6-2018

Security and privacy in smart health: Efficient policy-hiding attribute-based access control

Yinghui ZHANG
Xi'an Institute of Posts and Telecommunications

Dong ZHENG
Xi'an Institute of Posts and Telecommunications

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

DOI: https://doi.org/10.1109/JIOT.2018.2825289

Published in IEEE Internet of Things Journal,
2018 April, Volume 3, Issue 1,
http://doi.org/10.1109/JIOT.2018.2825289

Security and Privacy in Smart Health: Efficient Policy-Hiding Attribute-Based Access Control

Yinghui Zhang, *Member, IEEE,* Dong Zheng, Robert H. Deng, *Fellow, IEEE*

***Abstract*—With the rapid development of the Internet of Things (IoT) and cloud computing technologies, smart health (s-health) is expected to significantly improve the quality of health care. However, data security and user privacy concerns in s-health have not been adequately addressed. As a well-received solution to realize fine-grained access control, ciphertext-policy attribute-based encryption (CP-ABE) has the potential to ensure data security in s-health. Nevertheless, direct adoption of the traditional CP-ABE in s-health suffers two flaws. For one thing, access policies are in cleartext form and reveal sensitive health-related information in the encrypted s-health records (SHRs). For another, it usually supports small attribute universe, which places an undesirable limitation on practical deployments of CP-ABE because the size of its public parameters grows linearly with the size of the universe. To address these problems, we introduce PASH, a privacy-aware s-health access control system, in which the key ingredient is a large universe CP-ABE with access policies partially hidden. In PASH, attribute values of access policies are hidden in encrypted SHRs and only attribute names are revealed. In fact, attribute values carry much more sensitive information than generic attribute names. Particularly, PASH realizes an efficient SHR decryption test which needs a small number of bilinear pairings. The attribute universe can be exponentially large and the size of public parameters is small and constant. Our security analysis indicates that PASH is fully secure in the standard model. Performance comparisons and experimental results show that PASH is more efficient and expressive than previous schemes.**

***Index Terms*—Smart health, Attribute-based encryption, Privacy protection, Decryption test, Large universe, Full security.**

I. Introduction

SMART health (s-health) is the context-aware augmentation of mobile health in smart cities, and it provides an opportunity for accurate and efficient prevention of various diseases and accidents [1]. As a kind of fundamental technologies in smart cities, the Internet of Things (IoT) has been widely applied to interconnect available medical resources and provide reliable and effective s-health services to the elderly and patients [2]. With the rapid development of IoT [3], [4] and cloud computing technologies [5], [6], cloud-based s-health is expected to provide desirable health care in the near future. However, s-health is still in its early stages and many concerns remain to be solved for practical applications [7]. In particular, data security and privacy issues have become the biggest concerns of people in s-health. For example, a patient usually expects that his s-health records (SHRs), such as blood pressure and pulse rate, can only be accessed by authorized professional health caregivers. Whereas, if traditional access control techniques are adopted, either data security is violated or only coarse-grained access policies are allowed.

Attribute-based encryption (ABE) is envisioned as a highly promising solution for realizing fine-grained access control [8]. ABE is divided into key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) [9]. In s-health, we focus on the latter which allows SHR owners to establish specific access policies. In a CP-ABE scheme, a user's secret key is associated with a set of attributes and each ciphertext is embedded with an access policy over attributes. Although the traditional CP-ABE schemes can be directly adopted to design fine-grained access control systems, it is still necessary to simultaneously address the issues of *policy hiding*, *decryption test*, *large universe*, *full security* and *policy expressiveness* in CP-ABE to ensure its secure and efficient applications in s-health.

Policy Hiding. In the traditional CP-ABE schemes, an access policy is an access structure, which is expressed in terms of user attributes and is sent along with a ciphertext (e.g., an encrypted SHR) explicitly. Anyone who obtains the ciphertext knows the associated access policy. Therefore, the traditional CP-ABE is inappropriate for s-health since access policies usually contain sensitive information. Let's consider a s-health cloud storage scenario, as shown in Fig. 1. A hospital, which manages SHRs on behalf of its patients, outsources its encrypted SHRs to a s-health cloud for secure storage and sharing among health caregivers. Suppose the hospital encrypts a SHR using CP-ABE under an access policy "(SSN: *123-260-6* AND Status: *Normal*) OR (Affiliation: *City Hospital* AND Department: *Cardiologist*)", and then outsources the ciphertext together with the access policy to the cloud. The access policy specifies that the SHR can only be accessed by a *Cardiologist* in *City Hospital* or by a patient with social security number (SSN) *123-260-6* and *Normal* registration status. In Fig. 1, the data user Alice is capable of accessing the SHR while Bob is not an authorized data user. Although Bob comes from the *City Hospital* and the registration status is *Normal*, his department and SSN cannot match the access policy. Obviously, if the traditional CP-ABE scheme is used, everyone including the s-health cloud service provider can learn the access policy and infer that a normal user in the s-health system with social

Y. Zhang is with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China; the State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China; and the School of Information Systems, Singapore Management University, Singapore (Email: yhzhaang@163.com).

D. Zheng is with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China (Email: zhengdong@xupt.edu.cn).

R. H. Deng is with the School of Information Systems, Singapore Management University, Singapore (E-mail: robertdeng@smu.edu.sg).

Y. Zhang is the corresponding author.

security number *123-260-6* is suffering a heart problem. This violates user's privacy and shows the importance of hiding access policies in CP-ABE.

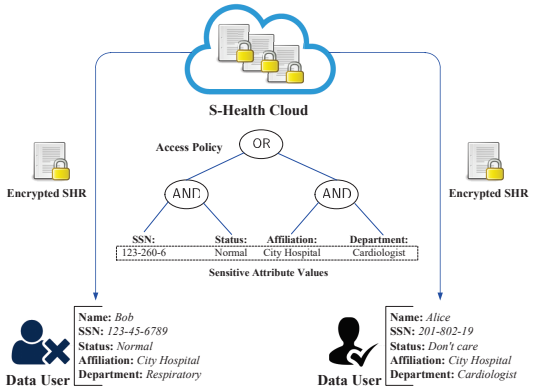


Fig. 1. An example of s-health cloud storage.

Policy hiding takes two forms, it can be either full hiding or partial hiding. In CP-ABE schemes with access policies fully hidden, none of the attributes in the access policy are revealed. Although a CP-ABE scheme with fully hidden policies can be built from attribute-hiding Inner-product Predicate Encryption (IPE) [10], the policies can only be threshold policies which are far less expressive than Linear Secret Sharing Scheme (LSSS) policies. A KP-ABE scheme with fully hidden policies can also be constructed from IPE, but it results in a superpolynomial blowup in size for arbitrary formulas [11]. For s-health, CP-ABE is more suitable than KP-ABE because it allows data owners to choose access policies themselves. With the goal of having a tradeoff between fully hidden policies and efficiency of CP-ABE, CP-ABE schemes with partially hidden policies have attracted many attentions recently. In CP-ABE schemes with partially hidden access policies, only sensitive attribute values are hidden and attribute names are sent along with ciphertexts. Note that partial hiding CP-ABE constructions have better performance and are adequate to protect users' privacy. In the above example of s-health cloud storage, if the hospital uses a partial hiding CP-ABE scheme to encrypt a SHR, the partially hidden access policy: "(SSN: $*$ AND Status: $*$) OR (Affiliation: $*$ AND Department: $*$)", is attached with the encrypted SHR while the sensitive attribute values, "*123-260-6*", "*Normal*", "*City Hospital*" and "*Cardiologist*", are hidden from the public including the cloud service provider.

Decryption Test. Recall that in CP-ABE a user's secret key is associated with a set of attributes, such as "(Doctor: Cardiologist; Hospital: City Hospital)". Given a ciphertext and the associated access policy, the decryption is successful only if the user's set of attributes satisfies the access policy. A decryption test is the testing whether the user's set of attributes satisfies the access policy of the ciphertext before the full decryption. In the existing policy-hiding CP-ABE schemes, however, a user usually has to decrypt and then decide whether or not his attributes satisfy the hidden policy in the ciphertext, and this decryption-testing process is repeated until a match between attributes and access policy is found or all the possible tests have been performed. Hence, the computation cost in decryption is very high. Some schemes add a decryption test before the full decryption. However, their efficiency of attribute matching needs to be improved because the computation overhead grows linearly with the complexity of the access policy. Accordingly, it is desirable for users to efficiently decide before the full decryption whether his attributes match the hidden policy in the ciphertext.

Large Universe. ABE can be classified into small universe ABE and large universe ABE. In an ABE scheme supporting small universe, the system public parameter size often depends on the number of attributes in the system, and hence the scale of the attribute universe is polynomially bounded in the security parameter. In the case of large universe, the attribute universe is allowed to exponentially scale in size, which is a desirable feature. In fact, it is challenging to achieve the large universe construction without imposing restrictions on the expressiveness of the policies. If the designer of an ABE system intends to remove the random oracle, he has to specify a bound for the expressiveness at the setup phase, such as the size of the attribute universe and the bound on the policies. If the bound is too small, the system might exhaust its functionality and will have to be thoroughly rebuilt. In a s-health system using ABE for secure and fine-grained data sharing, as its functionality expands and the number of user increases, a large number of new attributes have to be added to the system. If this number exceeds the initial bound set for ABE with small universe construction, the authority would have to re-deploy the system and re-encrypt all its data which will incur a very high cost. On the other hand, if the initial bound is too large, the increased size of public parameters will impose an unnecessary performance burden on the system.

Full Security. Most existing ABE constructions are proven secure in the selective model, which is a weaker security model compared with full security. In the selective model, the attacker must declare what the challenge ciphertext is, before seeing the system public parameters. Therefore, fully secure constructions are more desirable in cloud access control.

Besides the issues mentioned above, *expressiveness* is another important issue in ABE based access control systems. Many existing CP-ABE schemes with partially hidden policies only support restricted access policies, which can be expressed as AND gates on two/multi-valued attributes with wildcards. Access policies of the form LSSS are more expressive, which can be obtained from any monotonic boolean formula [12]. Specifically, the boolean formula can be represented as an access tree of ℓ nodes, which can also be converted to an LSSS matrix of ℓ rows [13].

To our knowledge, provably secure policy-hiding CP-ABE schemes supporting decryption test can be founded in [14], [15]. However, the scheme in [14] is proven selectively secure and only allows AND policies. Although the scheme [15] supports LSSS policies and achieves full security, the decryption test still suffers an efficiency drawback in that the number of expensive bilinear pairings is linearly proportional to the complexity of access policies. Besides, both schemes are not large universe constructions.

A. Our Contribution

In this paper, we efficiently address both data security and user privacy issues in s-health by introducing PASH, a privacy-aware s-health access control system. In PASH, we focus on the important issues mentioned above, i.e., attribute privacy, decryption test efficiency, large universe, expressiveness and full security. We simultaneously solve these issues by proposing a large universe CP-ABE scheme with partially hidden access policies, denoted as PH-CP-ABE, which is the main building block of PASH. Our rigorous security proofs and comprehensive performance comparisons based on experimental results indicate that PASH is fully secure and very efficient. Specifically, PASH is characterized by the following attractive features:

- *Attribute privacy*. In PASH, specific and sensitive attribute values in access policies are hidden in SHR ciphertexts and only generic attribute names are sent along with ciphertexts.
- *Decryption test efficiency*. In order to improve the decryption efficiency, we add a decryption test before full decryption by generating redundant ciphertext components which are half of that in previous work. The decryption test is very efficient because it only involves a small number of bilinear pairing operations in attribute matching detection.
- *Large universe*. PASH imposes no limitations on the attribute universe in the sense that the universe can be exponentially large and the size of public parameters is constant, while most of the previous schemes have public parameters linearly scaling with the size of the universe.
- *Expressiveness and full security*. PASH can handle any access policies that can be expressed as LSSS. Also, we use the dual system encryption methodology [11] to prove the full security of PASH in the standard model under static assumptions.

B. Related Work

Nowadays, an increasing number of people expect to obtain more proactive, quality and comprehensive health care. To achieve this, s-health is indispensable because it plays an important role in the early-stage diagnosis based on the real-time and long-term monitoring. As we know, many promising technologies are involved in s-health, such as wireless body area networks (WBAN), IoT, wireless communication, and cloud computing. Yan et al. [16] proposed a wearable wireless sensor network for anomaly detections of health conditions. In particular, there has been many research focusing on WBAN reliability [17], [18] and wireless handover authentication [19], [20]. Xu et al. [21] proposed an IoT-based system for emergency medical services, which can collect and interoperate IoT data flexibly. To ensure the quality of cloud-based service, Zheng et al. [22] proposed a mixed approach for cloud service negotiation. Yin et al. [2] presented an overview of the advancement of IoT in healthcare systems. At the same time, they pointed out that IoT-based applications are extremely vulnerable and intensive research is needed in the areas of security and privacy protection. In order to understand the development of IoT, Xu et al. [3] and Li et al. [4] pointed out the research trends and challenges of IoT in industries and future networks. Besides, substantial research has been done on IoT architectures [23], [24], applications of cloud computing [5], [6], [7] and cloud security [25], [26], [27].

Nevertheless, most of the above IoT and cloud computing technologies are not focusing on concrete data security and privacy issues in s-health, which urgently needs to be addressed for the wide public acceptance of s-health services. As an attractive primitive, ABE presents a promising solution to data security in s-health. The first KP-ABE construction [9] was proposed based on monotonic access policies. In the generic group model, the first CP-ABE scheme was proposed by Bethencourt et al. [28]. Cheung et al. [29] proposed another CP-ABE scheme and proved its security in the standard model. The scheme allows access policies of the form AND over different attributes. In addition, many other ABE schemes have been proposed, focusing on hierarchical structure [30], efficient construction [31], [32], [33], revocation [34], [35] and traceability [36], [37]. However, these schemes do not consider the abuse of users' attribute privacy and hence cannot be directly used in s-health.

