Noname manuscript No.
(will be inserted by the editor)

Designing Cloud-based Electronic Health Record System with Attribute-based Encryption

Fatos Xhafa · Jingwei Li · Gansen Zhao · Jin Li · Xiaofeng Chen · Duncan S. Wong

the date of receipt and acceptance should be inserted later

Abstract With the development of cloud computing, electronic health record (EHR) system has appeared in the form of patient-centric, in which patients store their personal health records (PHRs) at a remote cloud server and selectively share them with physicians for convenient medical care. Although the newly emerged form has many advantages over traditional client-server model, it inevitably introduces patients' concerns on the privacy of their PHRs due to the fact that cloud servers are very likely to be in a different trusted domain from that of the patients.

In this paper, aiming at allowing for efficient storing and sharing PHRs and also eliminating patients' worries about PHR privacy, we design a secure cloud-based EHR system, which guarantees security and privacy of medical data stored in the cloud, relying on cryptographic primitive but not the full trust over cloud servers. Based on our proposed basic EHR system, we provide

Fatos Xhafa
Dept de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Spain
E-mail: fatos@lsi.upc.edu

Jingwei Li
College of Information Technical Science, Nankai University, China
E-mail: lijw1987@gmail.com

Gansen Zhao
School of Computer Science,South China Normal University, China
E-mail: zhaogansen@gmail.com

Jin Li
School of Computer Science, Guangzhou University, China
E-mail: jinli71@gmail.com

Xiaofeng Chen
State Key Laboratory of Integrated Service Networks (ISN), Xidian University, China
E-mail: xfchen@xidian.edu.cn

Duncan S. Wong
Department of Computer Science, City University of Hong Kong, Hong Kong
E-mail: duncan@cityu.edu.hk

several extensions including adding searchability, supporting revocation functionality and enabling efficient local decryption, which fills the gap between theoretical proposal and practical application.

Keywords Electronic health record · Attribute-based encryption · Cloud computing

1 Introduction

Electronic health record (EHR) is an evolving concept defined as a systematic collection of electronic health information about individual patients or populations. Compared with traditional paper-based health record, EHR has many basic benefits including being easily accessed and computerized, and the elimination of poor penmanship, which has historically plagued the medical chart [38][33]. Besides the basic benefits, from the perspective of system level, EHR system can also have particular functionalities which hold great promise in improving the quality of care and reducing costs at the health care system [29]. Due to the great advantages of EHR, the Health Information Technology for Economic and Clinical Health (HITECH) Act of 2009 that was signed into law as part of the "stimulus package" represents the largest US initiative to date that is designed to encourage widespread use of EHRs.

Traditionally, health care system using EHRs (without loss of generality we call it EHR system throughout this paper) was built in the client-server model. Specifically, this type of system stores data in house, requiring a server, hardware and software to be installed in the physician's office. With the recent development of cloud computing [1], it greatly desires to migrate the patients' data at the in-house servers to the cloud, which leads to the emergence of a new type of EHR system, namely the cloud-based EHR system. Compared with the traditional client-server setting, the cloud-based EHR system has many great advantages, such as requiring simpler implementation and less IT resources, reducing the cost of EHR installation, proving superior accessibility and collaboration, providing better scalability, etc.

In tandem with the great use of cloud-based EHR system, it rises concerns on the privacy over patients' personal health records (PHRs). Specifically, although it is believed that security in today's cloud-based applications is generally improved than the security in traditional systems in the sense that more resources can be devoted to solving security issues, unauthorized access to sensitive data is one of the most critical concerns from cloud-based customers (i.e., patients in our applications). Such a concern originates from the fact that cloud servers are very likely to be in a different trusted domain from that of the customers. It is thus hard to imagine that the patients would like to store their PHRs on the cloud to selectively share with physicians for medical care.

In this paper, aiming at allowing for secure storing and sharing PHRs and also making patients cancel worries about the privacy of their PHRs, we design a secure cloud-based EHR system using attribute-based encryption (ABE). The system allows the patients to share their PHRs with physicians

selectively by encrypting the data using patients' symptoms of illness without knowing the precise description of their illnesses or the departments of physicians. In our proposed system, the actual PHR is encrypted with efficient symmetric key encryption, while the symmetric key is encapsulated with ABE. This hybrid encryption paradigm will not sacrifice the efficiency of our system too much. For key encapsulation, we use the threshold ABE [34] as underlying primitive to enable the physician to access with false-tolerant, which is realistic in practice. Then, based on the proposed basic EHR system, we provide several extensions to make it allow for search, revocation and efficient local decryption, which fills the gap between our system proposal and practical application.

