z', Z'', Z_0, Z_1, \dots, Z_n \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $u = g^{a_0}, h_1 = g^{a_1}, \dots, h_n = g^{a_n}$ and sends \mathcal{A} the public parameters

$$\text{PK} = (N, gZ, g^a Z', g^b Z'', e(g, g)^\alpha, V = uZ_0 \\ \{H_i = h_i \cdot Z_i\}_{1 \leq i \leq n}, X_4)$$

The first $k-1$ semi-functional keys of type 3, the normal keys should greater than k , and the challenge ciphertext are constructed exactly as in the Lemma 2.

To answer the k -th key quest for $\mathcal{S} = (s_1, \dots, s_n)$, \mathcal{B} proceeds as it did in the Lemma 2, but \mathcal{B} additionally chooses a random exponent $\delta \in \mathbb{Z}_N$ and sets

$$K = g^\alpha T^a T^{b\tilde{r}} \tilde{R} \cdot (Y_2 Y_3)^\delta, K' = T^{\tilde{r}} \cdot \tilde{R}' \\ K'' = T \cdot \tilde{R}'', \{K_i = T^{a_0 s_i + a_i} \cdot \tilde{R}'_i\}_{1 \leq i \leq n}$$

The only change we have made here is adding the $(Y_2 Y_3)^\delta$ term, which randomizes the \mathbb{G}_{p_2} part of K . If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 2. If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 3.

We can conclude that, if $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then \mathcal{B} has properly simulated $\text{Game}_{k,2}$. If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then \mathcal{B} has properly simulated $\text{Game}_{k,3}$. Hence, \mathcal{B} can use the output of \mathcal{A} to distinguish between these possibilities for T . \square

Lemma 6. Suppose that \mathcal{G} satisfies Assumption 5. Then, $\text{Game}_{Q,3}$ and $\text{Game}_{\text{Final}_0}$ are computationally indistinguishable.

Proof. Suppose there exists an algorithm \mathcal{A} that distinguishes $\text{Game}_{Q,3}$ and $\text{Game}_{\text{Final}_0}$. Then, we can build an algorithm \mathcal{B} with non-negligible advantage in breaking Assumption 5. \mathcal{B} is given $g, g_2, g^\alpha X_2, g^\delta Y_2, X_3, X_4, T$ and will simulate $\text{Game}_{Q,3}$ or $\text{Game}_{\text{Final}_0}$ with \mathcal{A} . \mathcal{B} chooses $a, b, a_0, a_1, \dots, a_n \in \mathbb{Z}_N$ and $Z, Z', Z'', Z_0, Z_1, \dots, Z_n \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $u = g^{a_0}, h_1 = g^{a_1}, \dots, h_n = g^{a_n}$ and sends \mathcal{A} the public parameters

$$\text{PK} = (N, gZ, g^a Z', g^b Z'', e(g, g^\alpha X_2) = e(g, g)^\alpha \\ V = uZ_0, \{H_i = h_i \cdot Z_i\}_{1 \leq i \leq n}, X_4)$$

Each time \mathcal{B} is asked to provide a key for $\mathcal{S} = (s_1, \dots, s_n)$, \mathcal{B} creates a semi-functional key of type 3 by choosing

random exponents $t, \tilde{t}, \tilde{d} \in \mathbb{Z}_N$, random elements $R, R', R'', R_1, \dots, R_n \in \mathbb{G}_{p_3}$, and setting

$$K = (g^\alpha X_2) g^{at} g^{b\tilde{t}} R \cdot g_2^{\tilde{d}}, K' = g^{\tilde{t}} R', K'' = g^t R'' \\ \{K_i = (u^{s_i} h_i)^t R_i\}_{1 \leq i \leq n}$$

We note that K can be written as $g^\alpha g^{at} g^{b\tilde{t}} R \cdot g_2^{\tilde{d}}$, where $g_2^{\tilde{d}} = X_2 g_2^{\tilde{d}}$, so this is a properly distributed semi-functional key of type 3.