In order to protect users' attribute privacy, policy-hiding CP-ABE schemes have been studied [14], [15], [38], [39], [40], [41], [42], [43], [44], [45], [46]. The notion of partially hidden CP-ABE was introduced by Nishide et al. [41], where AND gate policies are admissible. Similar schemes can also be found in [42], [43]. All these schemes are proven secure in the selective model. Based on the CP-ABE scheme [32] and the Garbled bloom filter [47], Yang et al. [44] realized a privacy-preserving access control mechanism. However, privacy protection is not formalized in its security proofs. Lewko et al. [11] proposed the first fully secure CP-ABE scheme, which is proven secure based on the dual system encryption methodology [48] in the standard model. Okamoto et al. [49] presented a fully secure functional encryption scheme, which covers KP-ABE and CP-ABE schemes with non-monotone access policies. Lai et al. [45] and Jin et al. [46] proposed fully secure CP-ABE schemes with partially hidden access policies. However, these schemes only support AND gate policies. Note that, although users' privacy is preserved in the above policy-hiding CP-ABE constructions, the decryption test is not considered. Partially hidden CP-ABE schemes in [14], [15], [38], [39] further realize decryption tests. However, the schemes in [14], [38], [39] are proven selectively secure and only allow AND gate policies. Although the scheme [15] supports LSSS policies and achieves full security, the decryption test is inefficiency in that the number of bilinear pairing operations linearly increases with the complexity of access policies. In addition, these schemes cannot support large universe. Lewko et al. [13] proposed the first unbounded KP-ABE scheme, which can support a large attribute universe in composite order groups. Furthermore, Rouselakis et al. [50] proposed both KP-ABE and CP-ABE schemes supporting large universe in groups of prime orders. Most recently, Cui et al. [40] proposed a partially hidden CP-ABE scheme supporting LSSS policies and large universe based on [50], however, the scheme is proven secure in the random oracle model and cannot achieve full security. Especially, the scheme fails to support decryption test before full decryption and hence it is inefficiency even if

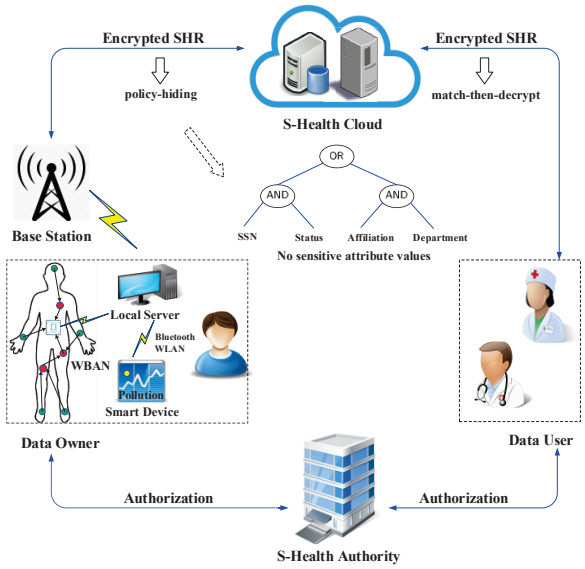


Fig. 2. Architecture of the privacy-aware s-health access control system.

it is designed based on groups of prime order.

C. Organization

The rest of the paper is organized as follows. We present the system architecture and adversary model in Section II. Some preliminaries are given in Section III. In Section IV, we describe a formalized definition and a security model of PH-CP-ABE. The proposed PASH is presented in Section V. Our security and performance analysis is given in Section VI. Finally, concluding remarks are made in Section VII. To make the paper complete and easy to follow, we present detailed security proofs in the APPENDIX.

II. System Architecture and Design Goals

A. System Architecture

As depicted in Fig. 2, four generic entities, S-Health Authority (SHA), S-Health Cloud (SHC), Data Owner (DO), and Data User (DU) are involved in the privacy-aware s-health access control system, which are described below.

- SHA is in charge of the system initialization and user authorization. It is trustworthy and grants fine-grained access privileges to DUs based on their attributes.
- SHC has abundant storage capacity and stores encrypted SHRs and their partially hidden access policies from DO.
- DO owns and manages SHRs, and outsources them to SHC in the form of ciphertexts for health care. DO can be a patient or a hospital which manages SHRs on behalf of its patients. DO has local servers, smart devices, and a WBAN which consists of a number of wearable or implantable sensors and a controller. SHRs from sensors or other smart devices are encrypted by the WBAN controller or the local server, and then are sent to SHC with the help of base stations for sharing with DU. DO is responsible for defining and enforcing access policies for encrypted SHRs.
- DU is a SHR user, such as a doctor or a medical researcher, who needs to access the encrypted SHRs on SHC. Every DU possesses a set of attributes and a secret key associated with the set of attributes. If a DU's set of attributes matches the underlying hidden access policy attached to an encrypted SHR, then he can successfully pass the attribute matching and decrypt the encrypted SHR.

We now give an overview of the proposed PASH system.

- **Initialization.** SHA initializes the system by generating system public parameters and a master key. Each user can get the system public parameters.
- **Authorization.** SHA grants access rights to DU by issuing a secret key based on the DU's set of attributes. Similarly, DO can also get his own secret key if necessary.
- **Privacy-Aware SHR Outsourcing.** Given a SHR, DO specifies an access policy and uses it to encrypt the SHR. The encrypted SHR is outsourced to SHC for health care while the access policy is partially hidden.
- **Privacy-Aware SHR Access.** After obtaining an encrypted SHR from SHC, DU first uses his secret key to check whether his attributes match the underlying access policy, and then decrypt the encrypted SHR only if the matching is successful.

B. Adversary Model and Design Goals

In PASH, SHC is assumed to be honest-but-curious. Exactly, it honestly performs the various procedures of PASH according to system specifications while trying to learn secret information from encrypted SHRs as much as possible. Both full data security and attribute privacy protection are taken into consideration, where "full" means the adversary does not need to declare which SHR is his target before getting system public parameters. Note that the adversary can be either a malicious DU or a combination of multiple DUs and SHC. As for data security, the adversary can eavesdrop encrypted SHRs transmitted on public channels and try to access SHRs it is not authorized to access. With respect to attribute privacy protection, the adversary aims to extract information on sensitive attribute values from encrypted SHRs. Concretely, we consider security requirements as below.

- *SHR Confidentiality.* The outsourced SHRs are private and sensitive for DOs, and hence should be protected from unauthorized access.
- *Collusion-Resistance.* Different users and SHC may collude by combining their secret keys to read SHRs anyone of them is not authorized to access. These collusion attacks must be prevented for data security.
- *Attribute Privacy Protection.* In PASH, specific attribute values associated with access policies are sensitive, which should be hidden in encrypted SHRs for privacy protection.

Besides, we aim to address the following performance issues.

- *Efficient Decryption Test.* For the sake of practicality, it is desirable for DUs in PASH to efficiently check before full decryption whether or not his attributes match the access policy associated with the encrypted SHR.

- *Expressive Access Policy.* In order to realize fine-grained access control on SHRs in PASH, the access policy should be as expressive as possible.
- *Large Universe.* As mentioned before, it is desirable for PASH to support an attribute universe of exponential scale while keeping the expressiveness of access policies.

III. Preliminaries

In Table I, we summarize the notations used in PASH.

TABLE I
Notation Description.

Notations	Descriptions
$a \in_R A$	The element a is randomly chosen from the set A.
$\|A\|$	The number of elements in the set A.
$\mathbb{G}$,$\mathbb{G}_T$	Two cyclic multiplicative groups.
$\mathbb{G}_{p_i}$	A subgroup of $\mathbb{G}$ with prime order p_i.
$\mathsf{SK}_{\mathcal{S}}$	A secret key associated with an attribute set $\mathcal{S}$.
$\mathcal{S} = (\mathcal{I}_S, S)$	$\mathcal{I}_S$ denotes the attribute name index,
	S is the attribute value set.
$\mathsf{CT}_{\mathbb{A}}$	A CP-ABE ciphertext associated with an access policy $\mathbb{A}$.
$\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$	$\mathbf{A}$ is the access policy matrix,
	ρ maps a row of $\mathbf{A}$ to an attribute name index,
	$\mathcal{T}$ is an attribute value set.
$\mathcal{I}$	A minimum authorized set of $(\mathbf{A}, \rho)$.
$\mathbf{I}_{\mathbf{A},\rho}$	The set of $\mathcal{I}$.

A. Cryptographic Background

Definition 1 (Composite Order Bilinear Groups): Composite order bilinear groups are widely used in IBE and ABE systems, which are first introduced in [51]. We denote by $\mathcal{G}$ a group generator, which takes a security parameter λ as inputs and outputs a description of a bilinear group $\mathbb{G}$. We define the output of $\mathcal{G}$ as $(N, p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, \hat{e})$ with $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3} \times \mathbb{G}_{p_4}$, where p_1, p_2, p_3, p_4 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1p_2p_3p_4$, and $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map satisfying:

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

Assume group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map $\hat{e}$ are computable in polynomial time with respect to λ. Let $\mathbb{G}_{p_i}$ be the subgroup of order p_i in $\mathbb{G}$ for $1 \le i \le 4$. Note that for any $X_i \in \mathbb{G}_{p_i}$ and $X_j \in \mathbb{G}_{p_j}$, $\hat{e}(X_i, X_j) = 1$ holds for $i \neq j$. The subgroups are said to be "orthogonal" to each other.

B. Access Structures and Linear Secret Sharing Schemes

Definition 2 (Access Structures [12]): Let $\mathcal{U}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\mathcal{U}}$ is monotone if $\forall B \in \mathbb{A}$ and $C \in 2^{\mathcal{U}}$: if $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (resp. monotone access structure) on $\mathcal{U}$ is a collection (resp. monotone collection) $\mathbb{A}$ of non-empty subsets of $\mathcal{U}$, *i.e.*, $\mathbb{A} \subseteq 2^{\mathcal{U}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, otherwise, the sets are called unauthorized sets.

Definition 3 (Linear Secret Sharing Schemes (LSSS) [12]): Let $\mathcal{U}$ be the attribute universe, where each attribute includes two parts: attribute name and its values. Each attribute has multiple values. An LSSS involves $(\mathbf{A}, \rho)$ on $\mathcal{U}$, where $\mathbf{A}$ is an $\ell \times n$ matrix over $\mathbb{Z}_p$ which is called the share-generating matrix and ρ maps a row of $\mathbf{A}$ into an attribute name index. An LSSS consists of two algorithms:

- **Share**$((\mathbf{A}, \rho), s)$: This algorithm is used to share a secret value s based on $\mathbf{A}$. Considering a vector $v = (s, y_2, ..., y_n)^{\mathrm{T}}$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $y_2, ..., y_n \in_R \mathbb{Z}_p$, then $\lambda_x = A_x \cdot v$ is a share of the secret s corresponding to the attribute name indexed by $\rho(x)$.
- **Reconstruction**$(\lambda_1, ..., \lambda_\ell, (\mathbf{A}, \rho))$: This algorithm is used to reconstruct s from secret shares. Let $\mathcal{P}$ be any authorized set and $\mathcal{I} = \{i | \rho(i) \in \mathcal{P}\} \subseteq \{1, 2, ..., \ell\}$. Then there exists coefficients $\{\omega_i \in \mathbb{Z}_p\}_{i \in \mathcal{I}}$ such that $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, ..., 0)$, thus we have $\sum_{i \in \mathcal{I}} \omega_i \lambda_i = s$.

We say that $\mathcal{I} \subseteq \{1, 2, \ldots, \ell\}$ satisfies $(\mathbf{A}, \rho)$ if there exists constants $\{\omega_i\}_{i \in \mathcal{I}}$ such that $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, ..., 0)$. A subset $\mathcal{I}$ of $\{1, 2, \ldots, \ell\}$ is said to be a minimum authorized set of $(\mathbf{A}, \rho)$ if $\mathcal{I}$ satisfies $(\mathbf{A}, \rho)$ and any $\mathcal{I}' \subset \mathcal{I}$ does not satisfy $(\mathbf{A}, \rho)$. We define $\mathbf{I}_{\mathbf{A},\rho}$ as the set of subsets of $\{1, 2, \ldots, \ell\}$ that are minimum authorized sets of $(\mathbf{A}, \rho)$.

Like [11], [15], [52], we will employ LSSS matrices over $\mathbb{Z}_N$, where N is a composite number. Our proposed scheme allows arbitrary monotone access structures. Suppose a user has a secret key associated with an attribute set $\mathcal{S} = (\mathcal{I}_S, S)$, where $\mathcal{I}_S \subseteq \mathbb{Z}_N$ denotes the attribute name index and the corresponding attribute value set is $S = \{s_i\}_{i \in \mathcal{I}_S}$. We use $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$ to represent the adopted access structure, where $\mathcal{T} = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)})$ is the attribute value set associated with $(\mathbf{A}, \rho)$. We also say that $\mathcal{S}$ matches $\mathbb{A}$ if there exists an $\mathcal{I} \subseteq \{1, 2, \ldots, \ell\}$ satisfying $(\mathbf{A}, \rho)$, $\{\rho(i) | i \in \mathcal{I}\} \subseteq \mathcal{I}_S$ and $s_{\rho(i)} = t_{\rho(i)}$ for each $i \in \mathcal{I}$.

C. Review of Lai et al.'s Scheme

For easy understanding of our ideas in Section V-A, we describe the CP-ABE scheme due to Lai et al. [15] as below.

- $\mathsf{Setup}(1^\lambda)$: It takes a security parameter λ as input and runs the group generator $\mathcal{G}(1^\lambda)$ to get $(N, p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, \hat{e})$. Then it sets the attribute universe as $U = \mathbb{Z}_N$, uniformly chooses $\alpha, a \in_R \mathbb{Z}_N$, $g, h, u_1, u_2, \cdots, u_N \in_R \mathbb{G}_{p_1}$, $X_3 \in_R \mathbb{G}_{p_3}$, $Z, X_4 \in_R \mathbb{G}_{p_4}$ and computes $Y = \hat{e}(g, g)^\alpha$, $H = hZ$. The system public parameters are published as $\mathsf{PK} = (N, g, u_1, u_2, \cdots, u_N, g^a, Y, H, X_4)$, and the master key is $\mathsf{MK} = (\alpha, h, X_3)$.
- $\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MK}, \mathcal{S})$: Let $\mathcal{S} = (\mathcal{I}_S, S)$ with $\mathcal{I}_S \subseteq \mathbb{Z}_N$ and $S = \{s_i\}_{i \in \mathcal{I}_S}$. It chooses $t \in_R \mathbb{Z}_N$, $R, R', R_i \in_R \mathbb{G}_{p_3}$ for $i \in \mathcal{I}_S$, and outputs the secret key $\mathsf{SK}_{\mathcal{S}} = (\mathcal{S}, K, K', \{K_i\}_{i \in \mathcal{I}_S})$, where
$$K = g^\alpha g^{at} R, K' = g^t R', K_i = (u_i^{s_i} h)^t R_i.$$
- $\mathsf{Encrypt}(\mathsf{PK}, M, \mathbb{A})$: $M \in \mathbb{G}_T$ and $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, where $\mathbf{A}$ is an $\ell \times n$ matrix, ρ is a map from each row A_x of $\mathbf{A}$ to an attribute name, and $\mathcal{T} = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$. The encryption algorithm chooses two vectors $v, v' \in_R \mathbb{Z}_N^n$ with $v = (s, v_2, \ldots, v_n)$ and $v' = (s', v'_2, \ldots, v'_n)$. It also uniformly chooses $r_x, r'_x \in_R \mathbb{Z}_N$ and $Z_{1,x}, Z'_{1,x}, Z_{2,x}, Z'_{2,x} \in_R \mathbb{G}_{p_4}$, for $1 \le x \le \ell$. Then it calculates the ciphertext as

$$\mathsf{CT}_{\mathbb{A}} = ((\mathbf{A}, \rho), \tilde{C}_1, C'_1, \{C_{1,x}, D_{1,x}\}_{1 \le x \le \ell}, \tilde{C}_2, C'_2, \{C_{2,x}, D_{2,x}\}_{1 \le x \le \ell}),$$

where

$$\tilde{C}_1 = M \cdot Y^s, C'_1 = g^s, C_{1,x} = g^{aA_x \cdot v} (u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x} Z_{1,x}, D_{1,x} = g^{r_x} Z'_{1,x},$$

$\tilde{C}_2 = Y^{s'}, C_2' = g^{s'}, C_{2,x} = g^{aA_x \cdot v'}(u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x'} Z_{2,x}, D_{2,x} = g^{r_x'} Z_{2,x}'$.

