A Distributed Multi-Authority Attribute Based Encryption Scheme for Secure Sharing of Personal Health Records

Harsha S. Gardiyawasam Pussewalage
Department of ICT, University of Agder
N-4898, Grimstad, Norway
harsha.sandaruwan@uia.no

Vladimir A. Oleshchuk
Department of ICT, University of Agder
N-4898, Grimstad, Norway
vladimir.oleshchuk@uia.no

ABSTRACT

Personal health records (PHR) are an emerging health information exchange model, which facilitates PHR owners to efficiently manage their health data. Typically, PHRs are outsourced and stored in third-party cloud platforms. Although, outsourcing private health data to third party platforms is an appealing solution for PHR owners, it may lead to significant privacy concerns, because there is a higher risk of leaking private data to unauthorized parties. As a way of ensuring PHR owners' control of their outsourced PHR data, attribute based encryption (ABE) mechanisms have been considered due to the fact that such schemes facilitate a mechanism of sharing encrypted data among a set of intended recipients. However, such existing PHR solutions suffer from inflexibility and scalability issues due to the limitations associated with the adopted ABE mechanisms. To address these issues, we propose a distributed multi-authority ABE scheme and thereby we show how a patient-centric, attribute based PHR sharing scheme which can provide flexible access for both professional users such as doctors as well as personal users such as family and friends is realized. We have shown that the proposed scheme supports on-demand user revocation as well as secure under standard security assumptions. In addition, the simulation results provide evidence for the fact that our scheme can function efficiently in practice. Furthermore, we have shown that the proposed scheme can cater the access requirements associated with distributed multi-user PHR sharing environments as well as more realistic and scalable compared with similar existing PHR sharing schemes.

KEYWORDS

Access control, Attribute based encryption, Security, Personal health records

1 INTRODUCTION

Personal health records (PHR) are health information of patients, which are maintained and kept under the control of themselves. There are several advantages of using PHRs from the patients' perspective. PHRs induce patient-centric health information sharing capability given that the private health data is always under the control of the patient. In addition, there are practical restrictions with regard to sharing of health information of patients between healthcare deliverers due to privacy and legal constraints. Hence, it would be an advantage to have a PHR which is shareable with different care deliverers.

The use of PHRs is an attractive option, however the difficulty associated with management of health information induces a significant management overhead to the owners of health records. But the utilization of cloud platforms for management of health information helps to resolve the aforementioned issue since it allows the PHRs to be outsourced to cloud infrastructures instead of storing them locally. This approach potentially leads to a better availability of health data as well as relieving the patients from the burden of maintaining them. However, considering the fact that cloud infrastructures are managed by third-parties who may be curious about the data being stored, privacy concerns have been raised on the stored data [6][11]. Also, such storage servers could become targets for various malicious activities and may lead to illegal exposure of sensitive data belonging to patients [14]. Therefore, it is crucial to adopt necessary privacy preserving mechanisms to ensure the security and privacy of outsourced PHRs of patients.

A promising approach would be to encrypt the PHR data before being outsourced to a cloud platform, so that the confidentiality of private health data is kept preserved. To achieve this, incorporation of attribute based encryption (ABE) schemes have been considered lately [15]. ABE schemes can be divided into two categories based on their functionality, as key-policy attribute based encryption (KP-ABE) schemes and ciphertext policy attribute based encryption (CP-ABE) schemes. In a KP-ABE scheme the ciphertext is associated with a set of attributes and users' secret keys are encoded with attribute based access structures [10]. If the access structure associated with a user's secret key satisfies the set of attributes which is used to generate a specific ciphertext, the user will be able to decrypt the ciphertext with the help of his secret key. CP-ABE can be considered as the dual of KP-ABE, where the ciphertext is encoded with the access structure while the users' secret keys are encoded with attributes [5]. In relation to a PHR sharing application, CP-ABE schemes seem to be more conducive compared to KP-ABE schemes, given that the PHR owner will be able to specify the intended recipients through an attribute based access structure while a user who possesses a set of attributes that satisfies the access structure could potentially decrypt the encrypted PHR data using his relevant secret keys.

Another important fact that must be considered is the access requirements for a PHR sharing scenario would be complex in nature where potential recipients may come from different domains such as the patients' relatives and healthcare professionals from

SACMAT'17, June 21-23, 2017, Indianapolis, IN, USA
© 2017 ACM. ACM ISBN 978-1-4503-4702-0/17/06…$15.00
DOI: http://dx.doi.org/10.1145/3078861.3078880

different care providers. Hence, the flexibility of the underlying access control mechanism is of paramount importance to cater the demands of access requirements. CP-ABE schemes do have the potential as we have mentioned above, but the existing schemes have some drawbacks which hinder the effectiveness and applicability with respect to secure sharing of PHRs.

The remainder of this paper is organized as follows. A brief description of related work is presented in Sec. 2 followed by the contributions of this paper in Sec. 3. In Sec. 4, we present the case that we have addressed along with the security requirements to be maintained in the proposed scheme. Preliminary knowledge corresponding to the proposed scheme is presented in Sec. 5. An overview of the proposed PHR sharing scheme is given in Sec. 6 while the phases of the proposed scheme are presented in detail in Sec. 7. In Sec. 8, we analyze the security of our scheme whereas in Sec. 9, we evaluate the performance and efficiency in terms of associated computational cost. Finally, we compare the proposed PHR sharing scheme with similar existing schemes in Sec. 10 before the paper is concluded in Sec. 11.

2 RELATED WORK

In this section, we summarize the most prominent existing research work on utilizing CP-ABE methods for secure sharing of PHRs while discussing associated weaknesses of the considered solutions.

Ibraimi et al. [12] have proposed a secure PHR sharing scheme using the CP-ABE scheme in [13] while introducing the concept of personal and public domains. The solution consists with a trusted authority (TA) for managing attributes in the public domain while the PHR owner himself acting as the TA for the personal domain for the purpose of issuing attributes relevant for the personal domain. Thus, the PHR owner can encrypt the private health data using an attribute based access structure, allowing only the users who have attributes that satisfy the associated access structure can successfully decrypt the data. Although this is a patient-centric solution, it has some drawbacks as well. The main issue is the use of a single TA for administrating the user attributes of the public domain. This approach could not only lead to a single point of failure but also may cause key-escrow problems given the fact that the TA can access all the encrypted files. In addition, the adoption of a single TA for managing all attributes in the public domain may also not be a realistic assumption with respect to an e-health environment which is (generally) inherently distributed. For instance, consider a scenario where a user's PHR requires to be encoded with attributes belonging to two healthcare providers. In such a situation, it is not realistic to assume that the attributes related to both organizations are handled by the same central TA, while it is more realistic to think of a scenario where each organization acts as an attribute authority to issue own attributes. We have noticed that cloud based personal health information sharing schemes for a similar setting but with a central TA are proposed in [1, 3, 4, 8, 16].