At some point, \mathcal{A} sends \mathcal{B} two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0)$, $(\mathbf{A}, \rho, \mathcal{T}_1)$, where \mathbf{A} is an $\ell \times m$ matrix. \mathcal{B} chooses $\beta \in \{0, 1\}$ randomly and does the following:

- (1) \mathcal{B} chooses random values $\tilde{v}_2, \dots, \tilde{v}_m \in \mathbb{Z}_N$ and creates the vector $\tilde{v} = (1, \tilde{v}_2, \dots, \tilde{v}_m)$. \mathcal{B} also chooses two random vectors $v' = (s', v'_2, \dots, v'_m)$ and $w' = (w'_1, \dots, w'_m) \in \mathbb{Z}_N^m$.
- (2) \mathcal{B} chooses random values $\tilde{r}_j, r'_j, \gamma'_j \in \mathbb{Z}_N$ and $Z'_1, Z''_1, Z'_2, Z''_2, \tilde{Z}_{1,j}, Z'_{1,j}, Z_{2,j}, Z''_{2,j} \in \mathbb{G}_{p_4}$ for $1 \leq j \leq \ell$.
- (3) Let $\mathcal{T}_\beta = (t_{\rho(1)}, \dots, t_{\rho(\ell)})$. \mathcal{B} chooses random exponent $c' \in \mathbb{Z}_N$ and computes

$$\tilde{C}_1 = M_\beta \cdot T, C'_1 = (g^s Y_2)^b \cdot Z'_1 \\ C''_1 = g^s Y_2 \cdot Z''_1$$

$$C_{1,j} = (g^s Y_2)^{a_{A_j} \tilde{v}} (g^s Y_2)^{-(a_0 t_{\rho(j)} + a_{\rho(j)}) \tilde{r}_j} \tilde{Z}_{1,j}$$

$$D_{1,j} = (g^s Y_2)^{\tilde{r}_j} \cdot Z'_{1,j}$$

$$\tilde{C}_2 = e(g, g)^{\alpha s'}, C'_2 = g^{bs'} g_2^{bc'} \cdot Z'_2$$

$$C''_2 = g^{s'} g_2^{c'} \cdot Z''_2$$

$$C_{2,j} = g^{a_{A_j} v'} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r'_j} Z_{2,j} \\ \cdot g_2^{a_{A_j} w' + \gamma'_j (a_0 t_{\rho(j)} + a_{\rho(j)})}$$

$$D_{2,j} = g^{r'_j} Z'_{2,j} \cdot g_2^{-\gamma'_j}$$

- (4) \mathcal{B} sets the challenge ciphertext as $C = ((\mathbf{A}, \rho), \tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$ and sends it to \mathcal{A} .

Let $g^s Y_2 = g^s g_2^c$, then

$$\tilde{C}_1 = M_\beta \cdot T, C'_1 = g^{bs} Z'_1 \cdot g_2^{bc}, C''_1 = g^s Z''_1 \cdot g_2^c$$

$$C_{1,j} = g^{a_{A_j} v} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r_j} Z_{1,j} \cdot g_2^{a_{A_j} w + \gamma_j \eta_{\rho(j)}}$$

$$D_{1,j} = g^{r_j} Z'_{1,j} \cdot g_2^{-\gamma_j}$$

$$\tilde{C}_2 = e(g, g)^{\alpha s'}, C'_2 = g^{bs'} Z'_2 \cdot g_2^{bc'}, C''_2 = g^s Z''_2 \cdot g_2^c$$

$$C_{2,j} = g^{a_{A_j} v'} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r'_j} Z_{2,j} \cdot g_2^{a_{A_j} w' + \gamma'_j \eta_{\rho(j)}}$$

$$D_{2,j} = g^{r'_j} Z'_{2,j} \cdot g_2^{-\gamma'_j}$$

where $v = (s, s\tilde{v}_2, \dots, s\tilde{v}_m)$, $r_j = s\tilde{r}_j$, $Z_{1,j} = \tilde{Z}_{1,j} (Z_0^{t_{\rho(j)}} Z_{\rho(j)})^{r_j}$, $w = c\tilde{v}$, $\gamma_j = -c\tilde{r}_j$, and $\eta_{\rho(j)} = a_0 t_{\rho(j)} + a_{\rho(j)}$. Note that the values of $a, b, a_0, a_{\rho(j)}, t_{\rho(j)}, \tilde{v}_2, \dots, \tilde{v}_m, \tilde{r}_j$ modulo p_1 are uncorrelated from their values modulo p_2 .

