Unified Fine-grained Access Control for Personal Health Records in Cloud Computing

Article *in* IEEE Journal of Biomedical and Health Informatics · June 2018

DOI: 10.1109/JBHI.2018.2850304

7 authors, including:

Dongxi Liu
Shanghai Maritime University
62 PUBLICATIONS **466** CITATIONS
SEE PROFILE

Ren Ping Liu
CSIRO, Macquarie University, University of Technology Sydney
149 PUBLICATIONS **1,289** CITATIONS
SEE PROFILE

Wei Ni
The Commonwealth Scientific and Industrial Research Organisation
157 PUBLICATIONS **608** CITATIONS
SEE PROFILE

Unified Fine-grained Access Control for Personal Health Records in Cloud Computing

Wei Li, Bonnie M. Liu, Dongxi Liu, Ren Ping Liu, *Senior Member, IEEE,* Peishun Wang, Shoushan Luo, and Wei Ni, *Senior Member, IEEE*

***Abstract*— Attribute-based encryption has been a promising encryption technology to secure personal health records (PHRs) sharing in cloud computing. PHRs consist of the patient data often collected from various sources including hospitals and general practice centres. Different patients' access policies have a common access subpolicy. In this paper, we propose a novel attribute-based encryption scheme for fine-grained and flexible access control to PHRs data in cloud computing. The scheme generates shared information by the common access subpolicy which is based on different patients' access policies. Then the scheme combines the encryption of PHRs from different patients. Therefore, both time consumption of encryption and decryption can be reduced. Medical staff require varying levels of access to PHRs. The proposed scheme can also support multi-privilege access control so that medical staff can access the required level of information while maximizing patient privacy. Through implementation and simulation, we demonstrate that the proposed scheme is efficient in terms of time. Moreover, We prove the security of the proposed scheme based on security of the ciphertext-policy attribute-based encryption scheme.**

***Index Terms*— Access control, attribute-based encryption, cloud computing, personal health records.**

I. Introduction

In recent years, with the rapid development of information technology and communication networks, these new technologies have been widely applied in the field of healthcare. Provision of health services using information technology has been termed e-health [1]. The development of e-health solutions made it possible to evolve from traditional paper-based medical records towards more efficient electronic health records. Personal Health Records(PHRs) is such an electronic version of patient health information [2]. In an electronic medical system, patients can share their PHRs with medical staff for monitoring and diagnosis [3]. The requirements for storage and continuous availability of PHRs necessitate the use of the cloud computing services [4], [5]. Cloud computing [6], [7] is one of the most promising applications to provide a more efficient way of data storage and computing power. Voluminous PHRs are stored in cloud computing platforms which are operated by cloud service provider (CSP). As the PHRs include the sensitive information, patient privacy should be guaranteed simultaneously during sharing of PHRs with others. In order to facilitate efficient and secure PHRs sharing in cloud computing environments, many research efforts have been focused on this problem [8]–[11].

The notion of attribute-based encryption (ABE) was first proposed in 2005 [12]. In an ABE scheme, ciphertexts are not encrypted for one specific user as in traditional public key encryption scheme. Both ciphertexts and users' secret keys are associated with a set of attributes or an access policy over the attributes of the users. A user is granted its secret key that permits it to decrypt a ciphertext if and only if there is a match between its secret key and the ciphertext. ABE is a promising cryptographic primitive which has been applied to cloud storage services due to its one-to-many, fine-grained, and flexible access control. ABE includes two categories, namely, ciphertext-policy ABE (CP-ABE) [13] and key-policy ABE (KP-ABE) [14]. In CP-ABE, each ciphertext is combined with an access policy describing who is entitled to decrypt it. Access policies are typically expressed by threshold gates and attributes. The attributes are embedded into the users' secret keys. KP-ABE inverts the relationship between ciphertext and secret key, i.e., the ciphertext is combined with attributes and users' secret keys have access policies embedded. CP-ABE is more flexible and appropriate for PHRs sharing than KP-ABE in practice, because it enables patients to specify an access policy over the attributes of data users.

In an electronic medical system, there are numerous patients, who share PHRs with various medical staff. A typical PHR file uses Continuity of Care Record (CCR) [15] which is a standard data format based on XML data structure. An individual patients PHR data can be divided into different classes [16] such as identification sheet, medication record, progress notes and so on. These classes of data are associated

Manuscript received December 18, 2017. This work was supported in part by the National Key R&D Program of China under Grant 2017YFB0802300, in part by the National High-tech R&D Program of China (863 Program) under Grant 2015AA016005 and Grant 2015AA017201, in part by Applied Sci-Tech R&D Special Fund Program of Guangdong Province under Grant 2015B010131007, and in part by China Scholarship Council under Grant 201506470040.

W. Li, S. Luo are with the Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: weilihero@bupt.edu.cn; buptlou@263.net).

B. M. Liu is with Royal North Shore Hospital, Sydney, NSW 2065, Australia (e-mail: bonnie.m.liu@hotmail.com).

D. Liu, W. Ni are with CSIRO, Sydney, NSW 2122, Australia (e-mail: dongxi.liu@data61.csiro.au; wei.ni@data61.csiro.au).

R. P. Liu is with the Global Big Data Technologies Centre, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: renping.liu@uts.edu.au).

P. Wang is with IBM, Sydney, Australia. The views published in the paper are mine only. (e-mail: wang_peishun@hotmail.com).

with different privilege levels. Medical staff who have a higher privilege can access more classes of data than those who have a lower privilege. For example, a patient's data consist of L classes, $\{m_1, m_2, \cdots, m_L\}$ denotes the data, these classes correspond to L privilege levels $\{p_1, p_2, \cdots, p_L\}$. A medical worker who has the privilege p_i can access the first i classes $\{m_1, m_2, \cdots, m_i\}$. p_1 is the lowest privilege that can only access one class m_1.

A. Problem Statement

In real world applications [18]–[20], different patients may share some classes of their PHRs with medical workers who have the same attributes. In other words, the access policies of the patients have a common sub-policy. To clarify this point, we take a simple example.

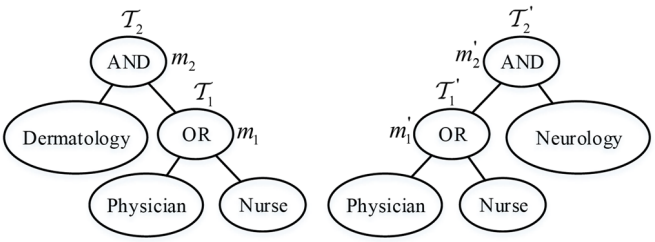


Fig. 1. Two patients have $\{m_1, m_2\}$ and $\{m_1', m_2'\}$ as PHRs separately. $\mathcal{T}_2, \mathcal{T}_1, \mathcal{T}_2'$ and $\mathcal{T}_1'$ are the access policies of m_2, m_1, m_2' and m_1'.

As illustrated in Fig. 1, $\{m_1, m_2\}$ is a patient's PHR and $\{m_1', m_2'\}$ is another patient's PHR. The access policies of m_1 and m_2 are $\mathcal{T}_1$\{"Physician" OR "Nurse"\} and $\mathcal{T}_2$\{"Dermatology" AND ("Physician" OR "Nurse")\}. $\mathcal{T}_1'$\{"Physician" OR "Nurse"\} and $\mathcal{T}_2'$\{"Neurology" AND ("Physician" OR "Nurse")\} are the access policies for m_1' and m_2'. In terms of policy, m_1 and m_1', such as patient's basic information can be accessed by all the physicians and nurses. m_2 and m_2' denotes the results of physical examination which can be accessed by staff from the two departments respectively. For each patient, the policies can be integrated into one policy and some efforts [6], [17] have been made to deliver the more efficient access control. Nevertheless, we can see that $\mathcal{T}_1$ is same with $\mathcal{T}_1'$. So there are some repetitive steps in the encryptions of m_1 and m_1' and this problem has not been considered before.

It is noted that this scenario is different from existing systems, where data access control is conducted separately for each individual data provider [6], [17], [21] or through an aggregate third-party platform using a publisher-subscriber model [22], [23].

B. Our Contributions

In this paper, we propose a new access control scheme for PHRs which can be provided by multiple patients. The scheme consists of ABE layer and symmetric key layer. In ABE layer, the scheme supports a multi-privilege access control for PHRs from multi-patients. The scheme combines the encryption of data from different patients where the data are under the same access policy to solve the problem of repetitive process in encryptions of these data, so that the cost of encryption and decryption can be reduced. The scheme achieves an efficient, flexible, and fine-grained access control on PHRs. In symmetric key layer, symmetric keys match medical workers' access privileges and the keys with higher privilege can derive keys with lower privilege, not the other way around. The patients encrypt each class of data with corresponding symmetric keys in symmetric key layer, and encrypt the symmetric keys in the ABE layer. We prove our scheme is secure based on security of CP-ABE. We also conduct comprehensive experiments for the proposed scheme, and the simulation results demonstrate that the scheme has low computation complexity on encryption and decryption. In addition, some preliminary results of this work has been reported in [24]. The work presented in that conference paper only considered one patient shares its PHR data with medical staff in cloud computing. We extend the scheme to support multi-patient sharing PHRs. We also provide a formal security proof for the scheme and carry out simulations to evaluate the performance of the proposed scheme in this paper.

C. Organization

The rest of this paper is arranged as follows. Section II introduces related work. In Section III, we present preliminaries which contain some definitions and notions. The system framework and security model are described in Section IV. Section V proposes our access control scheme. The security analysis of our scheme is given in Section VI. Simulation results are presented and analysed in Section VII. Finally, Section VIII concludes the paper and makes discussion for the future work.

II. Related Work

ABE is a cryptographic primitive which can provide fine-grained access control for encrypted data. Many research efforts have been devoted to improve the encryption efficiency and expand the application range of ABE. Hohenberger *et al.* [25] proposed an online/offline technique to reduce the encryption complexity. The encryption of ABE was split into the plaintext-independent offline pre-computation and the plaintext-dependent online computation. The offline pre-computation can produce intermediate ciphertext, which can be used with attributes to encrypt data online. However, this technique is only suitable for specific ABE schemes which have splittable algebraic structures, such as [26], [27]. Rouselakis *et al.* [26], proposed two practical large-universe ABE schemes by expanding the system from unbounded hierarchical identity-based encryption (HIBE) [28] and ABE schemes in to prime order settings. The schemes are based on CP-ABE and KP-ABE, respectively, and have a significant improvement of the efficiency over [27]. However, the two schemes are both selectively secure. This means the security is guaranteed only for messages that are fixed before the adversary interacts with the system [29]. This is too restrictive for many realistic applications. Han *et al.* [30], proposed a privacy-preserving decentralized ABE scheme, where all the decryption keys

of a user are tied to its global identifier (GID). Corrupted authorities cannot know the user's attributes by tracing the GID from the decryption keys. Unfortunately, two users can pool their decryption keys to generate an unauthorized user's decryption keys.