- $\mathsf{Decrypt}(\mathsf{PK}, \mathsf{CT}_{\mathbb{A}}, \mathsf{SK}_{\mathcal{S}})$: Given $\mathsf{CT}_{\mathbb{A}}$ and $\mathsf{SK}_{\mathcal{S}}$, it first calculates $\mathbf{I}_{\mathbf{A},\rho}$ from $(\mathbf{A}, \rho)$. Then it checks if there exists a subset $\mathcal{I} \in \mathbf{I}_{\mathbf{A},\rho}$ that satisfies $\{\rho(i) | i \in \mathcal{I}\} \subseteq \mathcal{I}_S$ and

$$\tilde{C}_2 = \frac{\hat{e}(C_2', K)}{\prod_{i \in \mathcal{I}} \left(\hat{e}(C_{2,i}, K')\hat{e}(D_{2,i}, K_{\rho(i)})\right)^{\omega_i}},$$

where $\sum_{i \in \mathcal{I}} \omega_i A_i = (1, 0, \ldots, 0)$ for some constants $\{\omega_i\}_{i \in \mathcal{I}}$. If no such $\mathcal{I}$ exists, it outputs $\perp$ to indicate that $\mathcal{S}$ does not satisfy the partially hidden $\mathbb{A}$. Otherwise, it computes

$$\frac{\hat{e}(C_1', K)}{\prod_{i \in \mathcal{I}} \left(\hat{e}(C_{1,i}, K')\hat{e}(D_{1,i}, K_{\rho(i)})\right)^{\omega_i}} = Y^s,$$

and outputs $M = \tilde{C}_1 / Y^s$.

IV. Formal Definition and Security Model

A. Definition of PH-CP-ABE with Decryption Test

A PH-CP-ABE scheme with decryption test consists of the following four algorithms: Setup, KeyGen, PH.Encrypt and PH.Decrypt.

- $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{PK}, \mathsf{MK})$: The setup algorithm takes a security parameter λ as input. It outputs the system public parameters PK and a master key MK.
- $\mathsf{KeyGen}(\mathsf{PK}, \mathsf{MK}, \mathcal{S}) \rightarrow \mathsf{SK}_{\mathcal{S}}$: The key generation algorithm takes as inputs the system public parameters PK, the master key MK and a set of attributes $\mathcal{S}$. It outputs a secret key $\mathsf{SK}_{\mathcal{S}}$ associated with $\mathcal{S}$.
- $\mathsf{PH.Encrypt}(\mathsf{PK}, M, \mathbb{A}) \rightarrow \mathsf{CT}_{\mathbb{A}}$: The encryption algorithm takes as inputs the system public key PK, a message M and an access structure $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$. It ouputs a ciphertext $\mathsf{CT}_{\mathbb{A}}$ of M with respect to $\mathbb{A}$. *In a PH-CP-ABE scheme, it is required that the attribute value set $\mathcal{T}$ in $\mathbb{A}$ is hidden and it is not explicitly included in* $\mathsf{CT}_{\mathbb{A}}$.
- $\mathsf{PH.Decrypt}(\mathsf{PK}, \mathsf{CT}_{\mathbb{A}}, \mathsf{SK}_{\mathcal{S}}) \rightarrow M$ or $\perp$: The decryption algorithm takes as inputs the system public key PK, the ciphertext $\mathsf{CT}_{\mathbb{A}}$ of a message M with respect to an access structure $\mathbb{A}$, and the secret key $\mathsf{SK}_{\mathcal{S}}$ associated with an attribute set $\mathcal{S}$. It outputs M if $\mathcal{S}$ satisfies $\mathbb{A}$, and the error symbol $\perp$ otherwise. Concretely, two phases, attribute matching detection (i.e., decryption test) and decryption phase (i.e., full decryption), are involved.
  - Matching Phase: It returns $\perp$ to terminate the decryption algorithm with overwhelming probability if $\mathcal{S}$ does not satisfy the partially hidden $\mathbb{A}$. Otherwise, it ends by initiating the decryption phase.
  - Decryption Phase: It returns M.

B. Security Model

In this section, we give the security model for PH-CP-ABE. It is described as a security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$. The game proceeds as follows:

- **Setup.** The challenger $\mathcal{B}$ runs $(\mathsf{PK}, \mathsf{MK}) \leftarrow \mathsf{Setup}(1^\lambda, U)$. It gives the system public key PK to $\mathcal{A}$ and keeps MK secret.
- **Phase 1.** The adversary $\mathcal{A}$ adaptively issues a polynomially bounded number of queries to the following key generation oracle.
  - O_{KeyGen}: The adversary $\mathcal{A}$ submits an attribute set $\mathcal{S}$. The challenger $\mathcal{B}$ runs $\mathsf{SK}_{\mathcal{S}} \leftarrow \mathsf{KeyGen}(\mathsf{PK}, \mathsf{MK}, \mathcal{S})$, and gives $\mathcal{A}$ the secret key $\mathsf{SK}_{\mathcal{S}}$.
- **Challenge.** Once the adversary $\mathcal{A}$ decides that **Phase 1** is over, it submits to the challenger $\mathcal{B}$ two messages M_0, M_1 of equal length and two access structures $\mathbb{A}_1 = (\mathbf{A}, \rho, \mathcal{T}_0)$, $\mathbb{A}_2 = (\mathbf{A}, \rho, \mathcal{T}_1)$ with the restriction that $\mathbb{A}_1$ and $\mathbb{A}_2$ cannot be satisfied by any of the queried attribute sets in **Phase 1**. The challenger $\mathcal{B}$ flips a random coin $\beta \in \{0, 1\}$, sets $\mathsf{CT}_{\mathbb{A}_\beta} \leftarrow \mathsf{PH.Encrypt}(\mathsf{PK}, M_\beta, \mathbb{A}_\beta)$ and sends $\mathsf{CT}_{\mathbb{A}_\beta}$ to $\mathcal{A}$ as its challenge ciphertext.
- **Phase 2.** The adversary continues to adaptively query the challenger for secret keys corresponding to sets of attributes with the restriction that none of these satisfies $\mathbb{A}_1$ and $\mathbb{A}_2$.
- **Guess:** The adversary $\mathcal{A}$ outputs a guess bit $\beta' \in \{0, 1\}$ and wins the game if $\beta' = \beta$.

The advantage of the adversary $\mathcal{A}$ in this game is defined as $\left|\Pr[\beta' = \beta] - \frac{1}{2}\right|$, where the probability is taken over the random bits used by the adversary $\mathcal{A}$ and the challenger $\mathcal{B}$.

Definition 4: A CP-ABE scheme with partially hidden access structures is fully secure if all polynomial time adversaries have at most a negligible advantage in the security game.

V. PASH: Privacy-Aware s-Health Access Control System

A. Challenges and Main Idea

In real life, attribute values always contain more sensitive information than the generic attribute names. Due to this observation, CP-ABE schemes with partially hidden access policies were proposed [14], [15], [40], [41], [42], in which the schemes [14], [15] support decryption tests. As shown in [14], adding an efficient decryption test before the full decryption is an important way to improve the computation efficiency, especially in the cloud-based s-health scenario where the cloud server may search and send a lot of CP-ABE ciphertexts to a user. The CP-ABE scheme [15] supports the decryption test, but the test is inefficient because the number of expensive pairing operations in the test grows linearly with the complexity of the access policy.

1) Challenge I: **How to realize a large universe construction?** In [15], we found that the public parameters $u_1, u_2, \cdots, u_n$ could be eliminated. In fact, u_i is used in the key generation algorithm for computing a secret key component $K_i = (u_i^{s_i} h)^t R_i$, where s_i is the user's attribute value and other parameters are given or randomly chosen from corresponding groups. In the encryption algorithm, $u_{\rho(x)}$ is used for computing ciphertext components $C_{1,x} = g^{aA_x \cdot v}(u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x} \cdot Z_{1,x}$ and $C_{2,x} = g^{aA_x \cdot v'}(u_{\rho(x)}^{t_{\rho(x)}} H)^{-r_x'} \cdot Z_{2,x}$, where $t_{\rho(x)}$ is an attribute value implicitly specified for the xth row of the matrix in the LSSS policy and other parameters are given or randomly chosen from corresponding groups. Because the attribute values are not sent along with ciphertexts and random values t, r_x, r_x' are used in the generation of $K_i, C_{1,x}, C_{2,x}$, we change $u_i^{s_i}$ and $u_{\rho(x)}^{t_{\rho(x)}}$ to g^{s_i} and $g^{t_{\rho(x)}}$, respectively. Hence, the public parameter size becomes constant and it means a large universe construction.

2) Challenge II: **How to significantly improve the decryption test efficiency?** In [15], a ciphertext includes two parts, one is an encryption of the message

and the other is redundant. The redundant part consists of $\tilde{C}_2 = \hat{e}(g,g)^{\alpha s'}, C_2' = g^{s'}, C_{2,x} = g^{aA_x \cdot v'}(u_{\rho(x)}{}^{t_{\rho(x)}} H)^{-r_x'} \cdot Z_{2,x}, D_{2,x} = g^{r_x'} \cdot Z_{2,x}'$ and it is used for decryption test. In each decryption test, the number of pairing operations is not constant but proportional to the complexity of the ciphertext policy. It can be easily noted that this is caused by $\prod_x \hat{e}(D_{2,x}, K_{\rho(x)})^{\omega_x}$. Obviously, $K_{\rho(x)}$ is the only key component associated with attribute values. Therefore, the only solution is to update $D_{2,x}$. In the proposed scheme, we delete C_2' from the ciphertext and change $D_{2,x} = g^{r_x'} \cdot Z_{2,x}'$ to $D_2 = g^s \cdot Z_2'$, where s is a random value from corresponding group. Besides, $C_{2,x}$ is changed to $C_{2,x} = g^{aA_x \cdot v'}(g^{t_{\rho(x)}} H)^{-s} \cdot Z_{2,x}$. In this case, $\prod_x \hat{e}(D_{2,x}, K_{\rho(x)})^{\omega_x}$ can be computed by $\hat{e}(D_2, \prod_x K_{\rho(x)}^{\omega_x})$ and the decryption test efficiency is significantly improved. Note that *the size of the ciphertext is also reduced by ℓ group elements*, where ℓ is the number of rows of the policy matrix.

3) Challenge III: **How to ensure correctness and security?** Based on the property of bilinear pairings, the correctness is not affected by the above updating. However, it is not easy to get the security results by following the security proof in [15]. For the simplicity of the description, we use some new symbols in our scheme, where Δ indicates the parameter is used for decryption test. Please refer to Section V-B for more details of the following parameters used in our PASH. Now, we explain the challenges in our security proofs. First, in the ciphertext, $\tilde{C}_\Delta = Y^{s'}, \hat{C}_\Delta = g^{s'} Z_\Delta, C_{\Delta,x} = g^{aA_x \cdot v'}(g^{t_{\rho(x)}} H)^{-s'} Z_{\Delta,x}$, where s' is shared by all components and it is the first element of v. Hence, the related exponents in the security proofs cannot be chosen as random elements and they must keep a consistency, which is different from the proof in [15]. Second, in a semi-functional key of type 1, the exponent z_i must be the same as the one used in the semi-functional ciphertext. Loosely speaking, we address these two challenges by the "orthogonal" subgroups and additionally choosing a vector v_Δ with 0 as the first element to adjust exponents and keep them consistent in the challenge phase. Third, in the ciphertext, $\hat{C}_\Delta = g^{s'} Z_\Delta$, $\hat{C}_1 = g^s$, where g and Z_Δ are chosen from different subgroups. We keep Z_Δ in $\hat{C}_\Delta$ but not in $\hat{C}_1$. According to the property of "orthogonal" subgroups, if we remove Z_Δ from $\hat{C}_\Delta$ and set $\hat{C}_\Delta = g^{s'}$, the decryption algorithm proceeds correctly. However, in this case, any adversary may use $\hat{C}_\Delta$ to guess some attribue values because s' is also used in $C_{\Delta,x} = g^{aA_x \cdot v'}(g^{t_{\rho(x)}} H)^{-s'} Z_{\Delta,x}$ and $\hat{e}(g, Z_{\Delta,x}) = 1$.

In summary, the differences between our policy-hiding CP-ABE scheme and the scheme [15] are threefold. First, the proposed scheme is a large universe construction. Second, we reduce the number of expensive pairing operation in a decryption test from $|\mathcal{I}| + 2$ to 2, where $|\mathcal{I}|$ is specified by the complexity of the access policy. Third, the ciphertext is reduced by ℓ group elements.

B. Design Details of PASH

1) Initialization: SHA first takes a security parameter λ as input and runs the group generator $\mathcal{G}(1^\lambda)$ to get $(N, p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, \hat{e})$. Then SHA sets the attribute universe as $U = \mathbb{Z}_N$ and performs the setup algorithm as below.

- Setup(1^λ): SHA uniformly chooses $\alpha, a \in_R \mathbb{Z}_N$, $g, h \in_R \mathbb{G}_{p_1}$, $X_3 \in_R \mathbb{G}_{p_3}$, $Z, X_4 \in_R \mathbb{G}_{p_4}$ and computes $Y = \hat{e}(g,g)^\alpha, H = hZ$. The system public parameters are published as $\mathsf{PK} = (N, g, g^a, Y, H, X_4)$, and the master key is $\mathsf{MK} = (\alpha, h, X_3)$.

2) Authorization: As shown in Figure 3, suppose DU has an attribute set $\mathcal{S} = (\mathcal{I}_S, S)$ with $\mathcal{I}_S \subseteq \mathbb{Z}_N$ and $S = \{s_i\}_{i \in \mathcal{I}_S}$. SHA grants access rights to DU based on the following algorithm KeyGen.

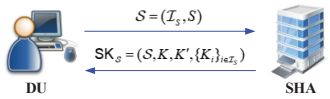


Fig. 3. The authorization in PASH.

- KeyGen($\mathsf{PK}, \mathsf{MK}, \mathcal{S}$): SHA uniformly chooses $t \in_R \mathbb{Z}_N$ and $R, R', R_i \in_R \mathbb{G}_{p_3}$ for $i \in \mathcal{I}_S$. It outputs the secret key $\mathsf{SK}_\mathcal{S} = (\mathcal{S}, K, K', \{K_i\}_{i \in \mathcal{I}_S})$, where

$$K = g^\alpha g^{at} R, K' = g^t R', K_i = (g^{s_i} h)^t R_i.$$

3) Privacy-Aware SHR Outsourcing: DO chooses a symmetric encryption scheme such as AES and uses it to encrypt his SHRs. Then, DO specifies an access policy $\mathbb{A} = (\mathbf{A}, \rho, \mathcal{T})$, where $\mathbf{A}$ is an $\ell \times n$ matrix, ρ is a map from each row A_x of $\mathbf{A}$ to an attribute name, and $\mathcal{T} = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$. DO encrypts the symmetric key which is used as a plaintext $M \in \mathbb{G}_T$ in the algorithm PH.Encrypt as below. Finally, as shown in Figure 4, DO outsources ciphertext data to SHC.