If $T = e(g, g)^{\alpha s}$, this is a properly distributed semi-functional encryption of M_β and \mathcal{B} simulates $\text{Game}_{Q,3}$. Otherwise, this is a properly distributed semi-functional encryption of a random message in \mathbb{G}_T , and \mathcal{B} simulates $\text{Game}_{\text{Final}_0}$. Hence, \mathcal{B} can use the output of \mathcal{A} to distinguish between these possibilities for T . \square

Lemma 7. Suppose that \mathcal{G} satisfies Assumption 6. Then $\text{Game}_{\text{Final}_0}$ and $\text{Game}_{\text{Final}_1}$ are computationally indistinguishable.

Proof. Suppose there exists an algorithm \mathcal{A} that distinguishes $\text{Game}_{\text{Final}_0}$ and $\text{Game}_{\text{Final}_1}$. Then we can build an algorithm \mathcal{B} with non-negligible advantage in breaking Assumption 6. \mathcal{B} is given $(g, g_2, g^a X_2, g^a Z', g^s Y_2, X_3, X_4, T)$ and will simulate $\text{Game}_{\text{Final}_0}$ or $\text{Game}_{\text{Final}_1}$ with \mathcal{A} . \mathcal{B} chooses $\alpha, b, a_0, a_1, \dots, a_n \in \mathbb{Z}_N$ and $Z, Z'', Z_0, Z_1, \dots, Z_n \in \mathbb{G}_{p_4}$ uniformly at random. It then sets $u = g^{a_0}, h_1 = g^{a_1}, \dots, h_n = g^{a_n}$ and sends \mathcal{A} the public parameters

$$\text{PK} = (N, gZ, g^a Z', g^b Z'', e(g, g)^\alpha \\ V = uZ_0, \{H = h_i \cdot Z_i\}_{1 \leq i \leq n}, X_4)$$

Each time \mathcal{B} is asked to provide a key for $\mathcal{S} = (s_1, \dots, s_n)$, \mathcal{B} creates a semi-functional key of type 3 by choosing random exponents $t, \tilde{t} \in \mathbb{Z}_N$, random elements $R, R', R'', R_1, \dots, R_n \in \mathbb{G}_{p_3}$, and setting

$$K = g^\alpha (g^a X_2)^t g^{b\tilde{t}} R, K' = g^{\tilde{t}} R', K'' = g^t R'' \\ \{K_i = (u^{s_i} h_i)^t R_i\}_{1 \leq i \leq n}$$

Observe that it is a properly distributed semi-functional key of type 3 because the values of t modulo p_2 is uncorrelated to their values modulo p_1 .

At some point, \mathcal{A} sends \mathcal{B} two (equal length) messages M_0, M_1 and two access structures $(\mathbf{A}, \rho, \mathcal{T}_0)$, $(\mathbf{A}, \rho, \mathcal{T}_1)$, where \mathbf{A} is an $\ell \times m$ matrix. \mathcal{B} chooses $\beta \in \{0, 1\}$ randomly and does the following:

- (1) \mathcal{B} chooses random vectors $\tilde{v} = (1, \tilde{v}_2, \dots, \tilde{v}_m)$ and $\tilde{v}' = (1, \tilde{v}'_2, \dots, \tilde{v}'_m)$.
- (2) \mathcal{B} also chooses $r_j, r'_j, \gamma_j, \gamma'_j, \eta_{\rho(j)} \in \mathbb{Z}_N$ and $\tilde{Z}'_1, \tilde{Z}''_1, \tilde{Z}'_2, \tilde{Z}''_2, \tilde{Z}_{1,j}, Z'_{1,j}, \tilde{Z}_{2,j}, Z'_{2,j} \in \mathbb{G}_{p_4}$ uniformly at random, for $1 \leq j \leq \ell$.

Table I. Comparison of CP-ABE schemes, where ‘linear’ means that the size of ciphertext scales linearly with the complexity of the access structure.