In the quest of dealing with the aforementioned issue, Li et al. [14] proposed an ABE based PHR sharing scheme using multiple authorities such that each authority administrates a disjoint set of attributes. Thus, users belonging to the public domain can ascertain required attributes from the relevant attribute authority (AA) while users in the private domain ascertain the attributes from the PHR owner similar to [12]. In this solution, the authors have utilized the multi-authority attribute based encryption (MA-ABE) scheme proposed by Chase and Chow [7] to achieve the secure sharing of PHRs. However, the main drawback of this MA-ABE scheme is that it requires users to obtain at least one attribute from each AA for the proper functioning of the encryption scheme. Due to this restriction in the utilized encryption scheme, the PHR sharing scheme in [14] is far from being effective in practice.

3 OUR CONTRIBUTIONS

In order to realize a flexible cloud based PHR sharing scheme, it is necessary to utilize a flexible and scalable multi-authority ABE scheme. As we have pointed out in Sec. 2, the lack of scalability and flexibility in existing distributed multi-authority ABE schemes have affected the evolution of such systems. Thus, the main contribution of this paper is constructing a novel distributed multi-authority CP-ABE scheme and propose a flexible cloud based PHR sharing scheme utilizing the proposed multi-authority CP-ABE construction.

In the proposed PHR sharing scheme, we define two user domains (as in [14]) public and private where public domain consists of healthcare professionals and personal domain consists with family and friends. We use a set of distributed, public attribute authorities (AAs) to manage public attributes while the PHR owner manages the private attributes. With the proposed multi-authority CP-ABE scheme, we are able to provide fine-grained PHR access for the users from both domains without requiring attributes from each existing AA, as it was in [14]. The proposed multi-authority CP-ABE scheme is collusion resistant, hence two or more users will not be able to collude their attributes and gain access to PHR data, given that it is not possible on their own. Furthermore, the proposed scheme supports on-demand revocation, which ensures that a user will not be able to use a revoked attribute for further access.

We also show that the proposed CP-ABE scheme is secure and thereby it can enforce the intended security and privacy requirements associated with the PHR sharing scenario. In addition, we also provide evidence for the feasibility and scalability of the proposed PHR sharing scheme in terms of associated computational cost based on simulation results.

4 CASE DESCRIPTION AND SECURITY REQUIREMENTS

In this section, we describe the PHR sharing scenario for which the multi-authority CP-ABE scheme is proposed. We also present the system model corresponding to the considered case while stating the security and privacy requirements that must be satisfied.

4.1 Case Description

We consider a cloud based PHR system which involves multiple PHR owners and PHR users. PHR owner is a patient who is interested in outsourcing his private health data while having the full control of outsourced data. PHR owner is capable of uploading, deleting PHR information along with sharing them among a set of PHR users based on user attributes. PHR users include both users from the professional domain such as healthcare professionals, insurance companies, etc. and users from the personal domain such as family and friends. PHRs are stored in a central cloud repository which

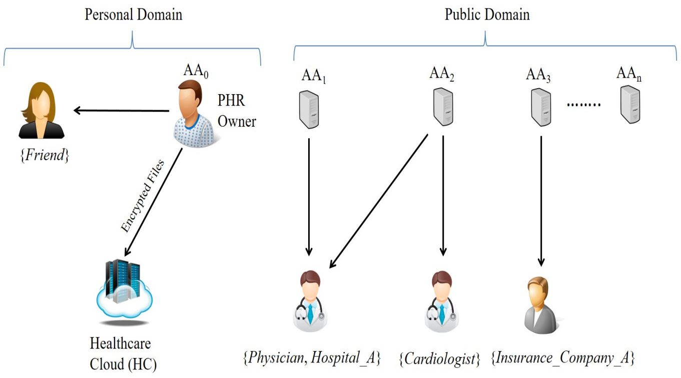


Figure 1: System model

we denote here on in as the healthcare cloud (HC). Users should be allowed to access the PHR data of patients as long as they satisfy the (attribute based) access requirements specified by the PHR owner.

We assume that the healthcare cloud is semi-trusted, which means that it will follow the specified operational protocol while being curious on the data being stored. We further assume that the users may also be curious on the stored data, hence they may want to extract more information than what they are allowed through colluding attributes with fellow users.

As shown in Figure 1, we use a set of distributed public attribute authorities (AAs) with each AA is responsible for managing a disjoint set of attributes and issuing PHR users with relevant attributes (in the form of secret keys) upon validating attribute requirements for the requested attributes. PHR owner also acts as an AA for providing secret keys relevant for the attributes of the personal domain for personal domain users.

4.2 Security Requirements

The main security requirements that we intend to achieve through the proposed scheme are outlined below.

- *Confidentiality of PHR data*: PHR data of patients must be kept secret from unauthorized parties.
- *Patient-centric access control*: PHR owners should have the full control of the outsourced health data, allowing them to determine who are eligible to access them.
- *Resistant to attribute collusion*: Multiple users should not be able to collude their attributes and decrypt PHR data.
- *Efficient on-demand user revocation*: Whenever an attribute of a certain user is no longer valid, the user should not be able to decrypt PHR data using the secret keys associated with the revoked attribute.

5 PRELIMINARIES

This section is dedicated to provide the required background details associated with the proposed PHR sharing scheme.

5.1 PHR Access Structure

We associate each PHR with a unique PHR identification PHR_{id}. Each PHR can have many information categories such as personal information, diagnosis, medications, allergies, emergency data, etc. and we define them as PHR objects (PHR_{obj}) of a PHR. Hence, each PHR can have many different PHR objects. Moreover, we associate each PHR_{obj} with an access structure ($\mathcal{T}$) which governs the attribute requirement for accessing the PHR_{obj}. We define an access structure as a Boolean statement with disjunction ($\lor$) and conjunction ($\land$) operations combining subject attributes. An example access structure $\mathcal{T}(P_k, CD)$ relevant for the PHR_{obj} cardiac diagnosis (CD) corresponding to $PHR_{id} = P_k$ is shown below.

$$\mathcal{T}(P_k, CD) : (Cardiologist \land (Hospital\_A \lor Hospital\_B)) \lor Family$$

This statement states that any user who is a family member of the PHR owner or work as a cardiologist at hospital A or hospital B is authorized to access the cardiac diagnosis PHR_{obj} associated with the PHR identification P_k.