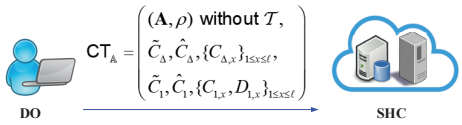


Fig. 4. The privacy-aware SHR outsourcing in PASH.

- PH.Encrypt($\mathsf{PK}, M, \mathbb{A}$): DO chooses two vectors $v, v' \in_R \mathbb{Z}_N^n$ with $v = (s, v_2, \ldots, v_n)$ and $v' = (s', v_2', \ldots, v_n')$. It also uniformly chooses $Z_\Delta \in_R \mathbb{G}_{p_4}$ based on X_4, $r_x \in_R \mathbb{Z}_N$ and $Z_{\Delta,x}, Z_{c,x}, Z_{d,x} \in_R \mathbb{G}_{p_4}$ based on X_4, for $1 \le x \le \ell$. Then it calculates the ciphertext as

$$\mathsf{CT}_\mathbb{A} = ((\mathbf{A}, \rho), \tilde{C}_\Delta, \hat{C}_\Delta, \{C_{\Delta,x}\}_{1 \le x \le \ell}, \tilde{C}_1, \hat{C}_1, \{C_{1,x}, D_{1,x}\}_{1 \le x \le \ell}),$$

where $\tilde{C}_\Delta = Y^{s'}, \hat{C}_\Delta = g^{s'} Z_\Delta, C_{\Delta,x} = g^{aA_x \cdot v'}(g^{t_{\rho(x)}} H)^{-s'} Z_{\Delta,x}$, $\tilde{C}_1 = M \cdot Y^s, \hat{C}_1 = g^s$, and

$$C_{1,x} = g^{aA_x \cdot v}(g^{t_{\rho(x)}} H)^{-r_x} Z_{c,x}, D_{1,x} = g^{r_x} Z_{d,x}.$$

4) Privacy-Aware SHR Access: DU can get SHRs based on the symmetric key, which acts as the plaintext M and is obtained in the algorithm PH.Decrypt as below. Let the ciphertext of M be

$$\mathsf{CT}_\mathbb{A} = ((\mathbf{A}, \rho), \tilde{C}_\Delta, \hat{C}_\Delta, \{C_{\Delta,x}\}_{1 \le x \le \ell}, \tilde{C}_1, \hat{C}_1, \{C_{1,x}, D_{1,x}\}_{1 \le x \le \ell}),$$

and the attribute secret key of DU be $\mathsf{SK}_\mathcal{S} = (\mathcal{S}, K, K', \{K_i\}_{i \in \mathcal{I}_S})$ with $\mathcal{S} = (\mathcal{I}_S, S)$ and $S = \{s_i\}_{i \in \mathcal{I}_S}$. Two phases in decryption of M are shown in Figure 5.

- PH.Decrypt($\mathsf{PK}, \mathsf{CT}_\mathbb{A}, \mathsf{SK}_\mathcal{S}$): DU first calculates $\mathbf{I}_{\mathbf{A},\rho}$ from $(\mathbf{A}, \rho)$, where $\mathbf{I}_{\mathbf{A},\rho}$ denotes the set of minimum subsets of $\{1, 2, \ldots, \ell\}$ that satisfies $(\mathbf{A}, \rho)$. Then DU does the following.

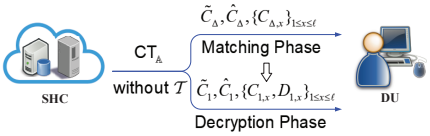


Fig. 5. The privacy-aware SHR access in PASH.

– Matching Phase: It checks if there exists a subset $\mathcal{I} \in \mathbf{I}_{\mathbf{A},\rho}$ that satisfies $\{\rho(i)|i \in \mathcal{I}\} \subseteq \mathcal{I}_S$ and

$$\tilde{C}_\Delta^{-1} = \hat{e}\left(\prod_{i\in\mathcal{I}} C_{\Delta,i}^{\omega_i}, K'\right)\hat{e}\left(\hat{C}_\Delta, K^{-1}\prod_{i\in\mathcal{I}} K_{\rho(i)}^{\omega_i}\right),$$

where $\sum_{i\in\mathcal{I}} \omega_i A_i = (1,0,\ldots,0)$ for some constants $\{\omega_i\}_{i\in\mathcal{I}}$. If no such $\mathcal{I}$ exists, it outputs $\perp$ to indicate that $\mathcal{S}$ does not satisfy the partially hidden $\mathbb{A}$. Otherwise, it initiates the decryption phase based on the eligible $\mathcal{I}$ and $\{\omega_i\}_{i\in\mathcal{I}}$.

– Decryption Phase: It returns $M = \tilde{C}_1/E$, where

$$E = \frac{\hat{e}(\hat{C}_1, K)}{\prod_{i\in\mathcal{I}}\left(\hat{e}(C_{1,i}, K')\hat{e}(D_{1,i}, K_{\rho(i)})\right)^{\omega_i}}.$$

C. Soundness of PASH

The proposed PASH achieves soundness, which means DU can access some SHRs if and only if his attributes match the underlying access policy. For one thing, the matching phase is correct. Indeed,

$$\begin{aligned}
&\tilde{C}_\Delta^{-1} = \hat{e}\left(\prod_{i\in\mathcal{I}} C_{\Delta,i}^{\omega_i}, K'\right)\hat{e}\left(\hat{C}_\Delta, K^{-1}\prod_{i\in\mathcal{I}} K_{\rho(i)}^{\omega_i}\right)\\
\iff\quad &\tilde{C}_\Delta = \frac{\hat{e}(\hat{C}_\Delta, K)}{\prod_{i\in\mathcal{I}}\left(\hat{e}(C_{\Delta,i}, K')\hat{e}(\hat{C}_\Delta, K_{\rho(i)})\right)^{\omega_i}}.
\end{aligned}$$

If and only if $t_{\rho(i)} = s_{\rho(i)}$, for $i \in \mathcal{I}$, we have

$$\begin{aligned}
&\frac{\hat{e}(\hat{C}_\Delta, K)}{\prod_{i\in\mathcal{I}}\left(\hat{e}(C_{\Delta,i}, K')\hat{e}(\hat{C}_\Delta, K_{\rho(i)})\right)^{\omega_i}}\\
=\ &\frac{\hat{e}(g^{s'}Z_\Delta, g^\alpha g^{at}R)}{\prod_{i\in\mathcal{I}}\left(\hat{e}(g^{aA_i\cdot v'}(g^{t_{\rho(i)}}H)^{-s'}Z_{\Delta,i}, g^t R')\hat{e}(g^{s'}Z_\Delta, (g^{s_{\rho(i)}}h)^t R_{\rho(i)})\right)^{\omega_i}}\\
=\ &\frac{\hat{e}(g^{s'}, g^\alpha g^{at})}{\prod_{i\in\mathcal{I}}\left(\hat{e}(g^{aA_i\cdot v'}, g^t)\right)^{\omega_i}} = \frac{\hat{e}(g^{s'}, g^\alpha g^{at})}{(\hat{e}(g^a, g^t))^{(\sum_{i\in\mathcal{I}}\omega_i A_i)\cdot v'}} = \tilde{C}_\Delta.
\end{aligned}$$

For another, the decryption phase is correct. In fact, if $\mathcal{S}$ satisfies $\mathbb{A}$, there exists an eligible set $\mathcal{I}$ and constants $\{\omega_i\}_{i\in\mathcal{I}}$ such that $\sum_{i\in\mathcal{I}}\omega_i A_i = (1,0,\ldots,0)$, $t_{\rho(i)} = s_{\rho(i)}$ for $i \in \mathcal{I}$, and

$$\begin{aligned}
E\ =\ &\frac{\hat{e}(\hat{C}_1, K)}{\prod_{i\in\mathcal{I}}\left(\hat{e}(C_{1,i}, K')\hat{e}(D_{1,i}, K_{\rho(i)})\right)^{\omega_i}}\\
=\ &\frac{\hat{e}(g^s, g^\alpha g^{at}R)}{\prod_{i\in\mathcal{I}}\left(\hat{e}(g^{aA_i\cdot v}(g^{t_{\rho(i)}}H)^{-r_i}Z_{c,i}, g^t R')\hat{e}(g^{r_i}Z_{d,i}, (g^{s_{\rho(i)}}h)^t R_{\rho(i)})\right)^{\omega_i}}\\
=\ &\frac{\hat{e}(g^s, g^\alpha g^{at})}{\prod_{i\in\mathcal{I}}\left(\hat{e}(g^{aA_i\cdot v}, g^t)\right)^{\omega_i}} = \frac{\hat{e}(g^s, g^\alpha g^{at})}{(\hat{e}(g^a, g^t))^{(\sum_{i\in\mathcal{I}}\omega_i A_i)\cdot v}} = Y^s.
\end{aligned}$$

Therefore, the proposed PASH is sound.

VI. Security and Performance Analysis

A. Security Analysis

Based on hybrid encryption mechanisms, if the proposed PH-CP-ABE is secure, then PASH is secure in our adversary model. Our security proofs are based on several complexity assumptions as below. Note that Assumptions 1, 2 and 3 are the same assumptions as in [11], and they are used as in [15] in the group of composite order. Assumption 4 was used in [53], [15]. The assumptions were proved to be generically secure in [54], [53].

Assumption 1. Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}) \xleftarrow{R} \mathcal{G}, g \xleftarrow{R} \mathbb{G}_{p_1}, X_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4},\\
&D = (N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, X_3, X_4), T_1 \xleftarrow{R} \mathbb{G}_{p_1}\times\mathbb{G}_{p_2}, T_2 \xleftarrow{R} \mathbb{G}_{p_1}.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is $\mathsf{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$.

Definition 5: $\mathcal{G}$ satisfies Assumption 1 if $\mathsf{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$.

Assumption 2. Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}) \xleftarrow{R} \mathcal{G},\\
&g, X_1 \xleftarrow{R} \mathbb{G}_{p_1}, X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, X_3, Y_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4},\\
&D = (N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, X_1X_2, Y_2Y_3, X_3, X_4),\\
&T_1 \xleftarrow{R} \mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_3}, T_2 \xleftarrow{R} \mathbb{G}_{p_1}\times\mathbb{G}_{p_3}.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$.

Definition 6: $\mathcal{G}$ satisfies Assumption 2 if $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any PPT algorithm $\mathcal{A}$.

Assumption 3. Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}) \xleftarrow{R} \mathcal{G}, \alpha, s \xleftarrow{R} \mathbb{Z}_N,\\
&g \xleftarrow{R} \mathbb{G}_{p_1}, g_2, X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}, X_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4},\\
&D = (N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_2, g^\alpha X_2, g^s Y_2, X_3, X_4),\\
&T_1 = \hat{e}(g, g)^{\alpha s}, T_2 \xleftarrow{R} \mathbb{G}_T.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is $\mathsf{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$.

Definition 7: $\mathcal{G}$ satisfies Assumption 3 if $\mathsf{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any PPT algorithm $\mathcal{A}$.

Assumption 4. Given a group generator $\mathcal{G}$, define the following distribution:

$$\begin{aligned}
&(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}) \xleftarrow{R} \mathcal{G}, t', r' \xleftarrow{R} \mathbb{Z}_N, g, h \xleftarrow{R} \mathbb{G}_{p_1},\\
&g_2, X_2, A_2, B_2, D_2 \xleftarrow{R} \mathbb{G}_{p_2}, X_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4, Z, A_4, D_4 \xleftarrow{R} \mathbb{G}_{p_4},\\
&D = (N, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_2, g^{t'}B_2, h^{t'}Y_2, X_3, X_4, hZ, g^{r'}D_2D_4),\\
&T_1 = h^{r'}A_2A_4, T_2 \xleftarrow{R} \mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_4}.
\end{aligned}$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is $\mathsf{Adv4}_{\mathcal{G},\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$.

Definition 8: $\mathcal{G}$ satisfies Assumption 4 if $\mathsf{Adv4}_{\mathcal{G},\mathcal{A}}(\lambda)$ is a negligible function of λ for any PPT algorithm $\mathcal{A}$.

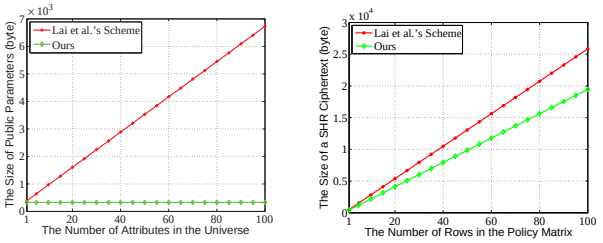
Theorem 1: If Assumptions 1, 2, 3 and 4 hold, then the proposed PH-CP-ABE scheme is fully secure in our formalized security model based on Definition 4.

Proof. Security proofs are given in the APPENDIX A. ■

B. Performance Analysis

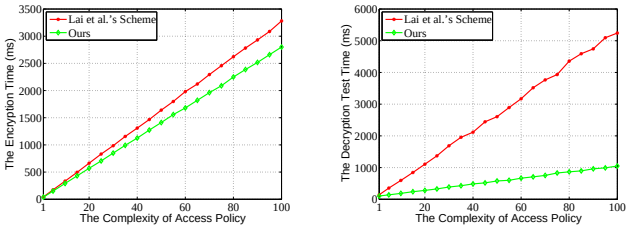
In this section, we compare our scheme with previous related work [41], [42], [43], [45], [46], [40], [14], [15] in terms of security and performance features. In Table II, we give comprehensive comparisons according to important features including privacy protection, decryption test, large universe, full security, access policy expressiveness, group type and security model. From Table II, we know that every scheme realizes users' privacy protection. However, only PASH and schemes in [14], [15] are capable of supporting decryption test. Only PASH and the scheme in [40] are large universe constructions. Full security is realized in PASH and the schemes in [45], [46], while only PASH supports LSSS access policies. In addition, PASH is proven secure in the standard model under static assumptions. The scheme in [40] cannot support decryption test and is proven secure under non-static assumptions. Based on the above comparisons, only PASH and Lai et al.'s scheme [15] can simultaneously realize decryption test and full security while the latter cannot support large universe. We further compare PASH with the scheme in [15] in Table III, where PK, SK and CT respectively denote the public parameters, the secret key and the SHR ciphertext. Note that the length of an element in each subgroup $\mathbb{G}_{p_i}$ and the target group $\mathbb{G}_T$ is set to 512 bits. We use Pair, Exp and Exp_T to represent a bilinear pairing operation, an exponentiation operation in $\mathbb{G}$ and an exponentiation operation in $\mathbb{G}_T$, respectively. Let N denote the size of the attribute universe, m the number of a user's attributes, and ℓ the number of rows in a policy matrix. Note that the minimum authorized set $\mathcal{I}$ is used in the SHR access phase and its size $|\mathcal{I}|$ is determined by the complexity of the access policy. From Table III, we know that the size of PK in PASH is small and constant, while it increases linearly with N in [15]. In addition, PASH is superior to [15] with respect to the size of CT, the encryption cost (i.e. the SHR outsourcing time), and the decryption test cost. The comparison of parameter size is also shown in Figure 6 for clarity.