Some access control schemes based on ABE are applied in PHRs sharing in cloud computing environments. Li *et al.* [31] presented a patient-centric secure data sharing framework, where PHR was separately encrypted for personal and public domains using an existing access control scheme [3] and a new multiple attribute authorities ABE (MA-ABE), respectively. MA-ABE improves the scalability of the system but limits the expressibility of the access policy, because it only supports conjunctive policy across multiple AAs. The conjunctive policy has a format like $\{(a_{1,1}$ AND $\cdots$ AND $a_{1,n_1})$ AND $\cdots$ AND $(a_{m,1}$ OR $\cdots$ OR $a_{m,n_m})\}$, where $a_{i,j}$ denotes an attribute. The feature of the format is the root of the access tree must be an AND gate. Barua *et al.* [32] developed a patient-centric PHRs access control scheme based on CP-ABE for cloud storage. The scheme takes advantage of identity-based encryption (IBE) [33] for secure transmission of the data between patients and the clouds. The scheme also supports access to the PHRs in terms of the access privileges and can resist the denial of service attacks. Nevertheless, the scheme lacks of dynamicity and flexibility in PHRs acquisition and transmitting to the servers of CSP. This makes the scheme inefficient. Eom *et al.* [34] proposed a patient-controlled attribute-based encryption scheme that provides a patient with fine-grained access control over the PHR and efficient healthcare services. The scheme employs not only attribute authority but also identity authority to split the key generation process into two phases. The scheme shifts the validating attributes of patient to authorities to reduce the operational burden for the patient. Whereas, both of the two authorities must be fully trusted and this increases the security requirements.

Some PHRs sharing schemes which not based on ABE also have been proposed. Wu *et al.* [35] presented a broker-based authorization scheme which can share the combined PHRs from the multiple e-health clouds selectively. The scheme designs a hierarchical structure to represent PHRs, then retrieves and aggregates PHRs from various healthcare service providers and generates a composite PHR instance based on the hierarchical structure. Furthermore, a policy manager implements the access control policies. The scheme realizes a secure and efficient PHRs sharing across different healthcare service providers. Nevertheless, the access control policy infrastructure may meet up with the policy composition issues. Chadwick and Fatema [36] proposed a policy-based infrastructure to protect the patients' PHRs. The infrastructure allows patients set their own privacy policies that may indicate who has authority to access the PHRs and makes the policies stuck to the PHRs, then enforces the policies. The infrastructure also supports various policy languages that used to write the privacy policies. However, the proposed infrastructure does not avoid the need for trust as the PHRs is still vulnerable to disclosure threats by the CSP. Kondylakis *et al.* [37] developed a PHR system and on top of which implemented an approach for e-consent. The approach allowed partial release of PHR data at different levels of granularity and facilitated interoperability among different e-health systems. The PHR system also supported various scenarios in which patients can define complicated access control policies. However, all these aforementioned PHRs sharing works did not take into account the further improvement of efficiency when access policies of patients have a common sub-policy and most of these schemes were not able to support multi-privilege access control.

III. Preliminaries

In this section, the basic concepts of one-way hash function, bilinear maps and access tree are introduced to make the paper self-contained. The notations used in the paper are as listed in Table I.

TABLE I
MAJOR NOTATIONS USED IN THIS PAPER

Notation	Description
m_i	the i-th class of PHR data
p_i	the i-th privilege with p_1 as the lowest privilege
$\mathcal{T}_i$	the access tree corresponds to m_i
$\widehat{e}(\cdot)$	a bilinear map function
$\mathcal{G}_1$	the input group of bilinear map $\widehat{e}(\cdot)$
$\mathcal{G}_2$	the output group of bilinear map $\widehat{e}(\cdot)$
p	the order of $\mathcal{G}_1$, i.e., $\mathcal{G}_1$ has p elements
$\mathbb{Z}_p$	the set of integers $[0, p-1]$
g	the generator of $\mathcal{G}_1$. $\forall x \in \mathbb{Z}_p, g^x \in \mathcal{G}_1$
N_j	the j-th node in the access tree
num_j	the number of children of N_j
K_j	the threshold value of N_j
AA	attribute authority
a_i	the i-th attribute of a medical worker
k_i	the symmetric key for encrypting PHR data m_i
PK, MSK	the public key and master secret key of an ABE scheme
$\mathcal{S}_j$	a medical worker in the j-th class
SK_j	secret key of $\mathcal{S}_j$ in ABE, issued by AA
$\mathbb{A}_j$	the attribute set of $\mathcal{S}_j$
N_ϵ	the aggregate node of two access trees
ϵ	the highest class number of the common access tree
$\mathbb{L}(\mathcal{T}_i)$	the leaf node set of $\mathcal{T}_i$
$\mathbb{A}(\mathcal{T}_i)$	the attribute set of $\mathcal{T}_i$
$\perp$	termination signal
$\mathcal{E}(\cdot)$	the symmetric encryption function
$\mathcal{D}(\cdot)$	the symmetric decryption function
$\mathcal{H}_1(\cdot)$	the hash function for hashing an attribute
$\mathcal{H}_2(\cdot)$	the hash function for hashing a symmetric key
$x \,\|\| \, y$	the concatenation of string x and string y
M_j	the data set that $\mathcal{S}_j$ can access
$C(\mathcal{G}_i)$	the operation in group $\mathcal{G}_i$
$C_{\widehat{e}}$	the operation in bilinear map $\widehat{e}(\cdot)$
$B(*)$	the bit size of an element in $*$
$\|*\|$	the number of elements in $*$
$\mathbb{S}(\mathcal{T}_j)$	the least nodes set satisfy $\mathcal{T}_j$

A. One-Way Hash Function

A one-way hash function, denoted as $\mathcal{H}(\cdot)$, takes data of an arbitrary size as the input and outputs data of a fixed size.

The input is called a pre-image and the output is called a hash value. $\mathcal{H}(\cdot)$ has the property of one-wayness. In other words, given a pre-image x, it is easy to compute a hash value $y = \mathcal{H}(x)$; but, it is hard to find a pre-image x' from a given hash value y' such that $\mathcal{H}(x') = y'$. $\mathcal{H}(\cdot)$ is collision-resistant, this means it is hard to find two pre-images x and x' that can generate the same hash value, i.e., $\mathcal{H}(x) = \mathcal{H}(x')$. There are many one-way hash functions, such as SHA-256 [38].

B. Bilinear Mapping

Bilinear mapping is called pairing because they associate pairs of elements from two groups to yield an element of a third group, e.g. it can be used to map a pair of points on an elliptic curve. It is also used extensively in this paper. Hence, a bilinear mapping is defined as follows:

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two multiplicative cyclic groups of prime order p. The generator of $\mathcal{G}_1$ is g. A bilinear mapping $\widehat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ satisfies the following properties:

- Bilinearity: For any $u, v \in \mathcal{G}_1$ and $a, b \in \mathbb{Z}_p$, it has $\widehat{e}(u^a, v^b) = \widehat{e}(u, v)^{ab}$.
- Non-degeneracy: $\widehat{e}(g, g) \neq 1$.
- Computability: For all $u, v \in \mathcal{G}_1$, there is an efficient computation $\widehat{e}(u, v)$.

C. Access Tree

An access tree, denoted by $\mathcal{T}$, is used as the graphic representation of the access policy. N_1 denotes the root node of the tree, $N_j (j = 2, 3, \cdots)$ denotes the other nodes of the tree. If N_j is a non-leaf node, it specifies a threshold gate, such as "AND", "OR", "n of m $(n < m)$". The threshold gate is described by node's children and a threshold value. If num_j is the number of children of a node N_j and K_j is the threshold value, $0 < K_j \leqslant num_j$. When $K_j = 1$, the threshold gate is an OR gate and when $K_j = num_j$, it is an AND gate. If N_j is a leaf node, it is associated with a medical worker attribute, and $K_j = 1$. Let $parent(N_i)$ be the parent node of N_i. Let $index(N_i)$ be the index for node N_i in its father's children, i.e., $1 \leqslant index(N_i) \leqslant num_i$.

Each non-leaf node is associated with a Boolean function derived from the access policy. The Boolean function of non-leaf node N_j is represented with K_j/num_j. This means that the Boolean value of N_j is calculated to be TRUE if there are at least K_j children whose Boolean values are calculated to be TRUE. If a medical worker's attribute set $\mathbb{A}$ satisfies $\mathcal{T}$, we denote it as $\mathcal{T}(\mathbb{A})$ =TRUE. $\mathcal{T}(\mathbb{A})$ is defined as follows. For any leaf node N_j, its Boolean value is TRUE if and only if N_j is associated with an attribute $a_i \in \mathbb{A}$, For any non-leaf node, its Boolean value is the value of its Boolean function. If and only if the tree's root node's Boolean value is TRUE, then $\mathcal{T}(\mathbb{A})$ =TRUE.

Take access trees in Fig. 1 for example. There are four access trees corresponding to four access policies. $\mathcal{T}_1$: {"Physician" OR "Nurse"}, $\mathcal{T}_2$: {"Dermatology" AND ("Physician" OR "Nurse")}, $\mathcal{T}_1'$: {"Physician" OR "Nurse"} and $\mathcal{T}_2'$: {"Neurology" AND ("Physician" OR "Nurse")}. Considering a medical worker whose attributes set $\mathbb{A}$ is {Neurology, Physician}. We could calculate that $\mathcal{T}_1(\mathbb{A}) = \mathcal{T}_1'(\mathbb{A}) =$ TRUE, $\mathcal{T}_2(\mathbb{A}) =$ FALSE and $\mathcal{T}_2'(\mathbb{A}) =$ TRUE. Table II lists some results of $\mathcal{T}(\mathbb{A})$ with various attributes $\mathbb{A}$.

TABLE II
EVALUATION OF THE ACCESS TREES IN FIG. 1

$\mathbb{A}$	$\mathcal{T}_1(\mathbb{A})$ / $\mathcal{T}_1'(\mathbb{A})$	$\mathcal{T}_2(\mathbb{A})$	$\mathcal{T}_2'(\mathbb{A})$
Dermatology, Physician	TRUE	TRUE	FALSE
Dermatology, Nurse	TRUE	TRUE	FALSE
Neurology, Physician	TRUE	FALSE	TRUE
Neurology, Nurse	TRUE	FALSE	TRUE
Pneumology, Physician	TRUE	FALSE	FALSE
Pharmacist	FALSE	FALSE	FALSE

IV. SYSTEM AND SECURITY MODELS

In this section, we describe the system model, security model and security assumptions of the system. We also discuss security requirements of the system that we intend to enforce through the proposed scheme.