5.2 Access Sub-structures

We represent an access structure $\mathcal{T}$ as the disjunction of a set of sub-structures $\{\mathcal{T}_i\}_{i=1,2,\ldots,q}$ such that, $\mathcal{T} = \mathcal{T}_1 \lor \mathcal{T}_2 \lor \ldots \lor \mathcal{T}_q$, where each $\mathcal{T}_i$ is a conjunction of some subject attributes (Boolean statement of $\land$ operations). We call each $\mathcal{T}_i$ as an access sub-structure of $\mathcal{T}$.

6 OVERVIEW OF THE PHR SHARING SCHEME

As explained in Sec. 4, our system consists with multiple distributed public AAs. Each AA manages a disjoint set of attributes and issues attributes for the users belonging to the public domain. In addition, PHR owner acts as a private AA to provide attributes that specify personal relationships such as for example *Family* for the personal domain users. During initialization of the system, every AA first defines a set of secret exponents and public exponents in such a way that each administered attribute is associated with a distinct secret attribute exponent and a corresponding public attribute exponent. Initialization of AAs can function independently without requiring any global coordination.

PHR users can obtain attributes from the relevant AAs by providing evidence that they are eligible for the requested attributes. If the AA responsible for the requested attribute is satisfied with regard to the eligibility of the attribute requesting user to ascertain the requested attribute, the AA will issue the relevant secret keys for the user. We assume that the secret keys are securely handed over to the corresponding user.

When a PHR owner wants to outsource a PHR_{obj} to HC, he should first construct the access structure $\mathcal{T}$. Note that the attributes in $\mathcal{T}$ can have a combination of attributes from both personal and public domains. Then, the PHR_{obj} is encrypted with the help of public attribute keys corresponding to the attributes in $\mathcal{T}$ defined by the relevant AA (details will be given in the following sections). The generated ciphertext along with $\mathcal{T}$ is sent to the HC to be stored. When a user is required to access a specific PHR_{obj}, he can request for the required PHR_{obj} from the HC by sending a PHR access request indicating the PHR_{id} and the relevant PHR_{obj}. However, the user will only be able to decrypt the encrypted PHR_{obj}, if and only if the user has a set of secret keys corresponding to a set of attributes which satisfies the $\mathcal{T}$ associated with the encrypted PHR_{obj}.

7 MULTI-AUTHORITY CP-ABE (MA-CP-ABE) SCHEME

In this section, we present the proposed multi-authority CP-ABE (MA-CP-ABE) scheme in detail. Our scheme is influenced by the single authority CP-ABE scheme of L. Ibraimi et al. [13]. We describe the functionality of the proposed MA-CP-ABE scheme by dividing it into five main phases: system initialization, key distribution, PHR encryption, PHR decryption and user revocation.

7.1 System Initialization

To initialize the system, first a set of global public parameters are generated which are shared among all AAs. AAs agree on two multiplicative cyclic groups $\mathbb{G}_0, \mathbb{G}_1$ of prime order p with g being a generator of $\mathbb{G}_0$ and a bi-linear map [9] $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ along with a secure hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ that maps each user identity string to a unique value in $\mathbb{Z}_p^*$. The user identity should be a unique identifier for a given user such as for example an E-mail address. Then, AAs publish the set of global public parameters of $(\mathbb{G}_0, \mathbb{G}_1, H, e, g, p)$. Therefore, any new AA can be globally initialized by acquiring the set of global parameters which are shared by the existing AAs. Then, each AA (including PHR owner) is locally initialized, and the initialization procedure is described below. We assume that k^{th} AA is denoted with AA_k while the attribute set administered by AA_k is denoted by AT^k.

- AA_k chooses two random exponents $\alpha_k, \beta_k \in \mathbb{Z}_p^*$ and computes $X_k = g^{\beta_k}$, $Y_k = e(g,g)^{\alpha_k}$. Then a unique random identifier $t_{k,i} \in \mathbb{Z}_p^*$ for each element i in AT^k is selected. In addition, each attribute administered by AA_k is also associated with a public attribute exponent $T_{k,i}$, where $T_{k,i} = g^{t_{k,i}}$.
- AA_k will keep $\{\alpha_k, \beta_k, t_{k,i}\}_{i=1,2,\ldots,|AT^k|}$ as the master secret (MK_k) and publish $\{X_k, Y_k, T_{k,i}\}_{i=1,2,\ldots,|AT^k|}$ as the authority's public key denoted by PK_k.

7.2 Attribute Key Distribution

Let us assume that user U_m wants to acquire attribute keys for the set of attributes AT_m. In addition, assume that AT_m^k denotes the subset of attributes in AT_m which should be acquired from AA_k. Suppose that AA_k has already validated the eligibility of U_m for ascertaining the requested attributes. The process of attribute key distribution is as follows.

- AA_k first maps the identity of U_m (we use the E-mail address as the user identity) to a unique identifier $r_m \in \mathbb{Z}_p^*$ with the use of the secure hash function H.
- Then, a secret key for each requesting attribute is generated as described below. If the secret key set is denoted by SK_m^k,
$$SK_m^k = \{sk_0^k, sk_i^k\}_{i=1,2,\ldots,|AT_m^k|}$$
and,
$$sk_0^k = g^{\frac{\alpha_k - r_m}{\beta_k}}, \tag{1}$$
$$sk_i^k = g^{\frac{r_m}{t_{k,i}}}, \tag{2}$$
where $t_{k,i}$ is the MK component of the i^{th} attribute in AT_m^k defined by AA_k. Note that secret key component sk_0^k relates the user identity to the identity of issuing authority AA_k whereas the secret key component sk_i^k relates the user identity to the attribute itself.
- The generated secret key set is securely transferred to U_m.