Scheme	Anonymity of access structures	Expressiveness of access structures	Security	Ciphertext size
CP-ABE [6]	No	LSSS	Fully secure	Linear
IPE* [6]	Fully hidden	Inner-product predicates	Fully secure	Linear
[23,24]	Partially hidden	AND-gates on multi-valued Attributes with wildcards	Selectively secure	Linear
[27]	Partially hidden	AND-gates on multi-valued Attributes with wildcards	Fully secure	Linear
Ours	Partially hidden	LSSS	Fully secure	Linear

CP-ABE, ciphertext-policy attribute-based encryption; IPE, inner-product predicate encryption; LSSS, linear secret-sharing scheme.

*In a CP-ABE scheme with fully hidden access structure which is derived from attribute-hiding IPE, the access structure must be converted to an inner-product predicate, and this causes a superpolynomial blowup in ciphertext size.

(3) Let $\mathcal{T}_\beta = (t_{\rho(1)}, \dots, t_{\rho(\ell)})$. \mathcal{B} chooses random exponents $\tilde{s} \in \mathbb{Z}_N$ and sets

$$\begin{aligned} \tilde{C}_1 &\stackrel{\$}{\leftarrow} \mathbb{G}_T, C'_1 = (g^s Y_2 Y_4)^b \cdot \tilde{Z}'_1 \\ C''_1 &= g^s Y_2 Y_4 \cdot \tilde{Z}''_1 \\ C_{1,j} &= T^{A_j \tilde{v}} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r_j} \cdot \tilde{Z}_{1,j} \cdot g_2^{\gamma_j \eta_{\rho(j)}} \\ D_{1,j} &= g^{r_j} Z'_{1,j} \cdot g_2^{-\gamma_j} \\ \tilde{C}_2 &= e(g, g^s Y_2 Y_4)^{\alpha \tilde{s}} = e(g, g)^{\alpha s \tilde{s}} \\ C'_2 &= (g^s Y_2 Y_4)^{b \tilde{s}} \cdot \tilde{Z}'_2, C''_2 = (g^s Y_2 Y_4)^{\tilde{s}} \cdot \tilde{Z}''_2 \\ C_{2,j} &= T^{\tilde{s} A_j \tilde{v}'} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r'_j} \cdot \tilde{Z}_{2,j} \cdot g_2^{\gamma'_j \eta_{\rho(j)}} \\ D_{2,j} &= g^{r'_j} Z'_{2,j} \cdot g_2^{-\gamma'_j} \end{aligned}$$

(4) \mathcal{B} sets the challenge ciphertext as $C = ((\mathbf{A}, \rho), \tilde{C}_1, C'_1, C''_1, \{C_{1,j}, D_{1,j}\}_{1 \leq j \leq \ell}, \tilde{C}_2, C'_2, C''_2, \{C_{2,j}, D_{2,j}\}_{1 \leq j \leq \ell})$ and sends it to \mathcal{A} .

If $T = g^{as} D_2 D_4$, let $D_2 = g_2^\gamma$, we have

$$\begin{aligned} \tilde{C}_1 &\stackrel{\$}{\leftarrow} \mathbb{G}_T, C'_1 = g^{bs} Z'_1 \cdot g_2^{bc}, C''_1 = g^s Z''_1 \cdot g_2^c \\ C_{1,j} &= g^{A_j v} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r_j} Z_{1,j} \cdot g_2^{a' A_j w + \gamma_j \eta_{\rho(j)}} \\ D_{1,j} &= g^{r_j} Z'_{1,j} \cdot g_2^{-\gamma_j} \\ \tilde{C}_2 &= e(g, g)^{\alpha s'}, C'_2 = g^{bs'} Z'_2 \cdot g_2^{bc'}, C''_2 = g^{s'} Z''_2 \cdot g_2^{c'} \\ C_{2,j} &= g^{A_j v'} (V^{t_{\rho(j)}} H_{\rho(j)})^{-r'_j} Z_{2,j} \cdot g_2^{a' A_j w' + \gamma'_j \eta_{\rho(j)}} \\ D_{2,j} &= g^{r'_j} Z'_{2,j} \cdot g_2^{-\gamma'_j} \end{aligned}$$