A. System Model

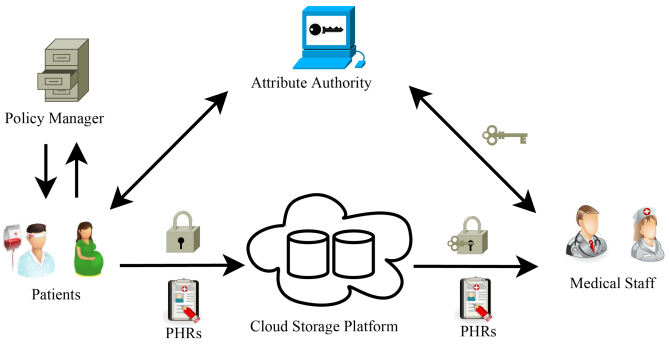


Fig. 2. The system model of secure sharing of PHRs in cloud computing using the proposed scheme

Fig. 2 shows the cloud computing system for PHRs access control, where there is a policy manager, an attribute authority (AA) , a cloud storage platform, multiple patients, as well as a large number of medical staff. The cloud storage platform has an extremely large storage capacity to offer data storage service. The patients encrypt their PHRs and upload them to the cloud storage platform. The medical staff download encrypted PHRs from the cloud and decrypt them. All patients and medical staff are administrated by an AA. AA is required to verify and authenticate the attributes of each medical worker. The policy manager assists the patients to encrypt their PHRs.

The patients' PHRs data may consist of the data collected from different departments of multiple hospitals. Each of the patients classifies and encrypts its PHR independently to protect data integrity. Given the large number of medical staff, the patient defines its own access policy that specifies the attributes of the medical workers who can access different parts of the PHR (or the same data at different classes of details). For illustration convenience, we consider two patients, and L classes of PHR data per patient.

The medical staff of the proposed system can review multiple classes of PHRs data from the two patients from the cloud storage platform. Consider a large number of medical staff, a mixed approach of ABE and symmetric key is taken. Specifically, different classes of PHRs data are encrypted by symmetric keys, while the symmetric keys are encrypted by ABE appropriate to the attribute requirements of medical staff. In other words, a medical worker can decrypt the PHRs data from different patients with its symmetric keys. So the patients encrypt the PHRs data with the same symmetric keys, although each patient derives its symmetric keys independently. To accomplish this, patients send their access policies to the policy manager and policy manager generates some shared information then returns to patients. The patients complete the encryptions of the PHRs on the basis of this shared information as parameters. We use shared symmetric keys in order to improve the efficiency of sharing PHRs. This is reflected in two aspects: In the encryption process, the access policies of the two patients have a common sub-policy. The shared symmetric keys can be only encrypted once under the common sub-policy. This reduces the total encryption time of the patients. In the decryption process, a medical worker whose attributes satisfy the common sub-policy can access PHRs from the two patients. Using shared symmetric keys, the worker only decrypts once in ABE layer. This reduces the decryption time of the individual.

B. Security Assumptions and Requirements

1) Security Assumptions of System: In the system model described in Fig. 2, we consider that AA is a trusted authority which is responsible for creating secret keys for medical staff. The cloud storage platform is honest but curious. On one hand, it performs the operations faithfully and returns correct results. On the other hand, it can attempt to intercept sensitive information. A medical worker may be compromised by an adversary, therefore, it would try to access data files either within or beyond the range of its access privilege level. Moreover, the cloud storage platform might collude with some of medical workers and medical workers may collude with each other to attain more sensitive information. We assume that the communication channels between the patients, the cloud storage platform and the medical staff are insecure. This means adversary could eavesdrop the data in these channels.

2) Security Requirements: The main security requirements that we intend to achieve through the proposed scheme are outlined below.

- *Data Confidentiality*: Preserving the confidentiality of PHR data is a key requirement for a PHR system. Unauthorized parties (either medical staff or an adversary) who do not have enough attributes satisfying the access policy should be prevented from accessing the PHR files.
- *Collusion-resistance*: Multiple medical workers will try to collaborate with each other in order to access the PHR files beyond their access privilege level. However, they will not be able to access the PHR data even though their combined attributes satisfying that access level.
- *Fine-grained and flexible access control*: The PHR system should support fine-grained and flexible access control so as to be practical and achieve better user satisfaction.

C. Security model

We describe the security model of our scheme by the following game between a challenger and an adversary. The security model allows the adversary to query for any secret keys that cannot be used to decrypt the challenge ciphertext. The security game is defined as follows:

Setup: The challenger runs the **Setup** algorithm to generate the system parameters and sends the public key to the adversary.

Phase 1: The adversary makes repeated secret key queries by submitting attributes sets $\mathbb{A}_{q1}, \mathbb{A}_{q2}, \cdots$ to the challenger. The challenger responds by running the **KeyGen** algorithm to generate the corresponding secret keys $SK_{q1}, SK_{q2}, \cdots$.

Challenge: The adversary submits two messages of the same length $\{k_{L,0}, \cdots, k_{1,0}\}$ and $\{k_{L,1}, \cdots, k_{1,1}\}$. In addition, the adversary gives a challenge access policy $\mathcal{T}_L^*$ such that none of the sets $\mathbb{A}_{q1}, \mathbb{A}_{q2}, \cdots$ from Phase 1 satisfy this access policy. The adversary also sends another access policy $\mathcal{T}_L$ which has a common access sub-policy with $\mathcal{T}_L^*$. The challenger then flips a random coin b, and encrypts $\{k_{L,b}, \cdots, k_{1,b}\}$ under the access policy $\mathcal{T}_L^*$. The ciphertext CT^* is given to the adversary.

Phase 2: The adversary may query more secret keys with the restriction that none of the attributes sets satisfy $\mathcal{T}_L^*$.

Guess: The adversary outputs a guess b' of b.

The advantage of the adversary $\mathcal{A}$ in this game is defined as $Pr[b' = b] - (1/2)$.

Definition 1: Our proposed scheme is secure if all polynomial time adversaries have at most a negligible advantage in the above game.

V. The Proposed Access Control Scheme

In this section, we articulate the proposed unified PHRs data access control framework in cloud computing environments, where the PHRs can be provided by multiple patients.

A. Overview of the Unified Access Control Framework

The access tree has the potential to support a multi-privilege access control by merging multiple sub-trees whose roots are in a trunk, where each sub-tree is a sub-policy of access control. Take Fig. 1 for example, "AND"–"OR" is the trunk of the access trees. The conditions on which two access trees can be merged are that: i) The access trees have a common sub-policy, and that ii) For each access tree, all roots of its sub-trees are in a trunk.

Fig. 3 illustrates the merging process of two access trees which come from the access policies of different patients. The access policies meet the two conditions, so they can be merged into one access tree. The merged access tree yields the structure of a multi-root tree. Every leaf node in the multi-root tree corresponds to an inseparable attribute of the medical staff, such as "Physician", "Nurse", "Neurology", "No.1 Hospital", and "No.2 Hospital". Apart from the leaf

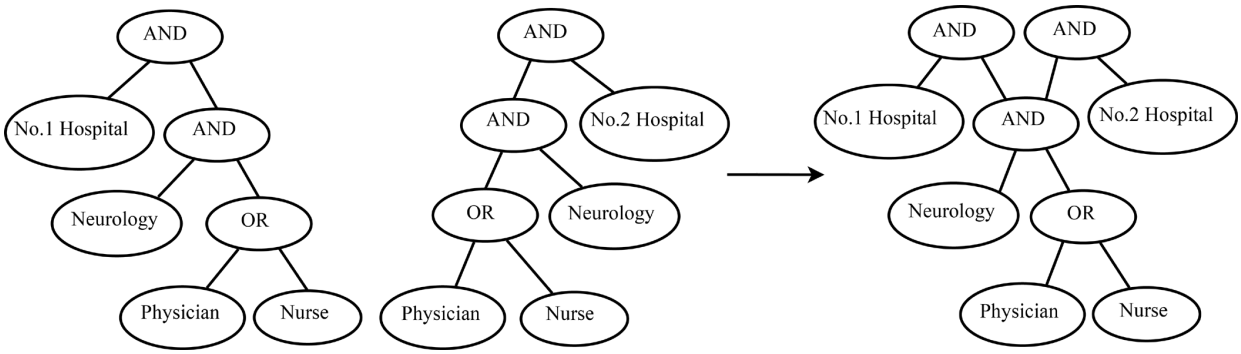


Fig. 3. Two access trees are merged into one access tree

nodes, each of the remaining nodes stands for a threshold gate. These nodes provide fine-grained resolutions corresponding to different parts or detail classes of PHRs data.

Every root in the multi-root tree and the nodes down along the branches correspond to the access policy of one of the patients. Particularly, the shared branches down from multiple roots are the common sub-policy of the patients' access policies. In this sense, each of the nodes along the shared branches is associated with different PHRs data from the two patients. The PHRs data are accessible by the medical workers with the attributes satisfying the requirements of either patient. The details of the access trees merging process will be described in Section V-B.

Following the access tree structure in Fig. 3, the proposed PHRs access control framework is composed of two overlaid layers, as shown in Fig. 4. A fine-grained access control based on ABE is on the top while symmetric keys, denoted by k_i, provide secrecy of the PHRs at the bottom. k_i is the plaintext of the ABE layer. In ABE scheme [13], plaintext is encrypted at the root of the access tree. So all symmetric keys are encrypted at the corresponding access sub-trees' roots, i.e., the all the nodes in the merged trunk.

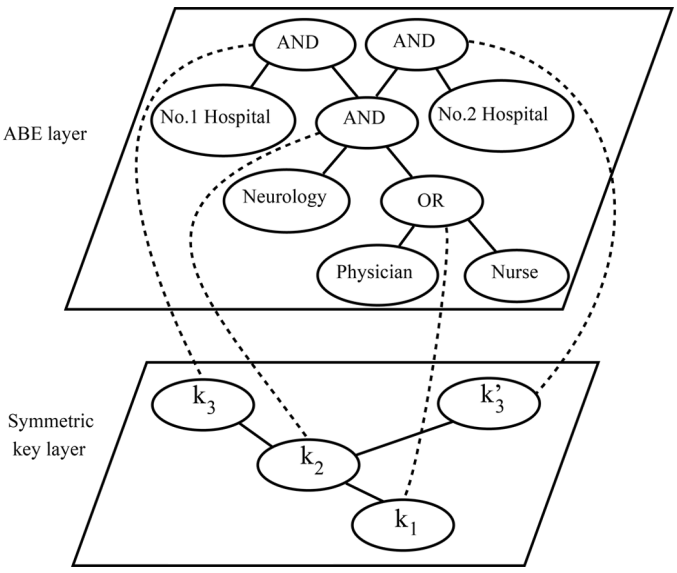


Fig. 4. The proposed two layers access control framework

B. The Design of ABE

In ABE layer, the scheme supports a multi-privilege access control on PHRs data. It also merges the access trees into one, so that the encryption of different patients' PHRs data under the same access policy can be combined. These result in the cost of encryption and decryption are reduced.