To explicitly demonstrate the performance advantage of PASH, we implement PASH and the scheme in [15] on a laptop (with 2.90 GHz Intel Pentium(R) CPU and 4 GB RAM memory) based on Ubuntu 16.04 LTS and the Java Pairing Based Cryptography Library (JPBC) 2.0.0 [55]. We evaluate the computation time of the SHR outsourcing phase, the attribute matching phase, and the entire SHR access phase. In our experiments, Type A1 pairings are adopted, which are constructed on the curve $y^2 = x^3 + x$ over the field $\mathbb{F}_N$ with the composite N being the universe size. For each access policy, the experiment is repeated 30 times, and the average values are adopted. In Figure 7, we compare PASH and [15] in terms of the SHR encryption time and the decryption test time. Figure 7(a) and Figure 7(b) respectively show the SHR encryption time and the decryption time versus the complexity $|\mathcal{I}|$ of access policy, which is measured by the size of the minimum authorized set $\mathcal{I}$. The comparison of SHR access time is shown in Figure 8, in which for a given $|\mathcal{I}|$, the access time varies with the number of SHR ciphertexts. In practice, the SHC server may search and send a lot of CP-ABE ciphertexts to DU of which only one can be decrypted. Therefore, in Figure 8(a), Figure 8(b), Figure 8(c) and Figure 8(d), the number of valid ciphertexts is 1 and $|\mathcal{I}|$ is set to 2, 5, 10 and 20, respectively. Obviously, our experiment results show that PASH is very efficient in addition to possessing other attractive features.



(a) Public Parameters (b) SHR Ciphertexts

Fig. 6. The size of parameters.



(a) The SHR Encryption Time (b) The Decryption Test Time

Fig. 7. The SHR encryption and decryption test time.

In summary, PASH is superior to the existing schemes since it can handle the most expressive access policies and is fully secure in the standard model. In particular, PASH supports large universe and the decryption test is most efficient. Therefore, PASH is highly suitable for data security and privacy protection in s-health.

VII. Conclusion and Future Work

In this paper, we efficiently addressed data security and user privacy issues in s-health by introducing PASH, a privacy-aware s-health access control system. The main building block of PASH is a CP-ABE scheme which supports large universe and partially hidden access policies. In PASH, sensitive attribute values involved in access policies are hidden and generic attribute names are public. We added an efficient decryption test before full decryption to improve efficiency. The large universe construction enables public parameters of a constant number of group elements. In addition, PASH supports LSSS policies and was proven fully secure in the standard model. Theoretical analysis and experimental results indicated that PASH is more secure, efficient, and expressive than existing schemes.

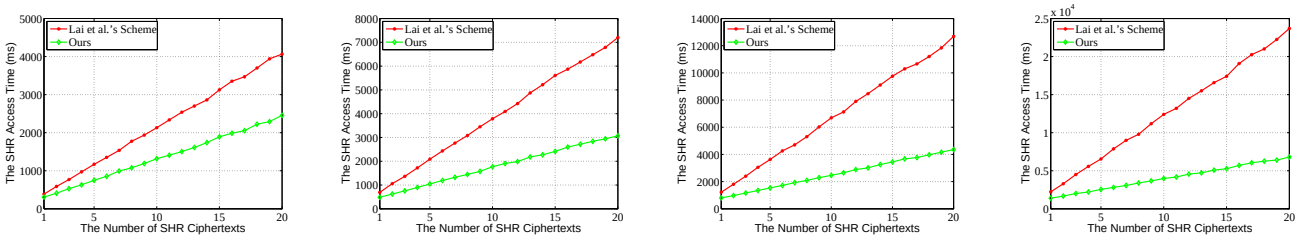
TABLE II
COMPARISONS OF CP-ABE SCHEMES

Schemes	Privacy Aware	Decryption Test	Large Universe	Full Security	Expressiveness	Group Order	Standard Model
[41], [42]	✓	×	×	×	AND†	Prime	×
[43]	✓	×	×	×	AND	Prime	✓
[45], [46]	✓	×	×	✓	AND	Composite	✓
[40]	✓	×	✓	×	LSSS	Prime	×
[14]	✓	✓	×	×	AND	Prime	×
[15]	✓	✓	×	✓	LSSS	Composite	✓
Ours	✓	✓	✓	✓	LSSS	Composite	✓

† AND-gates on two/multi-valued attributes with wildcards.

TABLE III
PERFORMANCE COMPARISONS BETWEEN FULLY SECURE CP-ABE SUPPORTING DECRYPTION TEST

Schemes	PK Size		SK Size	CT Size		Encryption Cost		Decryption Test Cost			Decryption Phase Cost		
	$\mathbb{G}_{p_i}$	$\mathbb{G}_T$	$\mathbb{G}_{p_i p_j}$	$\mathbb{G}_{p_i p_j}$	$\mathbb{G}_T$	Exp	Exp_T	Pair	Exp	Exp_T	Pair	Exp	Exp_T
[15]	$N+4$	1	$m+2$	$4\ell+2$	2	$7\|\mathcal{I}\|+2$	2	$\|\mathcal{I}\|+2$	$\|\mathcal{I}\|$	$\|\mathcal{I}\|$	$\|\mathcal{I}\|+2$	$\|\mathcal{I}\|$	$\|\mathcal{I}\|$
Ours	4	1	$m+2$	$3\ell+2$	2	$6\|\mathcal{I}\|+2$	2	2	$2\|\mathcal{I}\|$	0	$\|\mathcal{I}\|+2$	$\|\mathcal{I}\|$	$\|\mathcal{I}\|$



(a) The Access Policy Complexity ($|\mathcal{I}| = 2$) (b) The Access Policy Complexity ($|\mathcal{I}| = 5$) (c) The Access Policy Complexity ($|\mathcal{I}| = 10$) (d) The Access Policy Complexity ($|\mathcal{I}| = 20$)

Fig. 8. The SHR Access Time.

It would be interesting to construct a privacy-preserving and fully-secure s-health system supporting large universe and efficient decryption test under prime order groups. Another possible goal for future research would be to find attribute revocation and traceability mechanisms suitable for s-health.

ACKNOWLEDGMENT

We are grateful to the associate editor and reviewers for their invaluable suggestions. This research is supported by the National Key R&D Program of China (2017YFB0802000), the AXA Reserch Fund, the National Natural Science Foundation of China (Nos. 61772418, 61472472, 61402366), the Natural Science Basic Research Plan in Shaanxi Province of China (No. 2015JQ6236). Yinghui Zhang is supported by New Star Team of Xi'an University of Posts & Telecommunications.

REFERENCES

[1] A. Solanas, C. Patsakis, M. Conti, I. S. Vlachos, V. Ramos, F. Falcone, O. Postolache, P. A. Pérez-Martínez, R. Di Pietro, D. N. Perrea *et al.*, "Smart health: a context-aware health paradigm within smart cities," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 74-81, 2014.

[2] Y. Yuehong, Y. Zeng, X. Chen, and Y. Fan, "The internet of things in healthcare: An overview," *Journal of Industrial Information Integration*, vol. 1, pp. 3-13, 2016.

[3] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233-2243, 2014.

[4] S. Li, L. Da Xu, and S. Zhao, "5g internet of things: A survey," *Journal of Industrial Information Integration*, 2018. [Online]. Available: https://doi.org/10.1016/j.jii.2018.01.005

[5] C. Wang, Z. Bi, and L. Da Xu, "Iot and cloud computing in automation of assembly modeling systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1426-1434, 2014.

[6] X. Zheng, P. Martin, K. Brohman, and L. Da Xu, "Cloudqual: a quality model for cloud services," *IEEE transactions on industrial informatics*, vol. 10, no. 2, pp. 1527-1536, 2014.

[7] B. Xu, L. Xu, H. Cai, L. Jiang, Y. Luo, and Y. Gu, "The design of an m-health monitoring system based on a cloud computing platform," *Enterprise Information Systems*, vol. 11, no. 1, pp. 17-36, 2017.

[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology (EUROCRYPT'05)*, 2005, pp. 557-557.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of ACM Conference on Computer and Communications Security, (CCS'06)*, 2006, pp. 89-98.

[10] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'08)*, 2008, pp. 146-162.

[11] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'10)*, 2010, pp. 62-91.

[12] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Technion-Israel Institute of Technology, Faculty of Computer Science, 1996. Available: https://www.cs.bgu.ac.il/~beimel/Papers/thesis.pdf

[13] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'11)*, 2011, pp. 568-588.

[14] Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li, "Anonymous attribute-based encryption supporting efficient decryption test," in *Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS'13)*, 2013, pp. 511-516.

[15] J. Lai, R. H. Deng, and Y. Li, "Expressive cp-abe with partially hidden access structures," in *Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS'12)*, 2012, pp. 18-19.

[16] H. Yan, L. D. Xu, Z. Bi, Z. Pang, J. Zhang, and Y. Chen, "An emerging technology-wearable wireless sensor networks with applications in human health condition monitoring," *Journal of Management Analytics*, vol. 2, no. 2, pp. 121-137, 2015.

[17] M. Salayma, A. Al-Dubai, I. Romdhani, and Y. Nasser, "Wireless body area network (wban): A survey on reliability, fault tolerance, and technologies coexistence," *ACM Computing Surveys*, vol. 50, no. 1, p. 3, 2017.

[18] F. Hu, X. Liu, D. Sui, M. Shao, and L. Wang, "Performance analysis of reliability in wireless body area networks," *IET Communications*, vol. 11, no. 6, pp. 925-929, 2017.

[19] Q. Han, Y. Zhang, X. Chen, H. Li, and J. Quan, "Efficient and robust identity-based handoff authentication in wireless networks," in *Proceedings of International Conference on Network and System Security (NSS'12)*, 2012, pp. 180-191.

[20] Y. Zhang, X. Chen, J. Li, and H. Li, "Generic construction for secure and efficient handoff authentication schemes in eap-based wireless networks," *Computer Networks*, vol. 75, pp. 192-211, 2014.

[21] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, "Ubiquitous data accessing method in iot-based information system for emergency medical services," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1578-1586, 2014.

[22] X. Zheng, P. Martin, K. Brohman, and L. Da Xu, "Cloud service negotiation in internet of things environment: A mixed approach," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1506-1515, 2014.

[23] S. K. Datta, C. Bonnet, and N. Nikaein, "An iot gateway centric architecture to provide novel m2m services," in *Proceedings of IEEE World Forum on Internet of Things (WF-IoT'14)*, 2014, pp. 514-519.

[24] F. Liu, C.-W. Tan, E. T. Lim, and B. Choi, "Traversing knowledge networks: an algorithmic historiography of extant literature on the internet of things (iot)," *Journal of Management Analytics*, vol. 4, no. 1, pp. 3-34, 2017.

[25] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Efficient attribute-based data sharing in mobile clouds," *Pervasive and Mobile Computing*, vol. 28, pp. 135-149, 2016.

[26] C.-K. Chu, S. S. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," *IEEE transactions on parallel and distributed systems*, vol. 25, no. 2, pp. 468-477, 2014.

[27] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," *Information Sciences*, vol. 305, pp. 357-383, 2015.

[28] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of IEEE Symposium on Security and Privacy (SP'07)*, 2007, pp. 321-334.

[29] L. Cheung and C. Newport, "Provably secure ciphertext policy abe," in *Proceedings of ACM Conference on Computer and Communications Security (CCS'07)*, 2007, pp. 456-465.

[30] Z. Wan, J. Liu, and R. H. Deng, "Hasbe: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE transactions on information forensics and security*, vol. 7, no. 2, pp. 743-754, 2012.

[31] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts," in *Proceedings of International Conference on Provable Security (ProvSec'14)*, 2014, pp. 259-273.

[32] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proceedings of International Workshop on Public Key Cryptography (PKC'11)*, 2011, pp. 53-70.

[33] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1-12, 2018.

[34] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS'10)*, 2010, pp. 261-270.

[35] Y. Zhang, X. Chen, J. Li, H. Li, and F. Li, "Fdr-abe: Attribute-based encryption with flexible and direct revocation," in *Proceedings of International Conference on Intelligent Networking and Collaborative Systems (INCoS'13)*, 2013, pp. 38-45.

[36] Z. Liu, Z. Cao, and D. S. Wong, "Blackbox traceable cp-abe: how to catch people leaking their keys by selling decryption devices on ebay," in *Proceedings of ACM Conference on Computer & Communications Security (CCS'13)*, 2013, pp. 475-486.

[37] Y. Zhang, J. Li, D. Zheng, X. Chen, and H. Li, "Accountable large-universe attribute-based encryption supporting any monotone access structures," in *Proceedings of Australasian Conference on Information Security and Privacy (ACISP'16)*, 2016, pp. 509-524.

[38] Y. Zhang, J. Li, X. Chen, and H. Li, "Anonymous attribute-based proxy re-encryption for access control in cloud computing," *Security and Communication Networks*, vol. 9, no. 14, pp. 2397-2411, 2016.

[39] Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42-61, 2017.

[40] H. Cui, R. H. Deng, G. Wu, and J. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," in *Proceedings of International Conference on Provable Security (ProvSec'16)*, 2016, pp. 19-38.

[41] T. Nishide, K. Yoneyama, and K. Ohta, "Abe with partially hidden encryptor-specified access structure," in *Proceedings of International Conference on Applied Cryptography and Network Security (ACNS'08)*, 2008, pp. 111-129.

[42] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in *Proceedings of International Conference on Information Security (ISC'09)*, 2009, pp. 347-362.

[43] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 35-45, 2016.

[44] K. Yang, Q. Han, H. Li, K. Zheng, Z. Su, and X. Shen, "An efficient and fine-grained big data access control scheme with privacy-preserving policy," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 563-571, 2017.

[45] J. Lai, R. H. Deng, and Y. Li, "Fully secure cipertext-policy hiding cp-abe," in *Proceedings of International Conference on Information Security Practice and Experience (ISPEC'11)*, 2011, pp. 24-39.

[46] C. Jin, X. Feng, and Q. Shen, "Fully secure hidden ciphertext policy attribute-based encryption with short ciphertext size," in *Proceedings of International Conference on Communication and Network Security (ICCNS'16)*, 2016, pp. 91-98.

[47] C. Dong, L. Chen, and Z. Wen, "When private set intersection meets big data: an efficient and scalable protocol," in *Proceedings of ACM Conference on Computer & Communications Security (CCS'13)*, 2013, pp. 789-800.

[48] B. Waters, "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions," in *Proceedings of Annual International Cryptology Conference (CRYPTO'09)*, 2009, pp. 619-636.

[49] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Proceedings of Annual International Cryptology Conference (CRYPTO'10)*, 2010, pp. 191-208.

[50] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of ACM Conference on Computer & Communications security (CCS'13)*, 2013, pp. 463-474.

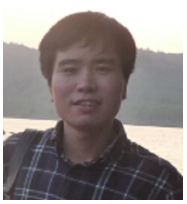
[51] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Proceedings of Theory of Cryptography Conference (TCC'05)*, 2005, pp. 325-341.