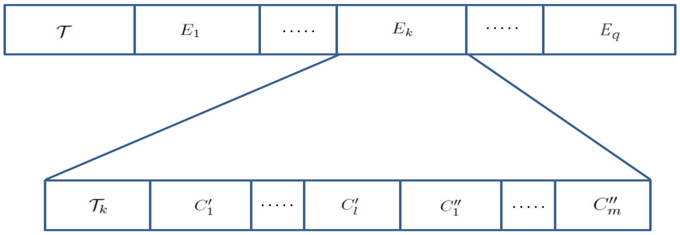


Figure 2: Structure of the ciphertext $E(M)$

7.3 PHR Encryption

Let us assume that the PHR owner wants to encrypt PHR data $M \in \mathbb{G}_1$, which includes information on his *allergies*. First, he generates the access structure $\mathcal{T}$ and deduce a set of access sub-structures $\{\mathcal{T}_k\}_{k=1,2,\ldots,q}$ as mentioned in Sec. 5.2. Thus the ciphertext of M encoded with $\mathcal{T}$ is given by $E(M)$,
$$E(M) = (\mathcal{T}, \{E_k\}_{k=1,2\ldots,q}),$$
where E_k denotes the ciphertext of M encoded with access substructure $\mathcal{T}_k$. The structure of the ciphertext $E(M)$ is illustrated in Figure 2 and the process of computing E_k is described below.

Let us assume that k^{th} sub-structure $\mathcal{T}_k$ contains m attributes and they are administered by l AAs such that, $l \leq n$, where n is the total number of AAs in the system. Note that any AA may administrate more than one attribute of the considered m attributes. Then, we can represent the ciphertext E_k using ciphertext components $C_0, \{C'_i\}_{i=1,2,\ldots,l}$ and $\{C''_i\}_{i=1,2,\ldots,m}$ such that,
$$E_k = (\mathcal{T}_k, C_0, \{C'_i\}_{i=1,2,\ldots,l}, \{C''_i\}_{i=1,2,\ldots,m}).$$

The computation of the aforementioned ciphertext components of E_k is as follows. PHR owner first generates a random exponent $s \in \mathbb{Z}_p^*$ and using the public keys of l AAs, he computes ciphertext components C_0 and $\{C'_i\}_{i=1,2,\ldots,l}$ such that,
$$C_0 = M \prod_{i=1}^{l} Y_i^s = Me(g,g)^{s\sum_{i=1}^{l}\alpha_i}, \tag{3}$$
$$C'_i = X_i^s = g^{\beta_i s}. \tag{4}$$

To compute $\{C''_i\}$ a secret share of s is assigned for each attribute in $\mathcal{T}_k$ by following the steps given below.

- For each attribute in $\mathcal{T}_k$ except the last, a random exponent $s_i \in \mathbb{Z}_p^*$ is assigned while the last element is assigned the value equals to $ls - \sum_{i=1}^{m-1} s_i$.
- Then, the PHR owner computes $\{C''_i\}_{i=1,2,\ldots,m}$ such that,
$$C''_i = T_i^{s_i} \tag{5}$$
where T_i corresponds to the public attribute exponent of the i^{th} attribute in $\mathcal{T}_k$.

Similarly, PHR owner generates ciphertexts relevant for all the sub-structures of $\mathcal{T}$. Finally, the PHR owner sends the ciphertext of M, $E(M)$ along with PHR identification (PHR_{id}) and PHR_{obj} information ($PHR_{obj} = allergies$) to the HC to be stored as a part of his PHR.

7.4 PHR Decryption

Suppose U_m wants to access a specific PHR_{obj} stored in the HC. U_m should first send an access request indicating the PHR_{id} and PHR_{obj} corresponding to the access required PHR information to the HC. Then, the HC fetches the corresponding $\mathcal{T}$ associated with the requested PHR_{obj} and sends it back to U_m. Let us assume that the attribute set owned by U_m is denoted with AT_m. Then, U_m determines the smallest subset of attributes AT'_m which he owns that satisfies the received $\mathcal{T}$. Based on AT'_m, U_m generates a sub-structure $\mathcal{T}'$ and sends it to the HC. According to the received $\mathcal{T}'$, HC fetches the corresponding PHR ciphertext E' and sends it back to U_m which enables him to decrypt the encrypted data using the relevant attribute secret keys. The decryption process is as follows.

For illustrative purposes, let us assume that the received ciphertext E' is encoded with m attributes which are administered by l AAs. Then,

$$E' = (\mathcal{T}', C_0, \{C'_i\}_{i=1,2,\dots,l}, \{C''_i\}_{i=1,2,\dots,m}),$$

where C_0, C'_i and C''_i are given in (3) - (5) respectively. Further assume that $\{sk_i\}_{i=1,2\dots,m}$ denotes the relevant attribute secret key set owned by U_m for the attribute subset AT'_m and $\{sk_0^i\}_{i=1,2,\dots,l}$ refers to the set of secret key components which relates the identity of U_m to the l AAs who issued the m attributes. According to (1) and (2) $sk_i = g^{\frac{rm}{t_i}}$ and $sk_0^i = g^{\frac{\alpha_i - rm}{\beta_i}}$, where t_i denotes the MK component defined by the AA who administrates the corresponding attribute. In order to decrypt E', U_m first computes,

$$\prod_{i=1}^{m} e(C''_i, sk_i) = \prod_{i=1}^{m} e(T_i^{s_i}, g^{r_m/t_i}) = e(g,g)^{ls(r_m)}. \quad (6)$$

Thereafter U_m computes,

$$\prod_{i=1}^{l} e(C'_i, sk_0^i) = \prod_{i=1}^{l} e(g^{s\beta_i}, g^{\frac{\alpha_i - rm}{\beta_i}}) = e(g,g)^{s\sum_{i=1}^{l}\alpha_i - ls(r_m)}. \quad (7)$$

From (6) and (7) U_m can compute the support string Ω such that,

$$\Omega = e(g,g)^{ls(r_m)} e(g,g)^{s\sum_{i=1}^{l}\alpha_i - ls(r_m)} = e(g,g)^{s\sum_{i=1}^{l}\alpha_i}. \quad (8)$$

Then, U_m will be able to discover M using (3) and (8) as follows.

$$\frac{C_0}{e(g,g)^{s\sum_{i=1}^{l}\alpha_i}} = \frac{Me(g,g)^{s\sum_{i=1}^{l}\alpha_i}}{e(g,g)^{s\sum_{i=1}^{l}\alpha_i}} = M$$

7.5 On-demand User Revocation

When a particular attribute belonging to a specific user is revoked, the user should not be able to use the secret keys related to the revoked attribute in any further transactions. In our proposed scheme, the revocation process is handled by the AA which is responsible for the attribute to be revoked. We summarize the revocation process as follows. Suppose AA_k requires to revoke the attribute ω from U_m. In addition, assume that the secret exponent associated with the attribute ω defined by AA_k is given by t_ω.