The rest of this paper is organized as follows. In Section 2, we review the works related to EHR and ABE. In Section 3, we provide the system model and design goals for the cloud-based EHR system. In Section 4, we design the basic EHR system. In Section 5, several extensions are added to the basic EHR system based on practical requirements. Finally conclusion is drawn in Section 6.

2 Related Work

2.1 Electronic Health Record System

The paper-based health records in use may generate an extensive paper trail. There is consequently a great interest in moving from paper-based health records to EHRs, and building new health care system with EHRs. Until now, many standards have related the regulations that an EHR system should satisfy. For example the Health Insurance Portability and Accountability Act (HIPAA) of 1996 which is the most frequently used regulation defines the privacy rules of USA health informatics. We suggest the readers refer to [10] for other detailed regulations and standards.

Security and privacy play an important role in today's EHR system. Aiming at allowing physicians to access patients' health data without disclosing patients' personal data, the pseudo anonymity technique has been applied in several works [31][32][8]. Of these, [8] used an approach for reversible pseudonym generation; [32] proposed the possibility of sharing pseudonyms based on Shamir secret sharing [35]. Besides anonymize identity, many works (e.g., [2][9][13][14][15] to list a few) suggested that EHR data should also be encrypted in order to increase security.

Concerning on the deployment of EHR system in cloud, Li et al. [28] addressed the problem of authorized private keyword searches on encrypted PHRs in cloud computing environment, and presented a scalable and fine-grained authorization framework for this purpose. Haas et al. [13] and Zhang et al. [40] considered that patients should not trust that the cloud service provider cannot access their EHR data, particularly when the cloud service provider is unrelated to patient or health institutions, and proposed crypto-

graphic solutions. The work most related to ours is from Narayan et al. [30] who proposed the use of ABE to ensure that the cloud service provider cannot see (or copy) EHR data. Compared with our work, the previous work [30] lacks of the consideration of the fault-tolerance and reducing local computation in decryption. In addition, Narayan et al.'s work [30] uses a public-key searchable encryption [4] which involves computationally intensive operations such as bilinear mappings and is not scalable. In this work, our searchability is just based on lightweight pattern match and suitable for the EHR application consisting of large number of PHRs.

2.2 Attribute-based Encryption

The notion of ABE, which was introduced as fuzzy identity-based encryption in [34], was firstly dealt with by Goyal et al. [11]. Two different and complementary notions of ABE were defined as key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). A construction of KP-ABE was provided in the same paper [11], while the first CP-APE construction supporting tree-based access structure in generic group model is presented by Bethencourt et al. [3].

Subsequently, a number of variants of ABE schemes have been proposed since its introduction. They range from extending its functionality to proposing schemes with stronger security proofs. Such as ABE schemes supporting for any kinds of access structures [7][37], ABE with multi-authorities [5][6][18], full secure ABE [16][17][20], unbounded ABE [19], etc.

Recently, a novel outsourcing paradigm for ABE was provided for reduce local computation [41][12][22] [23]. The common idea in these work is to utilize secure outsourcing technique to delegate the overhead computation during encryption/decryption/key-issuing to third party such that the local computation is minimized. In this paper we follow this outsourcing paradigm to realize efficient local decryption.

3 Problem Statement

3.1 System Model

In this paper, we consider a cloud computing environment which hosts the PHR service. Specifically, as shown in Fig. 1 there are four entities involved in this system.

- *Patient.* It is an entity which creates its PHRs, and stores them at the cloud server such that physicians equipped with professional capabilities are able to access them.
- *Cloud Server.* It is an entity which is responsible for storing patients' (encrypted) PHRs in a database and performing searches for the physicians.