where $v = (s, s\tilde{v}_2, \dots, s\tilde{v}_m)$, $Z'_1 = Y_4^b \tilde{Z}'_1$, $Z''_1 = Y_4 \tilde{Z}''_1$, $g_2^c = Y_2$, $s' = s\tilde{s}$, $v' = (s', s'\tilde{v}'_2, \dots, s'\tilde{v}'_m)$, $Z'_2 = Y_4^{b\tilde{s}} \tilde{Z}'_2$, $Z''_2 = Y_4^{\tilde{s}} \tilde{Z}''_2$, $g_2^{c'} = Y_2^{\tilde{s}}$, $a'w = \gamma\tilde{v}$, $a'w' = \gamma\tilde{s}\tilde{v}'$, $Z'_{1,j} = D_4^{A_j \tilde{v}} \tilde{Z}'_{1,j}$, and $Z_{2,j} = D_4^{\tilde{s} A_j \tilde{v}'} \tilde{Z}_{2,j}$. This is a properly distributed semi-functional encryption of a random message

in \mathbb{G}_T . If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$, this is a properly distributed semi-functional ciphertext with \tilde{C}_1 random in \mathbb{G}_T , and $C_{1,j}, C_{2,j}$ random in $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$.

We can conclude that, if $T = g^{as} D_2 D_4$, then \mathcal{B} has properly simulated $\text{Game}_{\text{Final}_0}$. If $T \stackrel{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$, then \mathcal{B} has properly simulated $\text{Game}_{\text{Final}_1}$. Hence, \mathcal{B} can use the output of \mathcal{A} to distinguish between these possibilities for T . \square

Comparison. There exist a few efforts [23,24,27] on CP-ABE with partially hidden access structure. However, these schemes only support restricted access structures, which can be expressed as AND gates on multi-valued attributes with wildcards. Compared with these schemes, our scheme is more flexible and expressive. An overview comparing our CP-ABE scheme to those of other CP-ABE schemes with hidden access structure is given in Table I. The table shows that our scheme is superior to all the other CP-ABE schemes with partially hidden access structure because our scheme handles the most expressive access structures and is fully secure in the standard model.

5. CONCLUSIONS

In this paper, we considered a new model for CP-ABE with partially hidden access structure and presented a concrete construction. Our scheme is able to handle any access structure that can be expressed as an LSSS. Previous CP-ABE schemes with partially hidden access structure [23,24,27] only support restricted access structures, which can be expressed as AND gates on multi-valued attributes with wildcards; thus, our scheme is more flexible and expressive. We also showed that CP-ABE with partially hidden access structure is more appropriate to use in constructing privacy-preserving EMR systems than the standard CP-ABE schemes used in previous work [30–33].

By applying the methodology proposed by Lewko and Waters [1] recently, we proved that our scheme is fully secure in the standard model. The security of our scheme

relies on some non-standard complexity assumptions. A further direction is to find expressive CP-ABE constructions with partially hidden access structure based on simple assumptions.

ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China (nos. 61572235, 61300226), Research Fund for the Doctoral Program of Higher Education of China (no. 20134401120017), Guangdong Natural Science Funds for Distinguished Young Scholar (no. 2015A030306045), ISN Research Fund (no. ISN15-04), and Pearl River S&T Nova Program of Guangzhou.