- First of all, a new random secret exponent t'_ω for the attribute to be revoked ω is selected and based on the new secret, the associated public attribute exponent $g^{t'_\omega}$ is generated and published.
- According to (2), it is evident that modification to the secret attribute exponent of a given attribute affects the secret keys associated with the considered attribute. Hence, the relevant secret keys need to be updated accordingly. Therefore, new secret keys are generated (using the new secret exponent t'_ω) and sent to the users who obtained the attribute ω previously except the user to be revoked (U_m).
- Given that the public attribute exponent related to the revoked attribute is modified, messages encoded with the attribute ω will be contaminated. We elaborate this further through the following example.

Consider the encryption of message M, with a sub-structure $\mathcal{T}_1 = \omega$. Let us assume that $E(M)$ represents the encryption of M prior to the revocation of attribute ω. Then, according to Sec. 7.3, $E(M) = (\mathcal{T}_1, C_0, C', C'')$, where $C_0 = MY_k^s$, $C' = X_k^s$ and $C'' = g^{t_\omega s}$. Note that the alteration of the public attribute exponent of attribute ω will only contaminate the ciphertext component C'' given that C_0, C' ciphertext components are independent of the public attribute exponent of attribute ω (revoking attribute). We use a re-encryption mechanism to update the contaminated ciphertext component. The process is described below using the aforementioned example.

- AA_k first generates a re-encryption key $RE_{key} = {}^{t'_\omega}/_{t_\omega}$.
- Then, AA_k sends RE_{key} to HC which enables the HC to re-encrypt the contaminated ciphertext components. If the corresponding updated ciphertext component is given by C''_{new} then,
  $$C''_{new} = C''^{RE_{key}} = g^{t_\omega s \frac{t'_\omega}{t_\omega}} = g^{t'_\omega s}.$$
- Thus, the ciphertext corresponding to the encryption of M after the revocation is given by $E_{new}(M)$, then

$$E_{new}(M) = (\mathcal{T}_1, C_0, C', C''_{new}).$$

After the revocation, the revoked user (U_m) will not be able to use his old secret keys corresponding to attribute ω due to the fact that the public attribute exponent related to the attribute ω is already modified. Given that other users who have ascertained the attribute ω from AA_k are issued with new secret keys, they will be able to use the new secret keys for future transactions.

8 SECURITY ANALYSIS

In this section, we evaluate the security of the proposed MA-CP-ABE scheme and discuss some important security properties of the proposed scheme. First, we introduce the following assumptions on which the security of the proposed scheme is based upon.

Discrete Logarithm (DL) Assumption: Suppose $\mathbb{G}$ is a cyclic group of order p with g being the generator. Given (g, g^a) there is no probabilistic polynomial time algorithm which can compute $a \in \mathbb{Z}_p^*$ with non-negligible probability.

Decisional Bi-linear Diffie-Hellman (DBDH) Assumption: Suppose $\mathbb{G}$ is a cyclic group of order p with a generator g and e being a bi-linear map. Given that $a, b, c, z \in \mathbb{Z}_p^*$, there is no polynomial-time adversary can distinguish the tuple $(g^a, g^b, g^c, e(g,g)^{abc})$ from the tuple $(g^a, g^b, g^c, e(g,g)^z)$ with non-negligible probability.

8.1 Resistant Against Chosen Plaintext Attacks

Our intention is to demonstrate that the proposed MA-CP-ABE scheme is indistinguishable under chosen plaintext attacks (IND-CPA secure), given that the DBDH assumption is held. Suppose there exist a polynomial-time adversary $\mathcal{A}$ that can break the MA-CP-ABE scheme with a non-negligible advantage ϵ. We show that it is possible to build a simulator $\mathcal{S}$ that can play the DBDH game with an advantage $\epsilon/2$ as follows.

Let us assume that $\mathbb{G}_0$ and $\mathbb{G}_1$ are two cyclic groups with g being a generator of $\mathbb{G}_0$. Further assume that e is an efficiently computable bi-linear map and $a, b, c, z \in \mathbb{Z}_p^*$ are randomly chosen. Suppose, the simulator $\mathcal{S}$ is fed with a DBDH instance $(g, g^a, g^b, g^c, R_\delta)$ in which R_δ is set through flipping a fair coin δ where,

$$\begin{aligned} R_\delta &= e(g,g)^{abc}, \quad if \quad \delta = 0 \\ &= e(g,g)^z, \quad if \quad \delta = 1. \end{aligned}$$

The game proceeds as follows.

Initialization phase: The adversary $\mathcal{A}$ selects a challenge access sub-structure $\mathcal{T}'$ with attributes from l out of n AAs, and sends it to $\mathcal{S}$. Note that we denote the attribute set in $\mathcal{T}'$ by AT'.

Setup: We assume that the simulator $\mathcal{S}$ simulates on-behalf of all n AAs. For each attribute ω_i in AT', the simulator $\mathcal{S}$ chooses a random element $q_i \in \mathbb{Z}_p^*$ and thereby sets the public attribute exponent for each element in AT' as $T_i = g^{q_i}$. For all the other attributes (which are not elements in AT'), the simulator $\mathcal{S}$ sets $T_i = g^{b/q_i}$. Furthermore, $\mathcal{S}$ selects a set of n random exponents $\{d_i, \beta_i\}_{i=1,2,..,n} \in \mathbb{Z}_p^*$ and by allowing, $e(g,g)^{\alpha_i} = e(g,g)^{ab/l} e(g,g)^{d_i}$, $\mathcal{S}$ implicitly sets each AA's secret key $\alpha_i = \frac{ab}{l} + d_i$. Then, all the public parameters of the simulator are forwarded to the adversary $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ sends attribute key requests to the simulator $\mathcal{S}$ for the attributes which are not elements in AT'. For the adversary $\mathcal{A}$, the simulator selects a random exponent $\hat{r} \in \mathbb{Z}_p^*$ and generates the secret key sk_0^i which relates the identity of the issuing authority and the identity of the adversary as follows.

$$sk_0^i = g^{\frac{d_i - \hat{r}b}{\beta_i}} = g^{\frac{\alpha_i - (\hat{r}b + \frac{ab}{l})}{\beta_i}}$$

Then, the Simulator $\mathcal{S}$ should generate the attribute secret keys sk_i corresponding to the each requested attribute. To have a valid simulation of attribute secret keys, sk_i must be in the form,

$$sk_i = g^{\frac{(\hat{r}b + \frac{ab}{l})}{t_i}} = g^{\frac{(\hat{r}b + \frac{ab}{l})q_i}{b}}.$$

Hence, $\mathcal{S}$ sets $sk_i = g^{\hat{r}q_i} g^{\frac{aq_i}{l}}$. It is evident that this is a valid simulation of secret keys, since, $sk_i = g^{\frac{(\hat{r}b + \frac{ab}{l})q_i}{b}} = g^{\hat{r}q_i} g^{\frac{aq_i}{l}}$. Then, $\mathcal{S}$ sends the secret keys $(sk_0^i,\ sk_i)$ for the attributes that are not elements of the challenge access sub-structure $\mathcal{T}'$ to $\mathcal{A}$.