Five algorithms are carried out. They are **Setup**, **KeyGen**, **SharedInfoGen**, **Encrypt**, and **Decrypt**. **Setup** is to initialize and generate the public key (PK) and the master secret key (MSK) of the system. The input of this algorithm is a security parameter λ.

The AA chooses a group $\mathcal{G}_1$ of prime order p with generator g, and two random exponents $\alpha, \beta \in \mathbb{Z}_p$, $\mathbb{Z}_p$ is the set of integers $[0, p-1]$. The public key of the system can be generated, as given by

$$PK = \left\{\mathcal{G}_1, g, g^{\beta}, \widehat{e}(g,g)^{\alpha}\right\}. \tag{1}$$

The master key of the system can be given by concatenating β and g^{α}, as given by

$$MSK = \left\{\beta, g^{\alpha}\right\}. \tag{2}$$

The public key is published to all the medical staff. Meanwhile, a secret key SK_j is assigned to a medical worker in the j-th class, denoted by $\mathcal{S}_j$, $j \in [1, L]$. AA generates SK_j, using the algorithm **KeyGen**, based on MSK and the attribute set of $\mathcal{S}_j$, denoted by $\mathbb{A}_j$. Specifically, random values $r, r_i \in \mathbb{Z}_p, \forall a_i \in \mathbb{A}_j$ are chosen, and the secret key of the j-th class is produced, as given by

$$SK_j = \left\{D = g^{(\alpha+r)/\beta}, \{D_i = g^r \cdot \mathcal{H}_1(a_i)^{r_i}, D_i' = g^{r_i}\}_{\forall a_i \in \mathbb{A}_j}\right\}, \tag{3}$$

where a_i denotes the i-th attribute of a medical worker.

Our derivation of the ABE ciphertexts starts from the aggregate node of the shared branch, denoted by $N_{\epsilon}(1 \leqslant \epsilon < L)$, as opposed to the root (as typically done in ABE designs). ϵ is the highest class number that medical staff in this class can access the corresponding PHRs data of the two patients. Patient 1 wants to offer its PHR $\{m_L, \cdots, m_{\epsilon}, \cdots, m_1\}$ and patient 2 needs to offer its PHR $\{m'_L, \cdots, m'_{\epsilon}, \cdots, m'_1\}$ with the medical staff. Two patients define their own access policy that describes L classes of medical workers' attributes separately. Then the patients send the access policies to the policy manager. The policy manager runs **SharedInfoGen** to

generate the shared information SI, which will be returned to the patients to assist the encryption.

According to [13], the access policy can be translated into a fixed-format expression by utilizing the cpabe toolkit [39]. So if the two patients' access policies have a common access sub-policy, the policy manager can extract the sub-policy and generate the common access tree as part of the shared information. The policy manager creates the access tree by mapping the attributes of medical workers to the ϵ privilege levels $\{p_1, \cdots, p_\epsilon\}$. Then it selects an attribute set according to the common sub-policy, and builds an access tree $\mathcal{T}_\epsilon$ whose leaf nodes are associated with this attribute set.

For each node N_j in tree $\mathcal{T}_\epsilon$, the policy manager selects a polynomial $f_j(x)$, and sets the polynomial degree $d_j = K_j - 1$, as given by

$$f_j(x) = \sum_{i=0}^{d_j} b_i x^i, \tag{4}$$

where b_i is the coefficient of the polynomial. The polynomials of the nodes are the part of ciphertext in ABE layer.

Furthermore, for the root of the common access tree N_ϵ, the policy manager selects a random value $s = f_\epsilon(0) \in \mathbb{Z}_p$ and chooses other d_ϵ points of the polynomial $f_\epsilon(x)$ randomly to define $f_\epsilon(x)$.

For any other node N_i, the policy manager chooses a polynomial $f_i(x)$, as such that

$$f_i(0) = f_{parent(N_i)}(index(N_i)) \tag{5}$$

Then the policy manager chooses the other d_i points randomly to define $f_i(x)$. So for any $N_j \in \mathcal{T}_\epsilon$, it can obtain the polynomials $f_{N_j}(\cdot)$. Specifically, the access tree $\mathcal{T}_\epsilon$ has a trunk. Each node in the trunk is the root of a sub-tree of $\mathcal{T}_\epsilon$. The sub-tree's leaf nodes are associated with the corresponding class medical workers' attributes.

The policy manager produces the symmetric keys $\{k_\epsilon, \cdots, k_1\}$ which are used to encrypt and decrypt the patients' PHRs. The details of symmetric keys derivation will be described in Section V-C.

Then policy manager creates the shared information SI:

$$\begin{aligned} SI = \Big\{ & \mathcal{T}_\epsilon, \epsilon, k_\epsilon, \\ & \widetilde{C_i} = \{\mathcal{F}(k_i) \cdot \widehat{e}(g,g)^{\alpha f_i(0)}\}_{i=1}^{\epsilon}, \\ & C_i = \{g^{\beta f_i(0)}\}_{i=1}^{\epsilon}, \\ & \{E_j = g^{f_j(0)}, E_j{}' = \mathcal{H}_1(a_i)^{f_j(0)}\}_{j \in \mathbb{L}(\mathcal{T}_\epsilon)} \Big\}. \end{aligned} \tag{6}$$

where $\mathbb{L}(\mathcal{T}_\epsilon)$ is the set of leaf nodes in the $\mathcal{T}_\epsilon$. Attribute $a_i \in \mathbb{A}(\mathcal{T}_\epsilon)$ is associated with a leaf node $N_j \in \mathbb{L}(\mathcal{T}_\epsilon)$. $\mathbb{A}(\mathcal{T}_\epsilon)$ is the attribute set of $\mathcal{T}_\epsilon$. $\mathcal{F}(\cdot)$ is a mapping that maps an element in $\mathbb{Z}_p$ to an element in $\mathcal{G}_2$ and its inverse mapping is denoted by $\mathcal{F}^{-1}(\cdot)$. The mappings are implemented based on the Pairing-Based Cryptography library [40].

The policy manager sends the SI to the two patients. The patients run **Encrypt** to encrypt their symmetric keys separately. Take patient 1 for example. It builds up the access tree $\mathcal{T}_L$ from bottom to top on the basis of $\mathcal{T}_\epsilon$. $\mathcal{T}_\epsilon$ is a sub-tree of $\mathcal{T}_L$, and the nodes in the trunk of $\mathcal{T}_L$ are mapped to L classes.

For any trunk node N_j above N_ϵ in $\mathcal{T}_L$, the polynomial $f_j(x)$ can be defined as follows.

$$f_j(index(trunkchild(N_j))) = f_{trunkchild(N_j)}(0) \tag{7}$$

where $trunkchild(N_j)$ denotes the child node of N_j along the trunk and $index(N_j)$ is the index value of N_j to its father node. For convenient calculation, let $index(trunkchild(N_j)) = 1$ for all trunk nodes. This can be easily done when patients create their access policies. Then patient 1 randomly chooses d_j coefficients of (4) except the constant coefficient $f_j(0)$, and calculates $f_j(0)$ with (7). For example, suppose N_δ is the parent node of N_ϵ, patient 1 wants to define the polynomial

$$f_\delta(x) = \sum_{i=0}^{d_\delta} b_i x^i \tag{8}$$

So patient 1 chooses d_δ coefficients $\{b_1, b_2, \cdots, b_{d_\delta}\}$ randomly, and $index(trunkchild(N_\delta)) = 1$. According to (7), patient 1 has the knowledge on

$$f_\delta(1) = f_\epsilon(0), \tag{9}$$

then computes the constant coefficient b_0 with (4), as given by

$$b_0 = f_\epsilon(0) - \sum_{i=1}^{d_\delta} b_i \tag{10}$$

In the same way, all the polynomials of the trunk nodes can be obtained. For any other node N_i in $\mathcal{T}_L$, the way of defining the polynomial $f_i(x)$ is the same as the way that the policy manager builds up $\mathcal{T}_\epsilon$.

Through (5) and (7), patient 1 gets all the polynomials of nodes in $\mathcal{T}_L$, and the access tree is built. Similarly, patient 2 builds up its access tree $\mathcal{T}_L'$ on the basis of $\mathcal{T}_\epsilon$. Fig. 5 shows an example of the access trees built.

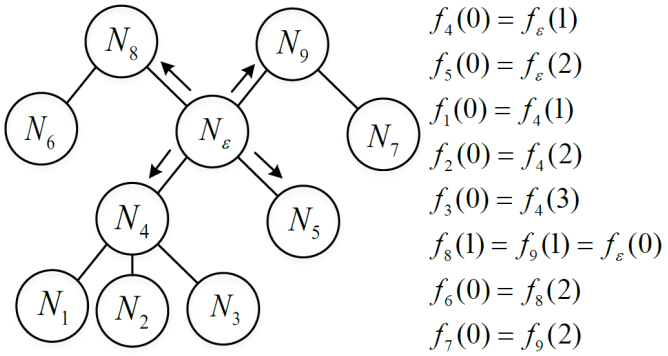


Fig. 5. An example of access trees built. It starts from N_ϵ which is the root of the largest common access tree.

Patient 1 creates the symmetric keys $\{k_L, \cdots, k_{\epsilon+1}\}$ and a value v. $\{k_L, \cdots, k_{\epsilon+1}\}$ and v are used to encrypt and decrypt data of patient 1. Patient 2 creates $\{k_L', \cdots, k_{\epsilon+1}'\}$ and v' as well. We will describe this in detail in Section V-C.

Patient 1 generates the ciphertext CT of symmetric keys as follows:

$$\begin{aligned} CT =&(\mathcal{T}_L, \widetilde{C_i} = \{\mathcal{F}(k_i)\cdot \widehat{e}(g,g)^{\alpha f_i(0)}\}_{i=1}^{\epsilon},\\ &\overline{C_i} = \{\mathcal{F}(k_i \,||\, v)\cdot \widehat{e}(g,g)^{\alpha f_i(0)}\}_{i=\epsilon+1}^{L},\\ &C_i = \{g^{\beta f_i(0)}\}_{i=1}^{L},\\ &\{E_j = g^{f_j(0)}, E_j{}' = \mathcal{H}_1(a_i)^{f_j(0)}\}_{j\in\mathbb{L}(\mathcal{T}_L)}). \end{aligned} \tag{11}$$

where $a_i \in \mathbb{A}(\mathcal{T}_L)$ is associated with a leaf node $N_j \in \mathbb{L}(\mathcal{T}_L)$.

Consider that $\mathcal{S}_J$ is a medical worker of patient 1 and $\mathcal{S}_J$ is in the J-th class. Its attribute set is $\mathbb{A}_J$ and its personal secret key SK_J. $\mathcal{S}_J$ wants to recover $\{k_J, \cdots, k_1\}$ from the ciphertext CT. The decryption procedure **Decypt** is a recursive algorithm as follows.