REFERENCES

- Lewko AB, Waters B. New proof methods for attribute-based encryption: achieving full security through selective techniques. In *CRYPTO*, Santa Barbara, CA, USA, 2012; 180–198.
- Sahai A, Waters B. Fuzzy identity-based encryption. In *EUROCRYPT*, Aarhus, Denmark, 2005; 457–473.
- Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, 2006; 89–98.
- Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, Oakland, California, USA, 2007; 321–334.
- Cheung L, Newport CC. Provably secure ciphertext policy ABE. *ACM Conference on Computer and Communications Security*, Alexandria, Virginia, USA, 2007; 456–465.
- Lewko AB, Okamoto T, Sahai A, Takashima K, Waters B. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner-product encryption. In *EUROCRYPT*, French Riviera, 2010; 62–91.
- Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In *Public Key Cryptography*, Taormina, Italy, 2011; 53–70.
- Katz J, Sahai A, Waters B. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, Istanbul, Turkey, 2008; 146–162.
- Lai J, Deng RH, Li Y. Expressive cp-abe with partially hidden access structures. In *ASIACCS*, Seoul, Korea, 2012.
- Waters B. Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. *CRYPTO*, Santa Barbara, CA, USA, 2009; 619–636.
- Goyal V, 0002 AJ, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. In *ICALP (2)*, Reykjavik, Iceland, 2008; 579–591.
- Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. In *ACM Conference on Computer and Communications Security*, Alexandria, Virginia, USA, 2007; 195–203.
- Chase M. Multi-authority attribute based encryption. In *TCC*, Amsterdam, The Netherlands, 2007; 515–534.
- Lin H, Cao Z, Liang X, Shao J. Secure threshold multi authority attribute based encryption without a central authority. In *INDOCRYPT*, Kharagpur, India, 2008; 426–436.
- Chase M, Chow S S M. Improving privacy and security in multi-authority attribute-based encryption. In *Acm Conference on Computer and Communications Security*, Chicago, Illinois, USA, 2009; 121–130.
- Lewko AB, Waters B. Unbounded HIBE and attribute-based encryption. In *EUROCRYPT*, Tallinn, Estonia, 2011; 547–567.
- Shi E, Waters B. Delegating capabilities in predicate encryption systems. In *ICALP (2)*, Reykjavik, Iceland, 2008; 560–578.
- Okamoto T, Takashima K. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, Tokyo, Japan, 2009; 214–231.
- Shen E, Shi E, Waters B. Predicate privacy in encryption systems. In *TCC*, San Francisco, CA, USA, 2009; 457–473.
- Okamoto T, Takashima K. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, Santa Barbara, CA, USA, 2010; 191–208.
- Müller S, Katzenbeisser S, Eckert C. Distributed attribute-based encryption. In *ICISC*, Seoul, Korea, 2008; 20–36.
- Lewko AB, Waters B. Decentralizing attribute-based encryption. In *EUROCRYPT*, Tallinn, Estonia, 2011; 568–588.
- Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures. In *ACNS*, New York, NY, USA, 2008; 111–129.
- Li J, Ren K, Zhu B, Wan Z. Privacy-aware attribute-based encryption with user accountability. *ISC*, Pisa, Italy, 2009; 347–362.
- Canetti R, Halevi S, Katz J. A forward-secure public-key encryption scheme. *EUROCRYPT*, Warsaw, Poland, 2003; 255–271.
- Boneh D, Boyen X. Efficient selective-ID secure identity-based encryption without random oracles.

- In *EUROCRYPT*, Interlaken, Switzerland, 2004; 223–238.
27. Lai J, Deng RH, Li Y. Fully secure ciphertext-policy hiding CP-ABE. In *ISPEC*, Guangzhou, China, 2011; 24–39.
 28. Lewko AB, Waters B. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, Zurich, Switzerland, 2010; 455–479.
 29. Lewko AB, Rouselakis Y, Waters B. Achieving leakage resilience through dual system encryption. In *TCC*, Providence, RI, USA, 2011; 70–88.
 30. Narayan S, Gagné M, Safavi-Naini R. Privacy preserving ehr system using attribute-based infrastructure. In *CCSW*, Chicago, IL, USA, 2010; 47–52.
 31. Kamara S, Lauter K. Cryptographic cloud storage. In *Financial Cryptography Workshops*, Tenerife, Canary Islands, Spain, 2010; 136–149.
 32. Li M, Yu S, Ren K, Lou W. Securing personal health records in cloud computing: patient-centric and fine-grained data access control in multi-owner settings. *Securecomm*, Singapore, 2010; 89–106.
 33. Akinyele JA, Lehmann CU, Green M, Pagano MW, Peterson ZNJ, Rubin AD. Self-protecting electronic medical records using attribute-based encryption. *IACR Cryptology ePrint Archive* 2010; **2010**: 565.
 34. Beimel A. Secure schemes for secret sharing and key distribution. *PhD Thesis*, Israel Institute of Technology, 1996.
 35. Boneh D, Goh EJ, Nissim K. Evaluating 2-dnf formulas on ciphertexts. In *TCC*, Cambridge, MA, USA, 2005; 325–341.
 36. Bellare M, Waters B, Yilek S. Identity-based encryption secure against selective opening attack. In *TCC*, Providence, RI, USA, 2011; 235–252.