Challenge phase: $\mathcal{A}$ sends two plaintexts $M_0, M_1 \in \mathbb{G}_1$ to $\mathcal{S}$. Then $\mathcal{S}$ will encrypt one of M_0, M_1 according to $\mathcal{T}'$ by flipping a fair binary coin v. To encrypt M_v, the simulator $\mathcal{S}$ first computes C_0 and $\{C_i'\}_{i=1,2,\ldots,l}$ such that,

$$\begin{aligned} C_0 &= M_v \prod_{i=1}^{l} Y_i^c = M_v e(g,g)^{\sum_{i=1}^{l}(\alpha_i)c} = M_v e(g,g)^{\sum_{i=1}^{l}(\frac{ab}{l} + d_i)c} \\ &= M_v e(g,g)^{abc} e(g,g)^{\sum_{i=1}^{l}(d_i)c} = M_v R_\delta e(g,g)^{\sum_{i=1}^{l}(d_i)c} \\ C_i' &= X_i^c = g^{\beta_i c}. \end{aligned}$$

For each attribute in AT' except the last, a random exponent $h_i \in \mathbb{Z}_p^*$ is assigned while the last element is assigned the value equals to $lc - \sum_{i=1}^{m-1} h_i$. Then, $\mathcal{S}$ computes $\{C_i''\}_{i=1,2,\ldots,m}$ such that, $C_i'' = T_i^{h_i}$. Then, $\mathcal{S}$ forwards the resulting ciphertext E_v to $\mathcal{A}$.

$$E_v = (\mathcal{T}', C_0, \{C_i'\}_{i=1,2,\ldots,l}, \{C_i''\}_{i=1,2,\ldots,m})$$

Phase 2: The simulator $\mathcal{S}$ acts exactly as it did in Phase 1.

Guess: The adversary $\mathcal{A}$ submits a guess $v' \in \{0, 1\}$. If $v' = v$ the simulator $\mathcal{S}$ will guess that $\delta = 0$ and outputs a 0 indicating that $R_\delta = e(g,g)^{abc}$. This will simulate a valid random encryption of the message M_v under the access structure $\mathcal{T}'$. If $v' \neq v$, simulator $\mathcal{S}$ outputs a 1 indicating $R_\delta = e(g,g)^z$, meaning that the adversary gains no information about the plaintext M_v. Thus, we can come to the following conclusions.

- If $v' \neq v$, then the advantage of $\mathcal{A}$ is given by, $Pr[v' \neq v | R_\delta = e(g,g)^z] = \frac{1}{2}$.
- We assumed that the advantage of the adversary $\mathcal{A}$ to break the MA-CP-ABE scheme is given by ϵ. Hence, the advantage of the adversary $\mathcal{A}$ in the DBDH game when $v' = v$ is given by, $Pr[v' = v | R_\delta = e(g,g)^{abc}] = \frac{1}{2} + \epsilon$.
- Given that the simulator $\mathcal{S}$ guesses $\delta = 0$ when $v' = v$ and $\delta = 1$ when $v' \neq v$, the total advantage of the simulator $\mathcal{S}$ in the DBDH game is given by,

$$\frac{1}{2}(Pr[v' \neq v | R_\delta = e(g,g)^z] + Pr[v' = v | R_\delta = e(g,g)^{abc}]) - \frac{1}{2} = \frac{\epsilon}{2}.$$

Therefore, we can conclude that the proposed MA-CP-ABE scheme is IND-CPA secure given that the DBDH assumption is held.

8.2 Resistant Against Attribute Collusion

For any attribute based system, it is crucial to prevent attribute collusion which may potentially leads to illegitimate access of resources. We ensure the prevention of collusion attacks via infusing identity related characteristic to each obtained secret key relevant for a given attribute. Suppose two PHR users U_1 and U_2 wish to collude secret keys of two attributes ω_1, ω_2 which are owned by U_1 and U_2 respectively. Further assume that ω_1 is administered by AA_1 and ω_2 is administered by AA_2 while t_1, t_2 denote the corresponding attribute secret exponents defined by the respective AA. Then, according to (3) - (5) the ciphertext E for the plaintext M encoded with the access sub-structure $\mathcal{T}' = \omega_1 \wedge \omega_2$ is given by,

$$E = (\mathcal{T}', C_0, \{C_i'\}_{i=1,2}, \{C_i''\}_{i=1,2}),$$

where $C_0 = Me(g,g)^{(\alpha_1+\alpha_2)s}$, $C_i' = g^{\beta_i s}$ and $C_i'' = g^{t_i s_i}$. In addition, the secret keys of U_1 and U_2 corresponding to attributes ω_1 and ω_2 are given by $(g^{r_1/t_1}, g^{\frac{\alpha_1 - r_1}{\beta_1}})$ and $(g^{r_2/t_2}, g^{\frac{\alpha_2 - r_2}{\beta_2}})$ respectively. In the attempt to decrypt E, according to (6) - (7) U_1 and U_2 can compute,

$$temp_1 = e(g^{t_1 s_1}, g^{r_1/t_1}) e(g^{t_2(2s - s_1)}, g^{r_2/t_2}), \tag{9}$$

$$temp_2 = e(g^{\frac{\alpha_1 - r_1}{\beta_1}}, g^{\beta_1 s}) e(g^{\frac{\alpha_2 - r_2}{\beta_2}}, g^{\beta_2 s}). \tag{10}$$

From (9) and (10) users can compute the helper string Ω such that,

$$\begin{aligned} \Omega &= temp_1 \cdot temp_2 \\ &= e(g,g)^{(\alpha_1+\alpha_2)s} e(g,g)^{r_1 s_1} e(g,g)^{r_2(2s - s_1)} e(g,g)^{-(r_1+r_2)s}. \end{aligned} \tag{11}$$

In order to recover M from C_0, the computation result in (11) must be equivalent to $e(g,g)^{(\alpha_1+\alpha_2)s}$. The aforementioned equivalence will only be possible if the following condition is held.

$$e(g,g)^{r_1 s_1} e(g,g)^{r_2(2s - s_1)} e(g,g)^{-(r_1+r_2)s} = 1 \tag{12}$$

The relation in (12) can only be maintained if and only of $r_1 = r_2$. Hence, it is infeasible to achieve a successful decryption via colluding attribute secret keys of more than one user.