In the access tree $\mathcal{T}_L$, if N_j is a leaf node and is associated with attribute $a_i \in \mathbb{A}_J \cap \mathbb{A}(\mathcal{T}_L)$, set the Boolean value of N_j to TRUE, and compute

$$\begin{aligned} F_j =&DecNode_1(CT, SK_J, N_j, a_i)\\ =&\frac{\widehat{e}(D_i, E_j)}{\widehat{e}(D_i{}', E_j{}')}\\ =&\frac{\widehat{e}(g^r\cdot \mathcal{H}_1(a_i)^{r_i}, g^{f_j(0)})}{\widehat{e}(g^{r_i}, \mathcal{H}_1(a_i)^{f_j(0)})}\\ =&\frac{\widehat{e}(g^r, g^{f_j(0)})\cdot \widehat{e}(\mathcal{H}_1(a_i)^{r_i}, g^{f_j(0)})}{\widehat{e}(g, \mathcal{H}_1(a_i))^{r_i f_j(0)}}\\ =&\widehat{e}(g,g)^{r f_j(0)} \end{aligned} \tag{12}$$

which is based on the property of bilinear maps as described briefly in Section III-B. If $a_i \notin \mathbb{A}_J \cap \mathbb{A}(\mathcal{T}_L)$, then $F_j = \perp$, where $\perp$ is a termination signal. That means Boolean value of N_j is not TRUE.

If N_j is a non-leaf node, for all nodes N_z that are children of N_j, let S_j be an arbitrary K_j-size set of N_z, such that $F_z \neq \perp$, if no such set S_j exists, then $F_j = \perp$. Otherwise let the Boolean value of N_j be TRUE, and calculate

$$\begin{aligned} F_j =&DecNode_2(CT, SK_J, N_j)\\ =&\prod_{z\in S_j} F_z^{\Delta_{z,S_j}(0)}\\ =&\prod_{z\in S_j} \widehat{e}(g,g)^{r f_z(0)\Delta_{z,S_j}(0)}\\ =&\prod_{z\in S_j} \widehat{e}(g,g)^{r f_j(z)\Delta_{z,S_j}(0)}\\ =&\widehat{e}(g,g)^{r\Sigma_{z\in S_j} f_j(z)\Delta_{z,S_j}(0)}\\ =&\widehat{e}(g,g)^{r f_j(0)} \end{aligned} \tag{13}$$

where $\Delta_{z,S_j}(x) = \Pi_{i\in S_j, i\neq j}\frac{x-i}{j-i}$ is the Lagrange coefficient polynomial.

If the medical worker $\mathcal{S}_J$ is not in a higher class than ϵ, i.e., $1 \leqslant J \leqslant \epsilon$. It calculates

$$\begin{aligned} \mathcal{F}^{-1}(\frac{\widetilde{C_J}F_J}{\widehat{e}(C_J, D)}) =&\mathcal{F}^{-1}(\frac{\mathcal{F}(k_J)\widehat{e}(g,g)^{\alpha f_J(0)}\widehat{e}(g,g)^{r f_J(0)}}{\widehat{e}(g^{\beta f_J(0)}, g^{(\alpha+r)/\beta})})\\ =&\mathcal{F}^{-1}(\frac{\mathcal{F}(k_J)\widehat{e}(g,g)^{(\alpha+r) f_J(0)}}{\widehat{e}(g,g)^{(\alpha+r) f_J(0)}})\\ =&\mathcal{F}^{-1}(\mathcal{F}(k_J))\\ =&k_J \end{aligned} \tag{14}$$

Then $\mathcal{S}_J$ can recover $\{k_{J-1}, \cdots, k_1\}$ with k_J, this will be described in Scetion V-C.

If $\mathcal{S}_J$ is in a class higher than ϵ, i.e., $\epsilon < J \leqslant L$. It calculates

$$\begin{aligned} \mathcal{F}^{-1}(\frac{\overline{C_J}F_J}{\widehat{e}(C_J, D)}) =&\mathcal{F}^{-1}(\frac{\mathcal{F}(k_J \,||\, v)\widehat{e}(g,g)^{\alpha f_J(0)}\widehat{e}(g,g)^{r f_J(0)}}{\widehat{e}(g^{\beta f_J(0)}, g^{(\alpha+r)/\beta})})\\ =&\mathcal{F}^{-1}(\frac{\mathcal{F}(k_J \,||\, v)\widehat{e}(g,g)^{(\alpha+r) f_J(0)}}{\widehat{e}(g,g)^{(\alpha+r) f_J(0)}})\\ =&\mathcal{F}^{-1}(\mathcal{F}(k_J \,||\, v))\\ =&k_J \,||\, v \end{aligned} \tag{15}$$

$\mathcal{S}_J$ can derive $\{k_{J-1}, \cdots, k_1\}$ with k_J and v. This will also be described in Scetion V-C.

For patient 2 and its medical workers $\mathcal{S}'_J$, the encryption and decryption procedures in the ABE layer are similar with patient 1 and $\mathcal{S}_J$. It is worth noting that if $1 \leqslant J \leqslant \epsilon$, medical staff $\mathcal{S}_J$ and $\mathcal{S}'_J$ have the same attributes, so they are considered the same in both of the PHRs access control. If $\epsilon < J \leqslant L$, the largest access sub-tree of $\mathcal{T}'_L$ that the attributes of medical staff $\mathcal{S}_J$ can satisfy is $\mathcal{T}_\epsilon$, so $\mathcal{S}_J$ is same as $\mathcal{S}'_\epsilon$ in patient 2's PHR access control. Also, medical staff $\mathcal{S}'_J$ is same as $\mathcal{S}_\epsilon$ in patient 1's PHR access control in the same way.

C. Symmetric Key Derivation and Data Encryption

$\{k_L, \cdots, k_{\epsilon+1}, k_\epsilon, \cdots, k_1\}$ denotes the symmetric keys which are used to encrypt the L classes of patient 1's PHR data and $\{k'_L, \cdots, k'_{\epsilon+1}, k_\epsilon, \cdots, k_1\}$ denotes the patient 2's symmetric keys. $\{k_\epsilon, \cdots, k_1\}$ is generated by policy manager. Patient 1 and patient 2 create the $\{k_L, \cdots, k_{\epsilon+1}\}$ and $\{k'_L, \cdots, k'_{\epsilon+1}\}$, respectively. The symmetric encryption and decryption algorithms are $\mathcal{E}(\cdot)$ and $\mathcal{D}(\cdot)$.

When the policy manager produces the shared information SI, it selects the symmetric key k_ϵ from $\mathbb{Z}_p$ randomly, and generates a hash chain for symmetric keys as

$$k_j = \mathcal{H}_2(k_{j+1} \,||\, j), j = \epsilon-1, \epsilon-2, \cdots, 1 \tag{16}$$

where $\mathcal{H}_2(\cdot)$ is a one-way function, $a \,||\, b$ is the concatenation of two strings a and b, and k_ϵ is sent to the patients as part of SI.

When the patients encrypt their own PHRs on the basis of SI, they obtain ϵ and k_ϵ from SI. Then they can derive $\{k_\epsilon, \cdots, k_1\}$ through (16). Patient 1 randomly chooses the symmetric key k_L from $\mathbb{Z}_p$, and generates another hash chain as

$$k_j = \mathcal{H}_2(k_{j+1} \,||\, j), j = L-1, L-2, \cdots, \epsilon+2, \epsilon+1 \tag{17}$$

Patient 1 calculates the value $v = \mathcal{E}(k_\epsilon, k_{\epsilon+1})$, and uses v, SI and $\{k_L, \cdots, k_{\epsilon+1}, k_\epsilon, \cdots, k_1\}$ to generate CT, as described in Section V-B.

Patient 2 obtains $\{k'_L, \cdots, k'_{\epsilon+1}, k_\epsilon, \cdots, k_1\}$, v' and generates CT' in the same manner as patient 1.

Patient 1 encypts its PHR $\{m_L, \cdots, m_\epsilon, \cdots, m_1\}$ with its symmetric keys by calling $\mathcal{E}(\cdot)$ and generates the ciphertext of PHR EM:

$$EM = (\{EM_i = \mathcal{E}(m_i, k_i)\}_{i=1}^{\epsilon}, \{EM_i = \mathcal{E}(m_i, k_i)\}_{i=\epsilon+1}^{L}) \tag{18}$$

where EM_i is the ciphertext of the i-th class PHR data.

Patient 2 encypts its PHR $\{m'_L, \cdots, m'_\epsilon, \cdots, m'_1\}$ and creats the ciphertext EM':

$$EM' = (\{EM'_i = \mathcal{E}(m'_i, k_i)\}_{i=1}^{\epsilon}, \{EM'_i = \mathcal{E}(m'_i, k'_i)\}_{i=\epsilon+1}^{L}) \tag{19}$$

The two patients upload their ciphertexts of symmetric keys and PHRs $\{CT, EM\}$ and $\{CT', EM'\}$ to the cloud storage platform.

Consider the aforementioned medical workers $\mathcal{S}_J$ discussed in Section V-B. M_J is the PHRs data set that $\mathcal{S}_J$ is able to access. If $\mathcal{S}_J$ is not in a class higher than ϵ, i.e., $1 \leqslant J \leqslant \epsilon$. It can access J classes PHRs data from the both patients, that is $M_J = \{m_J, \cdots, m_1, m'_J, \cdots, m'_1\}$. Medical workers $\mathcal{S}_J$ can obtain k_J from CT, as described in Section V-B. Then $\mathcal{S}_J$ recovers the symmetric keys $\{k_{J-1}, \cdots, k_1\}$ as

$$k_j = \mathcal{H}_2(k_{j+1} \,||\, j), j = J-1, \cdots, 1 \tag{20}$$

$\mathcal{S}_J$ decrypts the EM and EM' by its symmetric keys to get M_J as

$$\begin{aligned} M_J =& \left\{\{\mathcal{D}(EM_i, k_i)\}_{i=1}^{J}, \{\mathcal{D}(EM'_i, k_i)\}_{i=1}^{J}\right\} \\ =& \left\{\{m_i\}_{i=1}^{J}, \{m'_i\}_{i=1}^{J}\right\} \end{aligned} \tag{21}$$

If $\mathcal{S}_J$ is in a class higher than ϵ, i.e., $\epsilon < J \leqslant L$. It can access J classes PHR data from patient 1 and ϵ classes PHR data from patient 2, which is $M_J = \{m_J, \cdots, m_1, m'_\epsilon, \cdots, m'_1\}$. $\mathcal{S}_J$ can obtain k_J and v from CT. $\mathcal{S}_J$ first recovers $\{k_{J-1}, \cdots, k_{\epsilon+1}\}$ as

$$k_j = \mathcal{H}_2(k_{j+1} \,||\, j), j = J-1, \cdots, \epsilon+1 \tag{22}$$

then it calculates $k_\epsilon = \mathcal{D}(v, k_{\epsilon+1})$ to get k_ϵ, and recovers $\{k_{\epsilon-1}, \cdots, k_1\}$ by (16). $\mathcal{S}_J$ decrypts the EM and EM' by its symmetric keys to gain M_J as

$$\begin{aligned} M_J =& \left\{\{\mathcal{D}(EM_i, k_i)\}_{i=1}^{J}, \{\mathcal{D}(EM'_i, k_i)\}_{i=1}^{\epsilon}\right\} \\ =& \left\{\{m_i\}_{i=1}^{J}, \{m'_i\}_{i=1}^{\epsilon}\right\} \end{aligned} \tag{23}$$

VI. Security Analysis

A. Security Proof

Extended from CP-ABE, our scheme is expected to have the same property as CP-ABE, which has been proven to be secure under the generic bilinear group model and the random oracle model [13]. We prove the security of our scheme based on the security of CP-ABE.