[52] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Proceedings of Annual International Cryptology Conference (CRYPTO'12)*, 2012, pp. 180-198.

[53] A. De Caro, V. Iovino, and G. Persiano, "Fully secure anonymous hibe and secret-key anonymous ibe with short ciphertexts," in *Proceedings of International Conference on Pairing-Based Cryptography (Pairing'10)*, 2010, pp. 347-366.

[54] A. Lewko and B. Waters, "New techniques for dual system encryption and fully secure hibe with short ciphertexts," in *Proceedings of Theory of Cryptography Conference (TCC'10)*, 2010, pp. 455-479.

[55] A. D. Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *Proceedings of IEEE Symposium on Computers and Communications (ISCC'11)*, 2011, pp. 850-855.



Yinghui Zhang (M'18) received his Ph.D degree in Cryptography from Xidian University, China, in 2013. He is an associate professor at National Engineering Laboratory for Wireless Security (NELWS), Xi'an University of Posts & Telecommunications. Currently, he is also a research fellow at Singapore Management University. He has published over 50 research articles including ASIACCS, ACISP, Computer Networks, Computers & Security. His research interests include cloud security, public key cryptography and wireless network security.

Dong Zheng received his Ph.D. degree in communication engineering from Xidian University, China, in 1999. He was a Professor at the School of Information Security Engineering, Shanghai Jiao Tong University. He is currently a Professor at National Engineering Laboratory for Wireless Security (NELWS), Xi'an University of Posts & Telecommunications. He has published over 100 research articles including CT-RSA, IEEE Transactions on industrial electronics. His research interests include cloud computing and public key cryptography.

Robert H. Deng (F'16) is AXA Chair Professor of Cybersecurity and Professor of Information Systems in the School of Information Systems, Singapore Management University since 2004. His research interests include data security and privacy, multimedia security, network and system security. He served/is serving on the editorial boards of many international journals in security, including the IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, etc. He is a fellow of the IEEE.

Appendix

We first give the security proof of Theorem 1. Then, we prove the security of Lemma 2-Lemma 7 in detail.

A. Proof of Theorem 1

Proof. Following the dual system encryption methodology [11], we define two additional structures: *semi-functional ciphertext* and *semi-functional key*. These will not be used in the real system, but will be needed in our proof. Let g_2 denote a generator of the subgroup $\mathbb{G}_{p_2}$. Semi-functional ciphertexts and semi-functional keys are created as follows.

Semi-functional Ciphertext. We first choose two exponents $c, c' \in_R \mathbb{Z}_N$ and two vectors $\omega, \omega' \in_R \mathbb{Z}_N^n$. We also choose $z_i \in_R \mathbb{Z}_N$ associated to attributes and $\gamma_x, \gamma_x' \in_R \mathbb{Z}_N$ with respect to matrix rows A_x. Based on the normal ciphertext output by PH.Encrypt, the semi-functional ciphertext $\mathsf{CT}_{\mathbb{A}}$ is set as

$$\mathsf{CT}_{\mathbb{A}} = ((\mathbf{A}, \rho), \tilde{C}_\Delta, \hat{C}_\Delta, \{C_{\Delta,x}\}_{1\le x\le \ell}, \tilde{C}_1, \hat{C}_1, \{C_{1,x}, D_{1,x}\}_{1\le x\le \ell}),$$

where $\tilde{C}_\Delta = Y^{s'}, \hat{C}_\Delta = g^{s'} Z_\Delta g_2^{c'}$ and

$$\begin{aligned} C_{\Delta,x} &= g^{aA_x\cdot v'}(g^{t_{\rho(x)}}H)^{-s'} Z_{\Delta,x} g_2^{A_x\cdot\omega' + \gamma_x' z_{\rho(x)}}, \\ \tilde{C}_1 &= M\cdot Y^s, \hat{C}_1 = g^s g_2^c, \\ C_{1,x} &= g^{aA_x\cdot v}(g^{t_{\rho(x)}}H)^{-r_x} Z_{c,x} g_2^{A_x\cdot\omega + \gamma_x z_{\rho(x)}}, D_{1,x} = g^{r_x} Z_{d,x} g_2^{-\gamma_x}. \end{aligned}$$

Semi-functional Key. We choose exponents $d, d' \in_R \mathbb{Z}_N$ and $\{d_i \in_R \mathbb{Z}_N\}_{i\in I_S}$. Then, based on the normal secret key output by KeyGen, a semi-functional key will take one of three forms.

- A semi-functional key of type 1 is set as

$$(\mathcal{S}, K = g^\alpha g^{at} R g_2^d, K' = g^t R' g_2^{d'}, \{K_i = (g^{s_i} h)^t R_i g_2^{d' z_i}\}_{i\in I_S}).$$

- A semi-functional key of type 2 is set as

$$(\mathcal{S}, K = g^\alpha g^{at} R g_2^d, K' = g^t R', \{K_i = (g^{s_i} h)^t R_i\}_{i\in I_S}).$$

- A semi-functional key of type 3 is set as

$$(\mathcal{S}, K = g^\alpha g^{at} R g_2^d, K' = g^t R' g_2^{d'}, \{K_i = (g^{s_i} h)^t R_i g_2^{d_i}\}_{i\in I_S}).$$

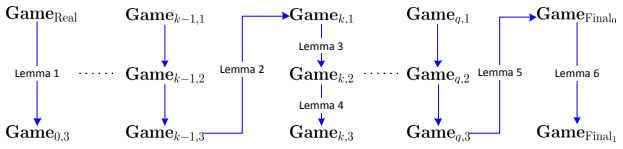


Fig. 9. The relationship between the lemmas and the games.

We will prove the security of the proposed scheme based on Assumptions 1, 2, 3 and 4 using a hybrid argument over a sequence of games as below.

- **Game**$_{\text{Real}}$. The first game is the real security game, in which the ciphertext and all the keys are normal.
- **Game**$_0$. In the second game, all the keys are normal, but the challenge ciphertext is semi-functional. It is also denoted as **Game**$_{0,3}$.

We let q denote the number of key queries made by the adversary. For $1 \le k \le q$, we define the following games.

- **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.
- **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.
- **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. Note that, in **Game**$_{q,3}$, the challenge ciphertext is semi-functional, all the keys are semi-functional of type 3.
- **Game**$_{\text{Final}_0}$. In this game, the challenge ciphertext is a semi-functional encryption of a random message, independent of M_0 and M_1 provided by the adversary, and all the keys are semi-functional of type 3.
- **Game**$_{\text{Final}_1}$. The final game is the same as **Game**$_{\text{Final}_0}$, except that $C_{\Delta,x}$ and $C_{1,x}$ in the challenge ciphertext are random elements in $\mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_4}$. The challenge ciphertext is independent of $\mathcal{T}_0$ and $\mathcal{T}_1$ provided by the adversary. Hence, in **Game**$_{\text{Final}_1}$, the adversary's advantage is 0.

We prove these games are indistinguishable based on Lemma 2-Lemma 7 given in the APPENDIX B. The relationship between the lemmas and the games is given in Fig. 9, where $2 \le k \le q$. Therefore, if Assumptions 1, 2, 3 and 4 hold, we have shown that **Game**$_{\text{Real}}$ is indistinguishable from **Game**$_{\text{Final}_1}$. Hence, the adversary cannot obtain a non-negligible advantage in breaking the proposed scheme. ■

B. Proofs of Lemma 2-Lemma 7

B.1. Proof of Lemma 2

Lemma 2: Suppose that $\mathcal{G}$ satisfies Assumption 1. Then $\mathsf{Game}_{\text{Real}}$ and Game_0 are computationally indistinguishable.

Proof. Suppose there exists an adversary $\mathcal{A}$ such that $|\mathsf{Game}_{\text{Real}}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_0\mathsf{Adv}_{\mathcal{A}}| = \epsilon$, then we construct a simulator $\mathcal{B}$ with $\mathsf{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$ in breaking Assumption 1. $\mathcal{B}$ is given g, X_3, X_4, T and simulates $\mathsf{Game}_{\text{Real}}$ or Game_0 with $\mathcal{A}$.

Setup. $\mathcal{B}$ uniformly chooses $\alpha, a, a_0 \in_R \mathbb{Z}_N$ and $Z \in_R \mathbb{G}_{p_4}$. It then sets $Y = \hat{e}(g,g)^{\alpha}, h = g^{a_0}, H = hZ$, and sends $\mathcal{A}$ the system public parameters $\mathsf{PK} = (N, g, g^a, Y, H, X_4)$.

Phase 1. $\mathcal{B}$ can generate normal keys in response to $\mathcal{A}$'s key requests by using the key generation algorithm, since it knows the master key $\mathsf{MK} = (\alpha, h, X_3)$.

Challenge. Once $\mathcal{A}$ submits to $\mathcal{B}$ two messages M_0, M_1 of equal length and two access structures $\mathbb{A}_1 = (\mathbf{A}, \rho, \mathcal{T}_0)$, $\mathbb{A}_2 = (\mathbf{A}, \rho, \mathcal{T}_1)$ with the restriction that $\mathbb{A}_1$ and $\mathbb{A}_2$ cannot be satisfied by any of the queried attribute sets in **Phase 1**. Let $\mathcal{T}_\beta = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ and does the following.

1) $\mathcal{B}$ creates $\tilde{v} = (1, \tilde{v}_2, \ldots, \tilde{v}_n)$, $\tilde{v}' = (1, \tilde{v}'_2, \ldots, \tilde{v}'_n)$ and $v_\Delta = (0, v_{\Delta,2}, \ldots, v_{\Delta,n})$ with $\tilde{v}_i, \tilde{v}'_i, v_{\Delta,i} \in_R \mathbb{Z}_N$ for $2 \le i \le n$.
2) $\mathcal{B}$ chooses $\tilde{r}_x \in_R \mathbb{Z}_N$, $\hat{s}_x = (a_0 + t_{\rho(x)})^{-1}$ and $Z_\Delta, \tilde{Z}_{\Delta,x}, \tilde{Z}_{c,x}, Z_{d,x} \in_R \mathbb{G}_{p_4}$, for $1 \le x \le \ell$.
3) $\mathcal{B}$ chooses $\tilde{s} \in \mathbb{Z}_N$ and calculates

$$\begin{aligned}\tilde{C}_\Delta &= \hat{e}(g^\alpha, T^{\tilde{s}}), \hat{C}_\Delta = T^{\tilde{s}} Z_\Delta, C_{\Delta,x} = T^{\tilde{s}aA_x\cdot\tilde{v}'} T^{((A_x\cdot v_\Delta)a\hat{s}_x - \tilde{s})(a_0 + t_{\rho(x)})} \tilde{Z}_{\Delta,x},\\ \tilde{C}_1 &= M_\beta \hat{e}(g^\alpha, T), \hat{C}_1 = T, C_{1,x} = T^{aA_x\cdot\tilde{v}} T^{-(a_0+t_{\rho(x)})\tilde{r}_x} \tilde{Z}_{c,x}, D_{1,x} = T^{\tilde{r}_x} Z_{d,x},\end{aligned}$$

where $1 \le x \le \ell$.

4) $\mathcal{B}$ sets the challenge ciphertext as $\mathsf{CT}_{\mathbb{A}_\beta}$ and sends it to $\mathcal{A}$:

$$\mathsf{CT}_{\mathbb{A}_\beta} = ((\mathbf{A}, \rho), \tilde{C}_\Delta, \hat{C}_\Delta, \{C_{\Delta,x}\}_{1\le x\le\ell}, \tilde{C}_1, \hat{C}_1, \{C_{1,x}, D_{1,x}\}_{1\le x\le\ell})$$

If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$, let $T = g^s g_2^c$, then $\tilde{C}_\Delta = Y^{s'}, \hat{C}_\Delta = g^{s'} Z_\Delta g_2^{c'}$,

$$C_{\Delta,x} = g^{aA_x\cdot v'} (g^{t_{\rho(x)}} H)^{-s'} Z_{\Delta,x} g_2^{A_x\cdot\omega' + \gamma'_x z_{\rho(x)}},$$

where $s' = s\tilde{s}$, $c' = c\tilde{s}$, $v' = (s\tilde{s})\tilde{v}' + sv_\Delta$.

We know $v'_1 = s'$, $z_{\rho(x)} = a_0 + t_{\rho(x)}$, $Z_{\Delta,x} = Z^{s'} \tilde{Z}_{\Delta,x}$, $\omega' = c\tilde{s}a\tilde{v}'$, $\gamma'_x = c((A_x \cdot v_\Delta)a\hat{s}_x - \tilde{s})$. Besides, $\tilde{C}_1 = M_\beta Y^s, \hat{C}_1 = g^s g_2^c$, and

$$\begin{aligned}C_{1,x} &= g^{aA_x\cdot v} (g^{t_{\rho(x)}} H)^{-r_x} Z_{c,x} g_2^{A_x\cdot\omega + \gamma_x z_{\rho(x)}},\\ D_{1,x} &= T^{\tilde{r}_x} Z_{d,x} = g^{r_x} Z_{d,x} g_2^{-\gamma_x},\end{aligned}$$

where $v = s\tilde{v}$ with $v_1 = s$, $r_x = s\tilde{r}_x$, $Z_{c,x} = Z^{s\tilde{r}_x} \tilde{Z}_{c,x}$, $z_{\rho(x)} = a_0 + t_{\rho(x)}$, $\omega = ca\tilde{v}$, $\gamma_x = -c\tilde{r}_x$. Hence, the challenge ciphertext is semi-functional and $\mathcal{B}$ simulates Game_0. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1}$, it is a normal ciphertext and $\mathcal{B}$ simulates $\mathsf{Game}_{\mathrm{Real}}$.

Phase 2. $\mathcal{B}$ does as in **Phase 1** with the restriction that none of queried attribute sets satisfies $\mathbb{A}_1$ and $\mathbb{A}_2$. Note that, if $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$, then $\mathcal{B}$ simulates Game_0. If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{\mathrm{Real}}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish T and $\mathsf{Adv1}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$. ∎

B.2. Proof of Lemma 3

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

Proof. Suppose there exists an adversary $\mathcal{A}$ such that $|\mathsf{Game}_{k-1,3}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_{k,1}\mathsf{Adv}_{\mathcal{A}}| = \epsilon$, then we can construct a simulator $\mathcal{B}$ with $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$ in breaking Assumption 2. $\mathcal{B}$ is given $g, X_1X_2, Y_2Y_3, X_3, X_4, T$ and will simulate $\mathsf{Game}_{k-1,3}$ or $\mathsf{Game}_{k,1}$ with $\mathcal{A}$.

Setup. $\mathcal{B}$ uniformly chooses $\alpha, a, a_0 \in_R \mathbb{Z}_N$ and $Z \in_R \mathbb{G}_{p_4}$. It then sets $Y = \hat{e}(g,g)^{\alpha}, h = g^{a_0}, H = hZ$, and sends $\mathcal{A}$ the system public parameters $\mathsf{PK} = (N, g, g^a, Y, H, X_4)$. Note that $\mathcal{B}$ knows the master key $\mathsf{MK} = (\alpha, h, X_3)$.