8.3 Enforcing Confidentiality of PHR Data

In the proposed PHR sharing scheme, PHR data are encoded with an attribute based access structure specified by the PHR owner himself, which is only decryptable by a user who possesses a set of attributes that satisfies the associated access structure. Furthermore, we have shown that the proposed MA-CP-ABE scheme is secure against chosen plaintext attacks and attacks mounted via attribute collusion. Thus, the scheme can guard against the possibility of illegal disclosure of patient's private health data and thereby the confidentiality of data is maintained.

9 PERFORMANCE EVALUATION

In this section, we evaluate the performance of the utilized MA-CP-ABE scheme which functions as the underlying access control mechanism for the proposed PHR sharing scheme.

Computational overhead of the proposed MA-CP-ABE scheme heavily depends upon the overhead associated with encryption and decryption operations given that they require exponentiation and pairing operations in $\mathbb{G}_0$. Thus, we conduct simulations on determining approximated computational cost for the above mentioned two processes. The simulations were run on a Core i5, 2.5 GHz PC with 8 GB of RAM. In order to generate the necessary cyclic groups, we used the elliptic curve $y^2 = x^3 + x$ over a 512 bit finite field having a group order of 160 bits. We choose this parameter setting by considering the fact that it can generate keys having the equivalence security of 1024 bit RSA keys [2].

For the analysis, we simulated a simple multi-authority environment with 5 AAs each managing 10 attributes. We conducted simulations to determine the behavior of encryption time and decryption time with the number of attributes in a given access sub-structure $\mathcal{T}'$ under the following four cases.

- Case 1: All the attributes in $\mathcal{T}'$ belong to the same AA.
- Case 2: Attributes in $\mathcal{T}'$ belong to 2 AAs.
- Case 3: Attributes in $\mathcal{T}'$ belong to 3 AAs.
- Case 4: Attributes in $\mathcal{T}'$ belong to 4 AAs.

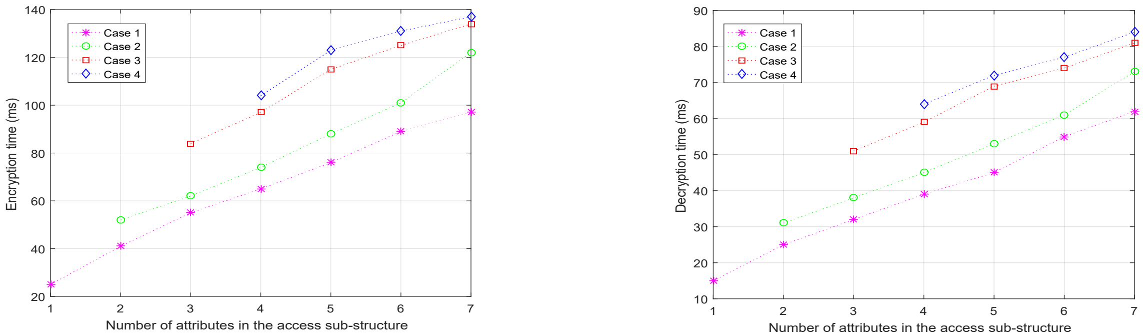
The obtained results are illustrated in Figure 3(a) and Figure 3(b). Figure 3(a) represents the variation of encryption time with respect to the four aforementioned cases while Figure 3(b) represents the variation of decryption time. Note that the corresponding encryption and decryption time values are average approximations obtained after 100 iterations. According to the results, it is obvious that both encryption time and the decryption time increase with the number of attributes in $\mathcal{T}'$. However, the variations exhibit nearly linear characteristics which speak for the scalability of the proposed MA-CP-ABE scheme. In addition, we can also observe that the decryption time is slightly lower than the encryption time since the decryption process requires less number of exponentiation operations compared to the process of encryption in the proposed MA-CP-ABE scheme. Note that we considered a maximum of 7 attributes in the access sub-structure $\mathcal{T}'$, since we rarely come across a sub-structure having more than 5 attributes in practice. However, given that the variation of computational cost is almost linear, it is fair to conclude that the proposed scheme is realistic and will function effectively under access sub-structures with a larger number of attributes as well.

Along with the computational cost, it is also important to analyze the size of an encrypted message when utilizing the proposed scheme under the considered parameter setting. Suppose a message $M \in \mathbb{G}_1$ is encrypted with a sub-structure $\mathcal{T}'$ such that the number of attributes in $\mathcal{T}'$ is m and they are administered by l AAs. Given that we use a 512 bit finite field to generate cyclic groups, each ciphertext component (C_0, C', C'') will be of 1024 bits. Hence, the size of the ciphertext is $1024(m + l + 1)$ bits. Although we assumed that $M \in \mathbb{G}_1$ (1024 bits), message sizes of health information could be much larger in practice, especially considering medical images. Thus, we can use an AES symmetric key K to encrypt the message M and then encrypt K with the proposed MA-CP-ABE scheme.

10 DISCUSSION

In this section, we compare our proposed PHR sharing scheme with similar schemes found in the literature. A comparison between our scheme and identified related works are tabulated in Table 1.

The PHR sharing scheme in [1] uses the CP-ABE scheme of J. Bethencourt et al. [5] as the underlying access control mechanism while the PHR sharing schemes proposed in [12, 16] use the CP-ABE scheme of L. Ibraimi et al. [13] and B. Waters [17] respectively.