Theorem 1: Suppose that there is no polynomial time adversary who can break the security of CP-ABE with non-negligible advantage; then there is no polynomial time adversary who can break our scheme with non-negligible advantage.

proof: Suppose an adversary $\mathcal{A}$ can break our scheme with non-negligible advantage $Adv_{\mathcal{A}}$. Using $\mathcal{A}$, we can build an adversary $\mathcal{B}$ that plays a similar game with CP-ABE and breaks CP-ABE with non-negligible advantage $Adv_{\mathcal{B}}$.

Setup: $\mathcal{B}$ takes public key of CP-ABE $PK = \left\{\mathcal{G}_1, g, g^\beta, \widehat{e}(g,g)^\alpha\right\}$, and sends $PK' = PK$ to the adversary $\mathcal{A}$.

Phase 1: $\mathcal{B}$ answers secret key queries from $\mathcal{A}$. Suppose $\mathcal{A}$ makes a query for attributes set $\mathbb{A}_{q1}$. $\mathcal{B}$ submits $\mathbb{A}_{q1}$ to CP-ABE challenger and obtains $SK_{q1} = \left\{D = g^{(\alpha+r)/\beta}, \{D_i = g^r \cdot \mathcal{H}_1(a_i)^{r_i}, D_i{}' = g^{r_i}\}_{\forall a_i \in \mathbb{A}_{q1}}\right\}$, where r and r_i are randomly selected from $\mathbb{Z}_p$. Then $\mathcal{B}$ returns $SK'_{q1} = SK_{q1}$ to $\mathcal{A}$ as the answer of the query. $\mathcal{B}$ will always answer until $\mathcal{A}$ stops querying.

Challenge: $\mathcal{A}$ gives two messages of the same length $\{k_{L,0}, \cdots, k_{1,0}\}$, $\{k_{L,1}, \cdots, k_{1,1}\}$ and a access policy $\mathcal{T}_L^*$ such that none of the attributes sets which $\mathcal{A}$ has queried in Phase 1 satisfy $\mathcal{T}_L^*$. $\mathcal{A}$ also sends another access policy $\mathcal{T}_L$ which has a common access sub-policy with $\mathcal{T}_L^*$. $\mathcal{B}$ takes two snippets of the messages $k_{L,0}$ and $k_{L,1}$. $\mathcal{B}$ extracts access policies $\mathcal{T}_{L-1}^*, \cdots, \mathcal{T}_1^*$ from $\mathcal{T}_L^*$. $\mathcal{B}$ sends $k_{L,0}$, $k_{L,1}$ and $\mathcal{T}_L^*$ to CP-ABE challenger, then receives the challenge ciphertext $CT = (\mathcal{T}_L^*, \widetilde{C_L} = k_{L,b} \cdot \widehat{e}(g,g)^{\alpha f_L(0)}, C_L = g^{\beta f_L(0)}, \{E_j = g^{f_j(0)}, E_j{}' = \mathcal{H}_1(a_i)^{f_j(0)}\}_{j \in \mathbb{L}(\mathcal{T}_L^*)})$. Then $\mathcal{B}$ constructs challenge ciphertext CT^* for $\mathcal{A}$ on the basis of CT as: $CT^* = (\mathcal{T}_L^*, \widetilde{C_L} = k_{L,b} \cdot \widehat{e}(g,g)^{\alpha f_L(0)}, \widetilde{C_i} = \{k_{i,0} \cdot \widehat{e}(g,g)^{\alpha f_i(0)}\}_{i=1}^{L-1}, C_L = g^{\beta f_L(0)}, C_i = \{g^{\beta f_i(0)}\}_{i=1}^{L-1}, \{E_j = g^{f_j(0)}, E_j{}' = \mathcal{H}_1(a_i)^{f_j(0)}\}_{j \in \mathbb{L}(\mathcal{T}_L^*)})$. In CT^*, $\widetilde{C_i}$ and C_i can be obtained from the CP-ABE challenger by submitting the pairs $(k_{i,0}, \mathcal{T}_i^*)$, where $i = L-1, \cdots, 1$. Finally, CT^* is returned to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ may query more secret keys with the restriction that none of the attributes sets satisfy $\mathcal{T}_L^*$. $\mathcal{B}$ responds as in Phase 1.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$, and $\mathcal{B}$ outputs b' in its own game. The advantage of $\mathcal{B}$ over CP-ABE is

$Adv_{\mathcal{B}} = |Pr[b = b'] - 1/2| = Adv_{\mathcal{A}}$

Therefore, $\mathcal{B}$ has a non-negligible advantage to break the CP-ABE scheme, which completes the proof of the theorem. ■

B. Discussion

In our proposed scheme, PHR files and the symmetric keys which are used to encrypted PHR files are stored in cloud in an encrypted format. Therefore, it is infeasible in practice to learn the content of the PHR files for any unauthorized parties including the CSP. For authorized medical workers, a covetous medical worker may try to elevate its lower privilege class to a higher class so that it can access the PHRs data beyond its privilege class. For instance, a medical worker has privilege p_i with symmetric key k_i and it tries to gain k_{i+1}. However, this is computationally prohibitive because of the one-wayness of

the hash function used in the generation of the symmetric key chain. Thus, the proposed scheme is capable of maintaining the confidentiality of the PHR data.

The proposed scheme is resistant against attacks through attributes collusion. We have achieved this by embedding a random value r into the secret key SK of every medical worker respectively. Therefore, even two medical workers have the same attributes, their secret keys are different. Hence, if two medical workers collude, they will not yield the desired value. For example, in Fig. 3, a medical worker who has the attribute "Neurology" can compute $F_{Neurology} = \widehat{e}(g,g)^{r_{Neurology} f_{AND}(1)}$. Another medical worker with the attribute "Nurse" can calculate $F_{Nurse} = \widehat{e}(g,g)^{r_{Nurse} f_{AND}(2)}$. Both medical workers could launch collusion attacks together to obtain F_{AND}. Nevertheless, the two random values $r_{Neurology}$ and r_{Nurse} are chosen securely and separately, the medical workers can by no means obtain F_{AND}. Our scheme can resist collusion attacks.

Since the symmetric key is encrypted under the access policy, only the medical worker who has the required attributes can recover the symmetric key to access the PHR data. So our scheme can achieve fine-grained access control. A patient defines its access policy based on the attributes of medical staff without explicitly listing their names and access policy can be defined with threshold gates operating on attributes. So patients can define complex access policies and the number of access policies supported by our scheme is very large. Thus, our scheme is capable of implementing flexible access control.

VII. Simulation and Evaluation

In this section, we first analyze the efficiency of our proposed scheme theoretically, then conduct the experiments and results show that our scheme is more efficient than other schemes.

A. Efficiency

Let $C(\mathcal{G}_i)$ be the operation in group $\mathcal{G}_i (i = 0, 1)$, that includes exponentiation and multiplication, $C_{\widehat{e}}$ be the operation in bilinear map $\widehat{e}(\cdot)$. To facilitate the analysis, we assume that two patients have a similar structure of access trees, this means the two access trees have the same number of classes and identical classes have the same number of attributes. $\mathbb{A}(\mathcal{T}_j)$ is the attribute set of j class access tree, and for each tree, $\mathbb{A}(\mathcal{T}_1) \subset \mathbb{A}(\mathcal{T}_2) \cdots \subset \mathbb{A}(\mathcal{T}_L)$. $B(*)$ is the bit size of an element in $*$, and $|*|$ is the number of elements in $*$. $\mathbb{S}(\mathcal{T}_j)$ is the least nodes set that satisfy the access tree $\mathcal{T}_j$. Table III summarizes the efficiency comparison among CP-ABE [13], MCP-ABE [17] and our proposed scheme. CP-ABE is a typical scheme for data sharing. MCP-ABE extends CP-ABE to support the multi-privilege access control. In addition, considering the hash computation cost is very small, so it does not have to be included.

As shown in Table III, suppose the values of L and $\mathbb{A}(\mathcal{T}_L)$ are given. When ϵ is fixed, the computational time of encryption decreases linearly with $|\mathbb{A}(\mathcal{T}_\epsilon)|$. The decline rate is equal to $2C(\mathcal{G}_1)$ in our scheme. The encryption time stays the same in MCP-ABE. In CP-ABE, the encryption time increases with the rising $|\mathbb{A}(\mathcal{T}_\epsilon)|$. Similarly, when $|\mathbb{A}(\mathcal{T}_\epsilon)|$ is fixed, the encryption time decreases linearly with ϵ and the decline rate is equal to $C(\mathcal{G}_1) + 2C(\mathcal{G}_2)$ in our scheme. The encryption time in MCP-ABE remains unchanged. Although the encryption time also decreases in CP-ABE, it is still much higher than that in our scheme. As for decryption, medical staff in different classes have different computational time. If $j > \epsilon$, the medical staff $\mathcal{S}_j$ can access $\{m_j, \cdots, m_1, m'_\epsilon, \cdots, m'_1\}$. If $j <= \epsilon$, the medical staff $\mathcal{S}_j$ can access $\{m_j, \cdots, m_1, m'_j, \cdots, m'_1\}$. So we distinguish the decryption time consumption according to whether the medical workers' class is greater than ϵ and the decryption time of our scheme is lower than others in every class. It is worth noting that in Table III, the encryption time is the total time of two patients and the decryption time is the time of an individual medical worker in the j-th class.

B. Simulation

In ABE layer, we have implemented our scheme based on the cpabe toolkit [39] which uses the Pairing-Based Cryptography library [40]. The implementation uses a 160-bit elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over a 512-bit finite field. In symmetric key layer, the implementation is based on the OpenSSL library [41]. The hash function SHA-256 is used to generate the key chain and the symmetric-key algorithm AES is used to encrypt/decrypt the data. Then experiments are conducted on a PC with Intel Core2 Duo@3.00GHz CPU and 2GB, running Ubuntu15.04. We compare our scheme with CP-ABE and MCP-ABE in terms of encyption time and decryption time. All the simulation results are the average of 1000 independent trials. It should be noted that the decryption time here is an average time of the medical staff from all classes, and we assume the medical staff in L classes follow a uniform distribution. In the simulation, we suppose the access policies only contain "AND" gate so that all the ciphertext components can be computed in the decryption algorithm. When $|\mathbb{A}(\mathcal{T}_L)|$ and L are given, in order to consider the worst situation of access policy (i.e. the most complexity), the attributes are constructed in lowest class of access policy as far as possible.