Phase 1. Let us now explain how $\mathcal{B}$ answers the j-th secret key query for $\mathcal{S} = (\mathcal{I}_S, S)$ with $S = \{s_i\}_{i\in\mathcal{I}_S}$.

- For $j < k$, $\mathcal{B}$ chooses $t, \tilde{d}, \tilde{d}' \in_R \mathbb{Z}_N$ and $\{\tilde{d}_i \in_R \mathbb{Z}_N\}_{i\in\mathcal{I}_S}$, then creates a semi-functional key of type 3 as follows:

$$K = g^\alpha g^{at} (Y_2Y_3)^{\tilde{d}}, K' = g^t (Y_2Y_3)^{\tilde{d}'}, \{K_i = (g^{s_i} h)^t (Y_2Y_3)^{\tilde{d}_i}\}_{i\in\mathcal{I}_S}.$$

  Note that this is a properly distributed semi-functional key of type 3 because the values of $\tilde{d}, \tilde{d}', \tilde{d}_i$ modulo p_2 are uncorrelated to their values modulo p_3.
- For $j > k$, $\mathcal{B}$ can generate normal keys by using the key generation algorithm, since it knows the master key $\mathsf{MK} = (\alpha, h, X_3)$.
- To answer the k-th secret key query, $\mathcal{B}$ chooses $\tilde{R}, \tilde{R}', \tilde{R}_i \in_R \mathbb{G}_{p_3}$ for $i \in \mathcal{I}_S$ and calculates $K = g^\alpha T^a \tilde{R}, K' = T\tilde{R}', \{K_i = T^{a_0+s_i}\tilde{R}_i\}_{i\in\mathcal{I}_S}$. We observe that
  - If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, let $T = g^t g_2^{d'} \hat{R}$, then

$$K = g^\alpha g^{at} R g_2^d, K' = g^t R' g_2^{d'}, \{K_i = (g^{s_i} h)^t R_i g_2^{d' z_i}\}_{i\in\mathcal{I}_S},$$

    where $R = \hat{R}^a \tilde{R}$, $d = ad'$, $R' = \hat{R}\tilde{R}'$, $R_i = \hat{R}^{a_0+s_i}\tilde{R}_i$, $z_i = a_0 + s_i$. It is a semi-functional key of type 1. Note that the values of a, a_0, s_i modulo p_1 are uncorrelated to their values modulo p_2.
  - If $T \overset{\$}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, it is a properly distributed normal key.

Challenge. Once $\mathcal{A}$ submits to $\mathcal{B}$ two messages M_0, M_1 of equal length and two access structures $\mathbb{A}_1 = (\mathbf{A}, \rho, \mathcal{T}_0)$, $\mathbb{A}_2 = (\mathbf{A}, \rho, \mathcal{T}_1)$ with the restriction that $\mathbb{A}_1$ and $\mathbb{A}_2$ cannot be satisfied by any of the queried attribute sets in **Phase 1**. Let $\mathcal{T}_\beta = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ and does the following.

1) $\mathcal{B}$ creates $\tilde{v} = (1, \tilde{v}_2, \ldots, \tilde{v}_n)$, $\tilde{v}' = (1, \tilde{v}'_2, \ldots, \tilde{v}'_n)$ and $v_\Delta = (0, v_{\Delta,2}, \ldots, v_{\Delta,n})$ with $\tilde{v}_i, \tilde{v}'_i, v_{\Delta,i} \in_R \mathbb{Z}_N$ for $2 \le i \le n$.
2) $\mathcal{B}$ chooses $\tilde{r}_x \in_R \mathbb{Z}_N$, $\hat{s}_x = (a_0 + t_{\rho(x)})^{-1}$ and $Z_\Delta, \tilde{Z}_{\Delta,x}, \tilde{Z}_{c,x}, Z_{d,x} \in_R \mathbb{G}_{p_4}$, for $1 \le x \le \ell$.
3) $\mathcal{B}$ chooses $\tilde{s} \in \mathbb{Z}_N$ and calculates

$$\begin{aligned}\tilde{C}_\Delta &= \hat{e}(g^\alpha, (X_1X_2)^{\tilde{s}}), \hat{C}_\Delta = (X_1X_2)^{\tilde{s}} Z_\Delta,\\ C_{\Delta,x} &= (X_1X_2)^{\tilde{s}aA_x\cdot\tilde{v}'} (X_1X_2)^{((A_x\cdot v_\Delta)a\hat{s}_x - \tilde{s})(a_0 + t_{\rho(x)})} \tilde{Z}_{\Delta,x},\\ \tilde{C}_1 &= M_\beta \hat{e}(g^\alpha, (X_1X_2)), \hat{C}_1 = (X_1X_2),\\ C_{1,x} &= (X_1X_2)^{aA_x\cdot\tilde{v}} (X_1X_2)^{-(a_0+t_{\rho(x)})\tilde{r}_x} \tilde{Z}_{c,x},\\ D_{1,x} &= (X_1X_2)^{\tilde{r}_x} Z_{d,x},\end{aligned}$$

   where $1 \le x \le \ell$.
4) $\mathcal{B}$ sets the challenge ciphertext as $\mathsf{CT}_{\mathbb{A}_\beta}$ and sends it to $\mathcal{A}$:

$$\mathsf{CT}_{\mathbb{A}_\beta} = ((\mathbf{A}, \rho), \tilde{C}_\Delta, \hat{C}_\Delta, \{C_{\Delta,x}\}_{1\le x\le\ell}, \tilde{C}_1, \hat{C}_1, \{C_{1,x}, D_{1,x}\}_{1\le x\le\ell}).$$

   Suppose $X_1X_2 = g^s g_2^c$, then $\tilde{C}_\Delta = Y^{s'}, \hat{C}_\Delta = g^{s'} Z_\Delta g_2^{c'}$, and

$$C_{\Delta,x} = = g^{aA_x\cdot v'} (g^{t_{\rho(x)}} H)^{-s'} Z_{\Delta,x} g_2^{A_x\cdot\omega' + \gamma'_x z_{\rho(x)}},$$

where $s' = s\tilde{s}$, $c' = c\tilde{s}$, $z_{\rho(x)} = a_0 + t_{\rho(x)}$, $v' = (s\tilde{s})\tilde{v}' + sv_\Delta$.

We know $v'_1 = s'$, $Z_{\Delta,x} = Z^{s'} \tilde{Z}_{\Delta,x}$, $\omega' = c\tilde{s}a\tilde{v}'$, $\gamma'_x = c((A_x \cdot v_\Delta)a\hat{s}_x - \tilde{s})$. Besides, $\tilde{C}_1 = M_\beta Y^s, \hat{C}_1 = g^s g_2^c$, and

$$\begin{aligned}C_{1,x} &= g^{aA_x\cdot v} (g^{t_{\rho(x)}} H)^{-r_x} Z_{c,x} g_2^{A_x\cdot\omega + \gamma_x z_{\rho(x)}},\\ D_{1,x} &= (X_1X_2)^{\tilde{r}_x} Z_{d,x} = g^{r_x} Z_{d,x} g_2^{-\gamma_x},\end{aligned}$$

where $v = s\tilde{v}$ with $v_1 = s$, $r_x = s\tilde{r}_x$, $Z_{c,x} = Z^{s\tilde{r}_x} \tilde{Z}_{c,x}$, $z_{\rho(x)} = a_0 + t_{\rho(x)}$, $\omega = ca\tilde{v}$, and $\gamma_x = -c\tilde{r}_x$. We know the

challenge ciphertext is semi-functional. Note that the values of $a, a_0, \{t_{\rho(x)}\}_{1\le x\le \ell}$, $\tilde{s}, \{\tilde{v}_i, \tilde{v}'_i\}_{2\le i\le n}$, $\{\tilde{r}_x\}_{1\le x\le \ell}$ modulo p_1 are uncorrelated to their values modulo p_2.

Phase 2. $\mathcal{B}$ does as in **Phase 1** with the restriction that none of queried attribute sets satisfies $\mathbb{A}_1$ and $\mathbb{A}_2$. Note that, if $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_3}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{k,1}$. If $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_3}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{k-1,3}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish T and $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$. ∎

B.3. Proof of Lemma 4

Lemma 4: Suppose that $\mathcal{G}$ satisfies Assumption 2. Then $\mathsf{Game}_{k,1}$ and $\mathsf{Game}_{k,2}$ are computationally indistinguishable.

Proof. Suppose there exists an adversary $\mathcal{A}$ such that $|\mathsf{Game}_{k,1}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_{k,2}\mathsf{Adv}_{\mathcal{A}}| = \epsilon$, then we can construct a simulator $\mathcal{B}$ with $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$ in breaking Assumption 2. $\mathcal{B}$ is given $g, X_1X_2, Y_2Y_3, X_3, X_4, T$ and will simulate $\mathsf{Game}_{k,1}$ or $\mathsf{Game}_{k,2}$ with $\mathcal{A}$.

Setup. $\mathcal{B}$ uniformly chooses $\alpha, a, a_0 \in_R \mathbb{Z}_N$ and $Z \in_R \mathbb{G}_{p_4}$. It then sets $Y = \hat{e}(g,g)^{\alpha}, h = g^{a_0}, H = hZ$, and sends $\mathcal{A}$ the system public parameters $\mathsf{PK} = (N, g, g^a, Y, H, X_4)$. Note that $\mathcal{B}$ knows the master key $\mathsf{MK} = (\alpha, h, X_3)$.

Phase 1. Let us now explain how $\mathcal{B}$ answers the j-th secret key query for $\mathcal{S} = (\mathcal{I}_S, S)$ with $S = \{s_i\}_{i\in\mathcal{I}_S}$. The first $k-1$ semi-functional keys of type 3 and the normal keys with $j > k$ are constructed exactly as in Lemma 3.

To answer the k-th key quest, $\mathcal{B}$ does as it did in Lemma 3, but $\mathcal{B}$ additionally chooses $\tau \in_R \mathbb{Z}_N$ and sets $K = g^{\alpha}T^a\tilde{R}(Y_2Y_3)^{\tau}, K' = T\tilde{R}', \{K_i = T^{a_0+s_i}\tilde{R}_i\}_{i\in\mathcal{I}_S}$. The only change we have made here is adding the $(Y_2Y_3)^{\tau}$ term, which randomizes the $\mathbb{G}_{p_2}$ part of K. Obviously, if $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 1. If $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 2.

Challenge. The same as Lemma 3.

Phase 2. $\mathcal{B}$ does as in **Phase 1** with the restriction that none of queried attribute sets satisfies $\mathbb{A}_1$ and $\mathbb{A}_2$. Hence, if $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_3}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{k,1}$. If $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_3}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{k,2}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish T and $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$. ∎

B.4. Proof of Lemma 5

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

Proof. Suppose there exists an adversary $\mathcal{A}$ such that $|\mathsf{Game}_{k,2}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_{k,3}\mathsf{Adv}_{\mathcal{A}}| = \epsilon$, then we can construct a simulator $\mathcal{B}$ with $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$ in breaking Assumption 2. $\mathcal{B}$ is given $g, X_1X_2, Y_2Y_3, X_3, X_4, T$ and will simulate $\mathsf{Game}_{k,1}$ or $\mathsf{Game}_{k,2}$ with $\mathcal{A}$.

Setup. $\mathcal{B}$ uniformly chooses $\alpha, a, a_0 \in_R \mathbb{Z}_N$ and $Z \in_R \mathbb{G}_{p_4}$. It then sets $Y = \hat{e}(g,g)^{\alpha}, h = g^{a_0}, H = hZ$, and sends $\mathcal{A}$ the system public parameters $\mathsf{PK} = (N, g, g^a, Y, H, X_4)$.

Phase 1. Let us now explain how $\mathcal{B}$ answers the j-th secret key query for $\mathcal{S} = (\mathcal{I}_S, S)$ with $S = \{s_i\}_{i\in\mathcal{I}_S}$. The first $k-1$ semi-functional keys of type 3 and the normal keys with $j > k$ are constructed exactly as in Lemma 3.

Let us now explain how $\mathcal{B}$ answers the k-th secret key query. $\mathcal{B}$ chooses exponents $r, \tau \in_R \mathbb{Z}_N$, $\tilde{R}, \tilde{R}', \tilde{R}_i \in_R \mathbb{G}_{p_3}$ for $i \in \mathcal{I}_S$ and calculates

$$K = g^{\alpha}T^{ra}\tilde{R}(Y_2Y_3)^{\tau}, K' = T^r\tilde{R}', \{K_i = T^{(a_0+s_i)r}\tilde{R}_i\}_{i\in\mathcal{I}_S}.$$

If $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_3}$, let $T = g^{t'}g_2^{\tilde{d}}\hat{R}$, then

$$K = g^{\alpha}g^{at}Rg_2^{d}, K' = g^tR'g_2^{d'}, \{K_i = (g^{s_i}h)^tR_ig_2^{d_i}\}_{i\in\mathcal{I}_S},$$

where $t = rt'$, $g_2^d = g_2^{ar\tilde{d}}Y_2^{\tau}$, $R = \hat{R}^{ar}\tilde{R}Y_3^{\tau}$, $R' = \hat{R}^r\tilde{R}'$, $d' = r\tilde{d}$, $R_i = \hat{R}^{r(a_0+s_i)}\tilde{R}_i$, $d_i = r(a_0+s_i)\tilde{d}$.

This is a semi-functional key of type 3. Note that the value of τ modulo p_2 is uncorrelated to its value modulo p_3. If $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_3}$, this is a properly distributed semi-functional key of type 2.

Challenge. The same as Lemma 3.

Phase 2. $\mathcal{B}$ does as in **Phase 1** with the restriction that none of queried attribute sets satisfies $\mathbb{A}_1$ and $\mathbb{A}_2$. Hence, if $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_2}\times\mathbb{G}_{p_3}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{k,3}$. If $T \xleftarrow{\$} \mathbb{G}_{p_1}\times\mathbb{G}_{p_3}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{k,2}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish T and $\mathsf{Adv2}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$. ∎

B.5. Proof of Lemma 6

Lemma 6: Suppose $\mathcal{G}$ satisfies Assumption 3. Then $\mathsf{Game}_{q,3}$ and $\mathsf{Game}_{\mathrm{Final}_0}$ are computationally indistinguishable.

Proof. Suppose there exists an adversary $\mathcal{A}$ such that $|\mathsf{Game}_{q,3}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_{\mathrm{Final}_0}\mathsf{Adv}_{\mathcal{A}}| = \epsilon$, then we can construct a simulator $\mathcal{B}$ with $\mathsf{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$ in breaking Assumption 3. $\mathcal{B}$ is given $g, g_2, g^{\alpha}X_2, g^sY_2, X_3, X_4, T$ and will simulate $\mathsf{Game}_{q,3}$ or $\mathsf{Game}_{\mathrm{Final}_0}$ with $\mathcal{A}$.

Setup. $\mathcal{B}$ uniformly chooses $a, a_0 \in_R \mathbb{Z}_N$ and $Z \in_R \mathbb{G}_{p_4}$. It then sets $Y = \hat{e}(g, g^{\alpha}X_2), h = g^{a_0}, H = hZ$, and sends $\mathcal{A}$ the system public parameters $\mathsf{PK} = (N, g, g^a, Y, H, X_4)$.