(a) Variation of encryption time with number of attributes in $\mathcal{T}'$

(b) Variation of decryption time with number of attributes in $\mathcal{T}'$

Figure 3: Variation of computational cost

Table 1: Comparison of PHR sharing schemes

Scheme	Access control mechanism	Attribute management	User domain	Drawbacks
L. Ibraimi et al. [12]	CP-ABE [13]	Centralized	Public & Personal	Central TA to manage all attributes in the public domain
S. Alshehri et al. [1]	CP-ABE [5]	Centralized	Public	”
C. Wang et al. [16]	CP-ABE [17]	Centralized	Public & Personal	”
M. Li et al. [14]	MA-ABE [7]	Distributed	Public & Personal	Not Scalable
Proposed Scheme	Proposed MA-CP-ABE	Distributed	Public & Personal	-

All of the above mentioned CP-ABE schemes use a centralized TA to manage and issue attributes to all users in the system. Such a centralized approach is not suitable for PHR sharing application in consideration with the associated access requirements. For instance, a PHR owner may want to share a specific PHR file with users having attributes from more than one organizational entity (ex: allowing access for any physician from hospital A or hospital B). In such a scenario, it is not realistic to assume that attributes specific for each organizational entity is issued by a centralized TA. In our solution, we adopt a fully distributed system architecture such that each entity have the capability of operating as an AA.

In contrast to the aforementioned solutions with a centralized TA, M. Li et al. [14] proposed a PHR sharing scheme supporting a distributed attribute architecture by utilizing the MA-ABE scheme of Chase and Chow [7]. This MA-ABE scheme in [7] requires a user to have at least one attribute from each of the available AAs and therefore the PHR sharing scheme in [14] is not scalable and far from being effective in practice. For instance, let us consider the following scenario. Assume that there are 100 AAs in the system and a PHR owner wants to encrypt a PHR_{obj} with only one attribute which belongs to a specific AA. However, for the proper operation of the utilized MA-ABE scheme in [14], the PHR_{obj} must be encrypted with at least one attribute from each AA (which can be achieved via dummy attributes). This applies for the decryption as well. Thus, the computation cost increases significantly with the number of AAs in the system, although the number of real attributes used for the encryption is significantly low. In our solution, we overcome this issue, and a PHR owner only needs the public keys of AAs corresponding to the attributes he uses to encrypt PHR data while decrypting user only needs secret keys corresponding to the attributes used during the encryption process (not necessary to have secret keys from all AAs in the system).

11 CONCLUSION

In this paper, we have proposed a distributed, multi-authority CP-ABE scheme (denoted as MA-CP-ABE) and thereby proposed a secure and scalable attribute based PHR sharing scheme using cloud computing which allows a PHR owner to flexibly share his private PHR data with users from both public and personal domains. Our scheme addresses the challenges brought by multiple PHR owners and users (who may come from different domains) while overcoming the practicality and scalability limitations associated with the existing PHR sharing frameworks. Our MA-CP-ABE scheme is more scalable, since it facilitates a PHR owner to encrypt private data with a set of attributes (from one or more AAs) in such a way that a user who possesses secret keys corresponding to the aforementioned attributes can successfully decrypt the data (i.e. the scheme does not require a user to have secret keys from all AAs). We have also shown that the proposed scheme is resistant against chosen plaintext attacks and attacks mounted via attribute collusion under standard security assumptions. Furthermore, the scheme can handle on-demand user revocation which helps in preventing illegitimate access via already revoked attributes. With the help of simulation results, we have shown that the proposed scheme is both efficient and realistic.

REFERENCES

[1] S. Alshehri, S. P. Radziszowski, and R. K. Raj. 2012. Secure Access for Healthcare Data in the Cloud Using Ciphertext-Policy Attribute-Based Encryption. In *Proc. of the 28th International Conference on Data Engineering Workshops*. IEEE, 143–146.

[2] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. 2012. Recommendation for Key Management – Part 1: General (Revision 3). *NIST Spec. Publ.* 800-57 (2012).

[3] M. Barua, X. Liang, R. Lu, and X. Shen. 2011. ESPAC: Enabling Security and Patient-Centric Access Control for eHealth in Cloud Computing. *International Journal of Security and Networks* 6, 2/3 (2011), 67–76.

[4] M. Barua, X. Liang, R. Lu, and X. Shen. 2011. PEACE: An Efficient and Secure Patient-Centric Access Control Scheme for eHealth Care System. In *Proc. of the IEEE Conference on Computer Communications Workshops*. IEEE, 970–975.

[5] J. Bethencourt, A. Sahai, and B. Waters. 2007. Ciphertext-Policy Attribute-Based Encryption. In *Proc. of the IEEE Symp. on Security and Privacy*. IEEE, 321–334.

[6] S. Bleikertz, M. Schunter, C. W. Probst, D. Pendarakis, and K. Eriksson. 2010. Security Audits of Multi-tier Virtual Infrastructures in Public Infrastructure Clouds. In *Proc. of the Workshop on Cloud Computing Security*. ACM, 93–102.

[7] M. Chase and S. S. M. Chow. 2009. Improving Privacy and Security in Multi-authority Attribute-based Encryption. In *Proc. of the 16th ACM Conference on Computer and Communications Security*. ACM, 121–130.

[8] D. Chen, L. Chen, X. Fan, L. He, S. Pan, and R. Hu. 2014. Securing Patient-Centric Personal Health Records Sharing System in Cloud Computing. *China Communications* 11, 13 (2014), 121–127.

[9] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren. 2005. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman & Hall/CRC.

[10] V. Goyal, O. Pandey, A. Sahai, and B. Waters. 2006. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*. ACM, 89–98.

[11] B. Grobauer, T. Walloschek, and E. Stocker. 2011. Understanding Cloud Computing Vulnerabilities. *IEEE Security & Privacy* 9, 2 (2011), 50–57.

[12] L. Ibraimi, M. Asim, and M. Petkovic. 2009. Secure Management of Personal Health Records by Applying Attribute-Based Encryption. In *Proc. of the 6th International Workshop on Wearable Micro and Nano Technologies for Personalized Health*. IEEE, 71–74.

[13] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker. 2009. Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes. In *Information Security Practice and Experience*. Springer Berlin Heidelberg, 1–12.

[14] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou. 2013. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption. *IEEE Transactions on Parallel and Distributed Systems* 24, 1 (2013), 131–143.

[15] H. S. G. Pussewalage and V. A. Oleshchuk. 2016. Privacy preserving mechanisms for enforcing security and privacy requirements in E-health solutions. *International Journal of Information Management* 36, 6, Part B (2016), 1161–1173.

[16] C. J. Wang, X. L. Xu, D. Y. Shi, and W. L. Lin. 2014. An Efficient Cloud-Based Personal Health Records System Using Attribute-Based Encryption and Anonymous Multi-receiver Identity-Based Encryption. In *Proc. 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. IEEE, 74–81.

[17] B. Waters. 2011. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In *Public Key Cryptography – PKC 2011*. Springer Berlin Heidelberg, 53–70.