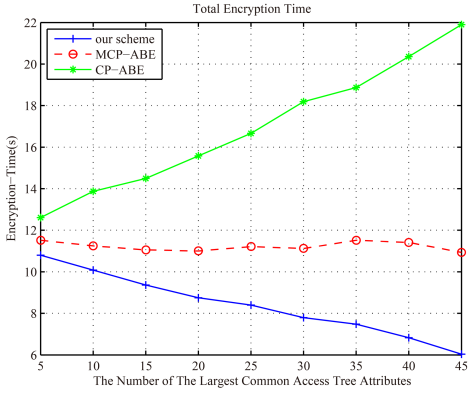


Fig. 6. Total encryption time for varying number of the largest common access tree attributes

Fig. 6 and Fig. 7 show the time consumption of encryption and decryption with given experimental conditions $L = 2$

TABLE III
EFFICIENCY ANALYSIS

Component	CP-ABE	MCP-ABE	Proposed
Public key size	$3B(\mathcal{G}_1) + B(\mathcal{G}_2)$	$3B(\mathcal{G}_1) + B(\mathcal{G}_2)$	$3B(\mathcal{G}_1) + B(\mathcal{G}_2)$
Master key size	$B(\mathcal{G}_1) + B(\mathcal{G}_2)$	$B(\mathcal{G}_1) + B(\mathcal{G}_2)$	$B(\mathcal{G}_1) + B(\mathcal{G}_2)$
Secret key size	$(2\|\mathbb{A}_j\| + 1)B(\mathcal{G}_1)$	$(2\|\mathbb{A}_j\| + 1)B(\mathcal{G}_1)$	$(2\|\mathbb{A}_j\| + 1)B(\mathcal{G}_1)$
Encryption Time	$[4(\|\mathbb{A}(\mathcal{T}_1)\| + \cdots + \|\mathbb{A}(\mathcal{T}_L)\|) + 2L]C(\mathcal{G}_1) + 4LC(\mathcal{G}_2)$	$(4\|\mathbb{A}(\mathcal{T}_L)\| + 2L)C(\mathcal{G}_1) + 4LC(\mathcal{G}_2)$	$[4(\|\mathbb{A}(\mathcal{T}_L)\| - 2\|\mathbb{A}(\mathcal{T}_\epsilon)\|) + (2L - \epsilon)]C(\mathcal{G}_1) + (4L - 2\epsilon)C(\mathcal{G}_2)$
Decryption Time ($j > \epsilon$)	$(4j\|\mathbb{A}_j\| + 2j)C_{\widehat{e}} + [4(\|\mathbb{S}(\mathcal{T}_1)\| + \cdots + \|\mathbb{S}(\mathcal{T}_j)\|) + 4j]C(\mathcal{G}_2)$	$2(\|\mathbb{A}_j\| + \|\mathbb{A}_\epsilon\| + 1)C_{\widehat{e}} + 2[\|\mathbb{S}(\mathcal{T}_j)\| + \|\mathbb{S}(\mathcal{T}_\epsilon)\| + j + \epsilon]C(\mathcal{G}_2)$	$(2\|\mathbb{A}_j\| + 1)C_{\widehat{e}} + 2(\|\mathbb{S}(\mathcal{T}_j)\| + j)C(\mathcal{G}_2)$
Decryption Time ($j <= \epsilon$)	$(4j\|\mathbb{A}_j\| + 2j)C_{\widehat{e}} + [4(\|\mathbb{S}(\mathcal{T}_1)\| + \cdots + \|\mathbb{S}(\mathcal{T}_j)\|) + 4j]C(\mathcal{G}_2)$	$(4\|\mathbb{A}_j\| + 2)C_{\widehat{e}} + 4(\|\mathbb{S}(\mathcal{T}_j)\| + j)C(\mathcal{G}_2)$	$(2\|\mathbb{A}_j\| + 1)C_{\widehat{e}} + 2(\|\mathbb{S}(\mathcal{T}_j)\| + j)C(\mathcal{G}_2)$

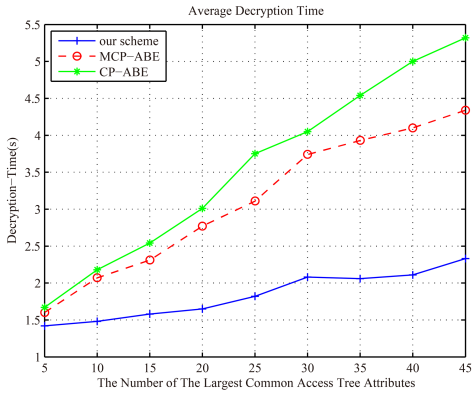


Fig. 7. Average decryption time for varying number of the largest common access tree attributes

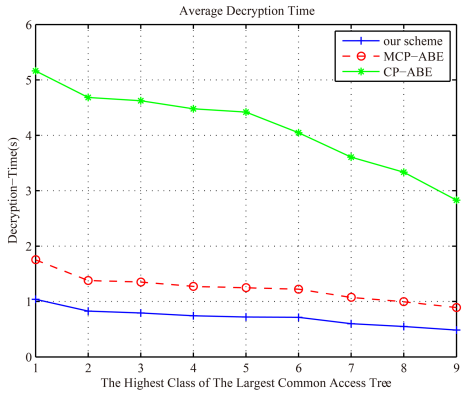


Fig. 9. Average decryption time for varying highest class of the largest common access tree

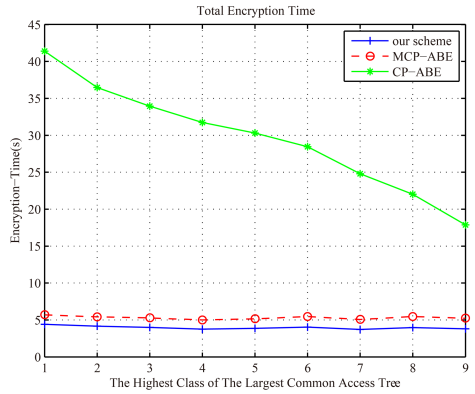


Fig. 8. Total encryption time for varying highest class of the largest common access tree

and $|\mathbb{A}(\mathcal{T}_L)| = 50$. Setting $\epsilon = 1$ and the number of attributes of the common access tree used in the simulation is $|\mathbb{A}(\mathcal{T}_\epsilon)| = \{5, 10, 15, 20, 25, 30, 35, 40, 45\}$. Fig.8 and Fig.9 display the encryption and decryption time with $L = 10$ and $|\mathbb{A}(\mathcal{T}_L)| = 20$. Setting $|\mathbb{A}(\mathcal{T}_\epsilon)| = 10$ and the highest class of the common access tree used in the experiments is $\epsilon = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

As shown in Fig. 6 and Fig. 7, we can see that the encryption time of our scheme is approximately following a linear decrease in number of the common access tree attributes. The result of MCP-ABE almost keeps unchanged and the result of CP-ABE grows. The decryption time of the three schemes all increase with the rising number of attributes in the common access tree. While in our scheme, the decryption time is much shorter than the results of other schemes. As shown in Fig. 8 and Fig. 9, when the number of the common access tree attributes is fixed, the time consumption of encryption is decreasing and following a linear relationship with the highest class of the common access tree in our scheme. The decryption time in our scheme is also shorter than other schemes' due to the fact that the individual medical worker's decryption time is shorter than the medical worker's of other schemes in every class. So our scheme improves the efficiencies of encryption and decryption.

VIII. CONCLUSION

In this paper, a novel scheme of fine-grained and flexible access control has been proposed for sharing patients PHRs data in cloud computing environments. The scheme introduces a policy manager which extracts a common access sub-policy on the basis of multi-patients access policies and then generates shared information. The scheme combines the encryption of PHRs data which are under the common access sub-policy to reduce the time consumption of encryption and decryption. Multi-privilege access control is also supported in our scheme, so that medical staff can access the required level

of data while maximizing patient privacy. Both theoretical and experimental analyses have been carried out to demonstrate the computational complexity and security performance of the scheme. It is worth noting that we assume there are two patients in the experiments. In practice, the more patients whose access policies have a common sub-policy, the more efficient our scheme shares the PHRs data.

The access policies go up to individual persons instead of generic categories in many PHR systems, our scheme can specify individual persons who have right to access the information with well-defined access policies. However, we would optimize the system implementation and conduct more comprehensive evaluations with a real-world healthcare dataset to make it more adapted to the real PHR systems. We would also apply our approach to support PHRs sharing using intelligent terminals such as smart phone and tablet to make it more practical in the future work.

REFERENCES

[1] A. Abbas, and S. U. Khan, "A review on the state-of-the-art privacy-preserving approaches in the e-health clouds," *IEEE J. Biomed. Health Inform.*, vol. 18, no. 4, pp. 1431–1441, Jul. 2014.

[2] D. C. Kaelber, A. K. Jha, D. Johnston, B. Middleton, and D. W. Bates, "A research agenda for personal health records (PHRs)," *J. Am. Med. Inform. Assn.*, vol. 15, no. 6, pp. 729–736, 2008.

[3] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," in *Proc. IEEE Infocom*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[4] E. Abukhousa, N. Mohamed, and J. Aljaroodi, "E-Health Cloud: Opportunities and Challenges," *Future Internet*, vol. 4, no. 3, pp. 621–645, Jul. 2012.

[5] J. Vilaplana, F. Solsona, F. Abella, R. Filgueira, and J. Rius, "The cloud paradigm applied to e-Health," *BMC Med. Inform. Decis.*, vol. 13, no. 1, pp. 35, 2013.

[6] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An Efficient File Hierarchy Attribute-Based Encryption Scheme in Cloud Computing," *IEEE Trans. Inf. Foren. Sec.*, vol. 11, no. 6, pp. 1265–1277, Jun. 2016.

[7] L. Li, X. Chen, H. Jiang, Z. Li, and K. C. Li, "P-CP-ABE: Parallelizing Ciphertext-Policy Attribute-Based Encryption for clouds," in *Proc. 17th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parall. Distr. Comput.*, Shanghai, China, May. 2016, pp. 575–580.

[8] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *Int. J. Inf. Secur.*, vol. 14, no. 6, pp. 487–497, 2015.

[9] C. Guo, R. Zhuang, Y. Jie, Y. Ren, T. Wu, and K. R. Choo, "Fine-grained Database Field Search Using Attribute-Based Encryption for E-Healthcare Clouds," *J. Med. Syst.*, vol. 40, no. 11, pp. 235, 2016.