Phase 1. Each time $\mathcal{B}$ is asked to provide a key for $\mathcal{S} = (\mathcal{I}_S, S)$ with $S = \{s_i\}_{i\in\mathcal{I}_S}$, $\mathcal{B}$ chooses $t, \tilde{d}, d' \in_R \mathbb{Z}_N$, $\{d_i \in_R \mathbb{Z}_N\}_{i\in\mathcal{I}_S}$, and $R, R', R_i \in_R \mathbb{G}_{p_3}$ for $i \in \mathcal{I}_S$, then creates a semi-functional key of type 3 as follows:

$$K = (g^{\alpha}X_2)g^{at}Rg_2^{\tilde{d}} = g^{\alpha}g^{at}Rg_2^{d}, K' = g^tR'g_2^{d'}, \{K_i = (g^{s_i}h)^tR_ig_2^{d_i}\}_{i\in\mathcal{I}_S},$$

where $g_2^d = X_2g_2^{\tilde{d}}$.

Challenge. Once $\mathcal{A}$ submits to $\mathcal{B}$ two messages M_0, M_1 of equal length and two access structures $\mathbb{A}_1 = (\mathbf{A}, \rho, \mathcal{T}_0)$, $\mathbb{A}_2 = (\mathbf{A}, \rho, \mathcal{T}_1)$ with the restriction that $\mathbb{A}_1$ and $\mathbb{A}_2$ cannot be satisfied by any of the queried attribute sets in **Phase 1**. Let $\mathcal{T}_{\beta} = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ and does the following.

1) $\mathcal{B}$ creates $\tilde{v} = (1, \tilde{v}_2, \ldots, \tilde{v}_n)$ with $\tilde{v}_i \in_R \mathbb{Z}_N$ for $2 \le i \le n$, and chooses $v' = (s', v'_2, \ldots, v'_n), \omega' = (\omega'_1, \omega'_2, \ldots, \omega'_n) \in_R \mathbb{Z}_N^n$.
2) $\mathcal{B}$ chooses $\tilde{r}_x, \gamma'_x \in_R \mathbb{Z}_N$ and $Z_{\Delta}, Z_{\Delta,x}, \tilde{Z}_{c,x}, Z_{d,x} \in_R \mathbb{G}_{p_4}$, for $1 \le x \le \ell$.
3) Let $\mathcal{T}_{\beta} = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses $c' \in \mathbb{Z}_N$, calculates $\tilde{C}_{\Delta} = Y^{s'}, \hat{C}_{\Delta} = g^{s'}Z_{\Delta}g_2^{c'}$, and

$$C_{\Delta,x} = g^{aA_x\cdot v'}(g^{t_{\rho(x)}}H)^{-s'}Z_{\Delta,x}g_2^{A_x\cdot\omega'+\gamma'_x(a_0+t_{\rho(x)})},$$

$$\tilde{C}_1 = M_{\beta}T, \hat{C}_1 = g^sY_2, C_{1,x} = (g^sY_2)^{aA_x\cdot\tilde{v}}(g^sY_2)^{-(a_0+t_{\rho(x)})\tilde{r}_x}\tilde{Z}_{c,x},$$

$$D_{1,x} = (g^sY_2)^{\tilde{r}_x}Z_{d,x},$$

where $1 \le x \le \ell$.

4) $\mathcal{B}$ sets the challenge ciphertext as $\mathsf{CT}_{\mathbb{A}_\beta}$ and sends it to $\mathcal{A}$:

$$\mathsf{CT}_{\mathbb{A}_\beta} = ((\mathbf{A}, \rho), \tilde{C}_\Delta, \hat{C}_\Delta, \{C_{\Delta,x}\}_{1\le x\le \ell}, \tilde{C}_1, \hat{C}_1, \{C_{1,x}, D_{1,x}\}_{1\le x\le \ell}).$$

Suppose $g^s Y_2 = g^s g_2^c$, then $\tilde{C}_\Delta = Y^{s'}, \hat{C}_\Delta = g^{s'} Z_\Delta g_2^{c'}$, and

$$\begin{aligned} C_{\Delta,x} &= g^{aA_x\cdot v'}(g^{t_{\rho(x)}}H)^{-s'} Z_{\Delta,x} g_2^{A_x\cdot\omega' + \gamma'_x z_{\rho(x)}}, \\ \tilde{C}_1 &= M_\beta T, \hat{C}_1 = g^s g_2^c, C_{1,x} = g^{aA_x\cdot v}(g^{t_{\rho(x)}}H)^{-r_x} Z_{c,x} g_2^{A_x\cdot\omega + \gamma_x z_{\rho(x)}}, \\ D_{1,x} &= (g^s Y_2)^{\tilde{r}_x} Z_{d,x} = g^{r_x} Z_{d,x} g_2^{-\gamma_x}, \end{aligned}$$

where $v = s\tilde{v}$, $r_x = s\tilde{r}_x$, $Z_{c,x} = Z^{r_x}\tilde{Z}_{c,x}$, $\omega = ca\tilde{v}$, $\gamma_x = -c\tilde{r}_x$, and $z_{\rho(x)} = a_0 + t_{\rho(x)}$. Note that the values of $a, a_0, \{t_{\rho(x)}\}_{1\le x\le\ell}, \{\tilde{v}_i\}_{2\le i\le n}, \{\tilde{r}_x\}_{1\le x\le\ell}$ modulo p_1 are uncorrelated to their values modulo p_2.

Phase 2. $\mathcal{B}$ does as in **Phase 1** with the restriction that none of queried attribute sets satisfies $\mathbb{A}_1$ and $\mathbb{A}_2$. Note that, if $T = \hat{e}(g,g)^{\alpha s}$, the challenge ciphertext is a properly distributed semi-functional encryption of M_β and $\mathcal{B}$ simulates $\mathsf{Game}_{q,3}$. Otherwise, it is a properly distributed semi-functional encryption of a random message in $\mathbb{G}_T$ and $\mathcal{B}$ simulates $\mathsf{Game}_{\mathrm{Final}_0}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish T and $\mathsf{Adv3}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$. ∎

B.6. Proof of Lemma 7

Lemma 7: Suppose $\mathcal{G}$ satisfies Assumption 4. Then $\mathsf{Game}_{\mathrm{Final}_0}$ and $\mathsf{Game}_{\mathrm{Final}_1}$ are computationally indistinguishable.

Proof. Suppose there exists an adversary $\mathcal{A}$ such that $|\mathsf{Game}_{\mathrm{Final}_0}\mathsf{Adv}_{\mathcal{A}} - \mathsf{Game}_{\mathrm{Final}_1}\mathsf{Adv}_{\mathcal{A}}| = \epsilon$, then we can construct a simulator $\mathcal{B}$ with $\mathsf{Adv4}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$ in breaking Assumption 4. $\mathcal{B}$ is given $g, g_2, g^{t'}B_2, h^{t'}Y_2, X_3, X_4, hZ, g^{r'}D_2D_4, T$ and will simulate $\mathsf{Game}_{\mathrm{Final}_0}$ or $\mathsf{Game}_{\mathrm{Final}_1}$ with $\mathcal{A}$.

Setup. $\mathcal{B}$ uniformly chooses $\alpha, a \in_R \mathbb{Z}_N$. It then sets $Y = \hat{e}(g,g)^\alpha, H = hZ$, and sends $\mathcal{A}$ the system public parameters $\mathsf{PK} = (N, g, g^a, Y, H, X_4)$.

Phase 1. Each time $\mathcal{B}$ is asked to provide a key for $\mathcal{S} = (\mathcal{I}_S, S)$ with $S = \{s_i\}_{i\in\mathcal{I}_S}$, $\mathcal{B}$ chooses $\tilde{t} \in_R \mathbb{Z}_N$, and $R, R', R_i \in_R \mathbb{G}_{p_3}$ for $i \in \mathcal{I}_S$, then creates a semi-functional key of type 3 as follows:

$$K = (g^\alpha g^{t'}B_2)^{a\tilde{t}} R, K' = (g^{t'}B_2)^{\tilde{t}} R', \{K_i = (g^{t'}B_2)^{s_i\tilde{t}}(h^{t'}Y_2)^{\tilde{t}} R_i\}_{i\in\mathcal{I}_S}.$$

We observe that $K = g^\alpha g^{at} R g_2^d, K' = g^t R' g_2^{d'}, \{K_i = (g^{s_i}h)^t R_i g_2^{d_i}\}_{i\in\mathcal{I}_S}$, where $t = t'\tilde{t}$, $g_2^d = B_2^{a\tilde{t}}$, $g_2^{d'} = B_2^{\tilde{t}}$, $g_2^{d_i} = B_2^{s_i\tilde{t}} Y_2^{\tilde{t}}$. Note that it is a properly distributed semi-functional key of type 3 in that the values of $a, \tilde{t}, \{s_i\}_{1\le i\le n}$ modulo p_1 are uncorrelated to their values modulo p_2.

Challenge. Once $\mathcal{A}$ submits to $\mathcal{B}$ two messages M_0, M_1 of equal length and two access structures $\mathbb{A}_1 = (\mathbf{A}, \rho, \mathcal{T}_0)$, $\mathbb{A}_2 = (\mathbf{A}, \rho, \mathcal{T}_1)$ with the restriction that $\mathbb{A}_1$ and $\mathbb{A}_2$ cannot be satisfied by any of the queried attribute sets in **Phase 1**. Let $\mathcal{T}_\beta = (t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(\ell)})$. $\mathcal{B}$ chooses $\beta \in \{0,1\}$ and does the following.

1) $\mathcal{B}$ chooses $v = (s, v_2, \ldots, v_n), \tilde{v}' = (s', \tilde{v}'_2, \ldots, \tilde{v}'_n), \tilde{v}_\Delta = (0, \tilde{v}_{\Delta,2}, \ldots, \tilde{v}_{\Delta,n})$, and $\omega, \tilde{\omega}' \in_R \mathbb{Z}_N^n$.
2) $\mathcal{B}$ chooses $\tilde{r}_x \in_R \mathbb{Z}_N$, $\hat{s}_x = t_{\rho(x)}^{-1}$, $Z_\Delta, \tilde{Z}_{\Delta,x}, \tilde{Z}_{c,x} \in_R \mathbb{G}_{p_4}$, for $1 \le x \le \ell$.
3) $\mathcal{B}$ chooses $c, c' \in \mathbb{Z}_N$ and calculates for $1 \le x \le \ell$:

$$\begin{aligned} \tilde{C}_\Delta &= Y^{s'}, \hat{C}_\Delta = g^{s'} Z_\Delta g_2^{c'}, \\ C_{\Delta,x} &= g^{aA_x\cdot\tilde{v}'}(g^{r'}D_2D_4)^{(A_x\cdot v_\Delta)a} T^{a\hat{s}_x(A_x\cdot v_\Delta)t_{\rho(x)}} (g^{t_{\rho(x)}}H)^{-s'} \tilde{Z}_{\Delta,x} g_2^{A_x\cdot\tilde{\omega}'}, \\ \tilde{C}_1 &\xleftarrow{\$} \mathbb{G}_T, \hat{C}_1 = g^s g_2^c, C_{1,x} = g^{aA_x\cdot v}(g^{r'}D_2D_4)^{-\tilde{r}_x t_{\rho(x)}} T^{-\tilde{r}_x} g_2^{A_x\cdot\omega} \tilde{Z}_{c,x}, \\ D_{1,x} &= (g^{r'}D_2D_4)^{\tilde{r}_x}. \end{aligned}$$

4) $\mathcal{B}$ sets the challenge ciphertext as $\mathsf{CT}_{\mathbb{A}_\beta}$ and sends it to $\mathcal{A}$:

$$\mathsf{CT}_{\mathbb{A}_\beta} = ((\mathbf{A}, \rho), \tilde{C}_\Delta, \hat{C}_\Delta, \{C_{\Delta,x}\}_{1\le x\le \ell}, \tilde{C}_1, \hat{C}_1, \{C_{1,x}, D_{1,x}\}_{1\le x\le \ell}).$$

If $T = h^{r'}A_2A_4$, suppose $h = g^{\tau_1}$, $D_2 = g_2^\gamma$, $A_2 = g_2^{\gamma\tau_2}$ with $\tau_1, \tau_2, \gamma \in \mathbb{Z}_N$. Then $\tilde{C}_\Delta = Y^{s'}, \hat{C}_\Delta = g^{s'} Z_\Delta g_2^{c'}$, $C_{\Delta,x} = g^{aA_x\cdot v'}(g^{t_{\rho(x)}}H)^{-s'} g_2^{A_x\cdot\omega' + \gamma'_x z_{\rho(x)}} Z_{\Delta,x}$, where $v' = \tilde{v}' + v_\Delta(r' + r'\tau_1)$.

We know that $v'_1 = s'$, $\gamma'_x = (A_x \cdot v_\Delta)\gamma\tau_2 a\hat{s}_x(1 - \tau_2(t_{\rho(x)} + \tau_2)^{-1})$, $z_{\rho(x)} = \tau_2 + t_{\rho(x)}$, $Z_{\Delta,x} = D_4^{(A_x\cdot v_\Delta)a} A_4^{(A_x\cdot v_\Delta)a} \tilde{Z}_{\Delta,x}$, $\omega' = \tilde{\omega}' + a\gamma v_\Delta$. Besides, $\tilde{C}_1 \xleftarrow{\$} \mathbb{G}_T, \hat{C}_1 = g^s g_2^c$, and $C_{1,x} = g^{aA_x\cdot v}(g^{t_{\rho(x)}}H)^{-r_x} g_2^{A_x\cdot\omega + \gamma_x z_{\rho(x)}} Z_{c,x}$, $D_{1,x} = (g^{r'}D_2D_4)^{\tilde{r}_x} = g^{r_x} Z_{d,x} g_2^{-\gamma_x}$, where $r_x = r'\tilde{r}_x$, $\gamma_x = -\gamma\tilde{r}_x$, $z_{\rho(x)} = \tau_2 + t_{\rho(x)}$, $Z_{c,x} = D_4^{-\tilde{r}_x t_{\rho(x)}} A_4^{-\tilde{r}_x} Z'^{\tilde{r}_x} \tilde{Z}_{c,x}$, $Z_{d,x} = D_4^{\tilde{r}_x}$.

Phase 2. $\mathcal{B}$ does as in **Phase 1** with the restriction that none of queried attribute sets satisfies $\mathbb{A}_1$ and $\mathbb{A}_2$. Note that, if $T = h^{r'}A_2A_4$, the challenge ciphertext is a properly distributed semi-functional encryption of a random message in $\mathbb{G}_T$ and $\mathcal{B}$ simulates $\mathsf{Game}_{\mathrm{Final}_0}$. Otherwise, if $T \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$, the challenge ciphertext component $\tilde{C}_1$ is a random element in $\mathbb{G}_T$ and $C_{\Delta,x}, C_{1,x}$ are random elements in $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{\mathrm{Final}_0}$. Finally, $\mathcal{B}$ can use the output of $\mathcal{A}$ to distinguish T and $\mathsf{Adv4}_{\mathcal{G},\mathcal{A}}(\lambda) = \epsilon$. ∎