[10] L. Liu, J. Lai, R. H. Deng, and Y. Li, "Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 4897–4913, 2016.

[11] H. S. G. Pussewalage, and V. Oleshchuk, "A Patient-Centric Attribute Based Access Control Scheme for Secure Sharing of Personal Health Records Using Cloud Computing," in *Proc. 2nd IEEE Int. Conf. Collaboration Int. Comput.*, Pittsburgh, PA, USA, Nov. 2016, pp. 46–53.

[12] A. Sahai, and B. Waters, "Fuzzy identity-based encryption," *Adv. Cryptol. Eurocrypt*, vol. 3494, pp. 457–473, May. 2005.

[13] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Sympo. Secur. Priv.*, Oakland, CA, USA, May. 2007, pp. 321–334.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Alexandria, VA, USA, Oct. 2006, pp. 89–98.

[15] ASTM E2369 Standard Specification for Continuity of Care Record(CCR), ASTM International, 2012.

[16] J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. Peterson, and A. D. Rubin, "Securing electronic medical records using attribute-based encryption on mobile devices," in *Proc. 1st ACM Workshop Secur. Priv. Mob. Dev.*, Chicago, IL, USA, Oct. 2011, pp. 75–86.

[17] Y. Wu, Z. Wei, and R. H. Deng, "Attribute-based access to scalable media in cloud-assisted content sharing networks," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 778–788, 2013.

[18] T. S. Chen, C. H. Liu, T. L. Chen, C. S. Chen, J. G. Bau, and T. C. Lin, "Secure Dynamic Access Control Scheme of PHR in Cloud Computing," *J. Med. Syst.*, vol. 36, no. 6, pp. 4005–4020, 2012.

[19] D. Chen, L. Chen, X. Fan, L. He, S. Pan, and R. Hu, "Securing Patient-Centric Personal Health Records Sharing System in Cloud Computing," *China Commun.*, vol. 11, no. 13, pp. 121–127, 2014.

[20] F. Prasser, F. Kohlmayer, H. Spengler, and K. Kuhn, "A scalable and pragmatic method for the safe sharing of high-quality health data," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 2, pp. 611–622, 2018.

[21] H. Kuehner, and H. Hartenstein, "Decentralized Secure Data Sharing with Attribute-Based Encryption: A Resource Consumption Analysis," in *Proc. 4th ACM Int. Workshop Security Cloud Comput.*, Xi'an, Shaanxi, China, May. 2016, pp. 74–81.

[22] K. Yang, K. Zhang, X. Jia, M. A. Hasan, and X. S. Shen, "Privacy-preserving attribute-keyword based data publish-subscribe service on cloud platforms," *Inf. Sci.*, vol. 387, pp. 116–131, 2017.

[23] L. Duan, D. Liu, Y. Zhang, S. Chen, R. P. Liu, B. Cheng, and J. Chen, "Secure Data-centric Access Control for Smart Grid Services based on Publish/Subscribe Systems," *ACM Trans. Internet. Technol.*, vol. 16, no. 4, pp. 23, 2016.

[24] W. Li, W. Ni, D. Liu, R. P. Liu, P. Wang, and S. Luo, "Fine-grained Access Control for Personal Health Records in Cloud Computing," in *Proc. 85th Vehicular Tech. Conf. (VTC Spring)*, Sydney, NSW, Australia, Jun. 2017, pp. 1–5.

[25] S. Hohenberger, and B. Waters, "Online/offline attribute-based encryption," in *Proc. Int. Workshop Public-Key Cryptography*, Buenos Aires, Argentina, Mar. 2014, pp. 293310.

[26] Y. Rouselakis, and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM SIGSAC conf. Comput. Commun. Security*, Berlin, Germany, Nov. 2013, pp. 463–474.

[27] A. Lewko, and B. Waters, "Unbounded HIBE and attribute-based encryption," in *Proc. EUROCRYPT*, Tallinn, Estonia, May. 2011, pp. 547–567.

[28] J. Horwitz, and B. Lynn, "Toward Hierarchical Identity-Based Encryption," in *Proc. EUROCRYPT*, Amsterdam, The Netherlands, Apr. 2002, pp. 466–481.

[29] P. Ananth, Z. Brakerski, G. Segev, and V. Vaikuntanathan, "From selective to adaptive security in functional encryption," in *Proc. CRYPTO*, Santa Barbara, CA, USA, Aug. 2015, pp. 657–677.

[30] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Trans. Parall. Distr.*, vol. 23, no. 11, pp. 2150–2162, 2012.

[31] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Trans. Parall. Distr.*, vol. 24, no. 1, pp. 131–143, 2013.

[32] M. Barua, X. Liang, R. Lu, and X. Shen, "ESPAC: Enabling Security and Patient-centric Access Control for eHealth in cloud computing," *Int. J. Security Netw.*, vol. 6, no. 2–3, pp. 67–76, 2011.

[33] B. Dan, and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," in *Proc. CRYPTO*, Santa Barbara, CA, USA, Aug. 2001, pp. 213–229.

[34] J. Eom, D. H. Lee, and K. Lee, "Patient-Controlled Attribute-Based Encryption for Secure Electronic Health Records System," *J. Med. Syst.*, vol. 40, no. 12, pp. 253, 2016.

[35] R. Wu, G. J. Ahn, and H. Hu, "Secure sharing of electronic health records in clouds," in *Proc. 8th Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*, Pittsburgh, PA, USA, Oct. 2012, pp. 711–718.

[36] D. W. Chadwick, and K. Fatema, "A privacy preserving authorisation system for the cloud," *J. Comput. Syst. Sci.*, vol. 78, no. 5, pp. 1359-1373, 2012.

[37] H. Kondylakis, G. Flouris, I. Fundulaki, V. Papakonstantinou, and M. Tsiknakis, "Flexible Access to Patient Data through e-Consent," in *Proc. 5th EAI Int. Conf. Wireless Mob. Commun. Healthcare*, London, Great Britain, Oct. 2015, pp. 263-266.

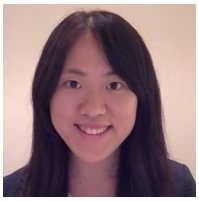
[38] Secure Hash Standard (SHS), FIPS Publication 180-4, National Inst. Standards and Technol., 2015.

[39] The Ciphertext-Policy Attribute-Based Encryption Toolkit, [Online]. Available: http://acsc.cs.utexas.edu/cpabe/

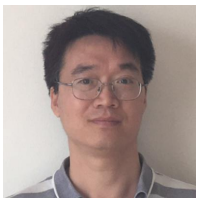
[40] The Pairing-Based Cryptography Library (PBC), [Online]. Available: http://crypto.stanford.edu/pbc/

[41] The OpenSSL Library, [Online]. Available: https://www.openssl.org/

Wei Li received the B.E. and M.E. degrees from Xidian University, Xi'an, China, in 2009 and 2012. He is currently pursuing the Ph.D. degree with the Information Security Center, Beijing University of Posts and Telecommunications, Beijing, China. He visited Data61, CSIRO, Australia, in 2015. His research interests include cryptography, cloud computing, and cyber security.



Bonnie M. Liu received her Bachelor of Medicine and Bachelor of Surgery degrees from Monash University. She currently works as a medical registrar at Royal North Shore Hospital. Her research interests include polypharmacy, healthy ageing and health informatics.



Dongxi Liu was a Researcher with the University of Tokyo from 2004 to 2008. He is a Senior Research Scientist in CSIRO since 2008. His research interests include IoT security, encrypted data processing, and system security. His work aims to develop the research direction of designing public key encryption based on checkable hardness facts instead of more conventional hardness assumption.

Ren Ping Liu (M'09–SM'14) is a Professor and Head of Discipline of Network and Cybersecurity in the School of Electrical and Data Engineering at University of Technology Sydney. In his industry engagements, Professor Liu is the CTO of Ultimo Digital Technologies, developing IoT and Blockchain technologies. He is also the Research Program Leader in Food Agility Cooperative Research Centre, a government-industry-research initiative to empower Australia's agriculture and food industry through digital transformation. Prior to that he was a Principal Scientist and Research Leader at CSIRO, where he led wireless networking research activities. Professor Liu was the winner of Australian Engineering Innovation Award and CSIRO Chairmans medal. He specialises in protocol design and modelling and has delivered networking solutions to a number of government agencies and industry customers. His research interests include 5G, VANET, IoT, cybersecurity, and Blockchain.

Professor Liu was the founding chair of IEEE NSW VTS Chapter and a Senior Member of IEEE. He served as Technical Program Committee chairs, Organising Committee chairs, and delivered keynote speeches in a number of IEEE Conferences. Ren Ping Liu received his B.E.(Hon) and M.E. degrees from Beijing University of Posts and Telecommunications, China, and the Ph.D. degree from the University of Newcastle, Australia.

Peishun Wang was a Research Fellow at Macquarie University and University of Wollongong and a Leading Scientist at Smart Services CRC. He joined IBM as a Senior Security Compliance Specialist in 2011. His research interests include cryptography, information security, network security, operating system security, application security, cloud computing, security GRC (Governance, Risk and Compliance), and privacy. He served as paper reviewer for over thirty international conferences and journals, and developed a system of Security Health Checking, which achieved IBM System Services Client Value Recognition Win for Automation Innovation for Global Use.

Shoushan Luo received the B.Sc. degree from Beijing Normal University and M.Sc. and Ph.D. degrees from Beijing University of Posts and Telecommunications, Beijing, China. He is a professor and Ph.D. supervisor in the School of Cyberspace Security at Beijing University of Posts and Telecommunications. His main research interests include information security, cryptography and secure multi-party computation.

Wei Ni (M'09–SM'15) received the B.E. and Ph.D. degrees in Electronic Engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. Currently he is a Team Leader, Senior Scientist, and Project Leader at CSIRO, Sydney, Australia. He also holds honorary positions at the University of New South Wales (UNSW), Macquarie University (MQ) and the University of Technology Sydney (UTS). Prior to this he was a postdoctoral research fellow at Shanghai Jiaotong University (2005 – 2008), Research Scientist and Deputy Project Manager at the Bell Labs R&I Center, Alcatel/Alcatel-Lucent (2005 – 2008), and Senior Researcher at Devices R&D, Nokia (2008-2009). His research interests include optimization, game theory, graph theory, as well as their applications to network and security.

Dr Ni serves as Vice Chair of IEEE NSW VTS Chapter since 2018, served as Editor for Hindawi Journal of Engineering from 2012 – 2015, Secretary of IEEE NSW VTS Chapter from 2015-2018, Track Chair for VTC-Spring 2017, Track Co-chair for IEEE VTC-Spring 2016, and Publication Chair for BodyNet 2015. He also served as Student Travel Grant Chair for WPMC 2014, a Program Committee Member of CHINACOM 2014, a TPC member of IEEE ICC14, ICCC15, EICE14, and WCNC10.