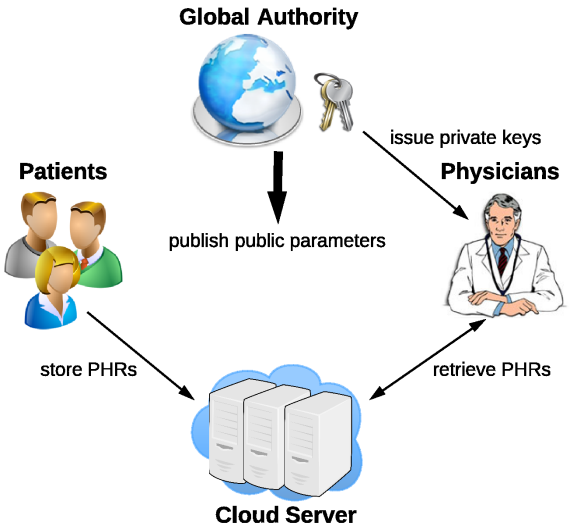

Fig. 1 Architecture for EHR System

- *Global Authority.* It is an entity which is responsible for key management. Specifically, it generates keys for physicians of the system, and publishes public parameters needed by cryptographic operations.
- *Physicians.* It is an entity which can submit query to retrieve PHRs stored at cloud server. More precisely, in this paper we consider the physicians to be from the public domain, that is they are usually not personally known by the patients. The physicians are able to obtain their private keys due to their professional responsibilities, and need to access the PHRs for providing medical care.

Based on the system above, we then provide an overview of our proposed EHR system.

- **System Setup**. Public parameter and master secret key are initialized for the system, and the global authority keeps the master secret key and publishes the public parameters outside.
- **Physician Authorization**. When a physician wants to join the system, he/she has to apply for his/her private key at the authority according to his/her professional responsibilities.
- **PHR Storage**. When a patient wants to create and share his/her PHR at the cloud server, he/she encrypts and sends the PHR to cloud server.
- **PHR Access**. When a physician wants to access a PHR, he/she downloads ciphertext from the cloud server and decrypts it.

3.2 Adversary Model and Design Goals

In this paper, we assume the global authority is fully trusted, and consider a "honest-but-curious" server which has been widely adopted in many existing

works [36][24][25]. Specifically, the cloud server will be "curious" for learning the underlying meanings of the PHRs, but still honestly follow our proposed protocols. Also, the cloud server could collude with some physicians to help them derive additional information about patients' PHRs beyond physicians' accessing scope.

Then, under the adversary model clarified above, we attempt to address the problem of building an efficient fine-grained access control EHR system in the cloud environment. Specifically, we allow the patient to enforce an access policy on EHR according his/her disease which precisely designates the group of physicians allowed to access the EHR. Also the cloud server is prevented from learning the underlying meaning of the EHRs even if it colludes with physicians not in the scope of the target EHRs. Besides this, we require that all the goals should be achieved efficiently in the sense that the EHR system is scalable.

4 Basic EHR System

In this section, we will firstly introduce the background of ABE, and then describe our design specification adapting to practical needs. Finally, the basic EHR system as well as its security analysis are provided.

4.1 Attribute-based Encryption

Attribute-based encryption has been widely applied to impose fine-grained access control on encrypted data recently. There are two kinds of ABE having been proposed: KP-ABE and CP-ABE. In KP-ABE, the access policy is assigned in private key, whereas, in CP-ABE, it is specified in ciphertext. Without loss of generality, we are able to denote $(I_{\text{enc}}, I_{\text{key}})$ as the input to encryption and key generation of ABE. Accordingly, in CP-ABE scheme, $(I_{\text{enc}}, I_{\text{key}}) = (\mathbb{A}, \omega)$ while that is $(\omega, \mathbb{A})$ in KP-ABE, where ω and $\mathbb{A}$ are attribute set and access structure, respectively. Then, an ABE scheme is consisted of four algorithms below.

- $\mathsf{Setup}(\lambda)$: The setup algorithm takes as input – a security parameter λ. It outputs the public key pk and the master secret key msk.
- $\mathsf{KeyGen}(I_{\text{key}}, msk)$: The key extraction algorithm takes as input – an access structure (resp. attribute set) I_{key} and the master secret key msk. It outputs the user's private key sk.
- $\mathsf{Encrypt}(m, I_{\text{enc}})$: The encryption algorithm takes as input – a message m and the attribute set (resp. access structure) I_{enc}. It outputs the ciphertext ct.
- $\mathsf{Decrypt}(ct, sk)$: The decryption algorithm takes as input – a ciphertext ct which was assumed to be encrypted under the attribute set (resp. access structure) I_{enc} and the private key sk for access structure (resp. attribute

Name	Age	Sex	Region	Possible Illness	Contact Information
Alice	65	female	Boston	heart disease	XXX-XXX-XXXX
Bob	30	male	New York		XXX-XXX-XXXX
Lisa	45	female	Washington DC		XXX-XXX-XXXX

Table 1 An Example of PHR

set) I_{key}. It outputs the message m if $\gamma(I_{\text{key}}, I_{\text{enc}}) = 1$ and the error symbol $\perp$ otherwise, where the predicate γ is predefined.

4.2 Design Specification

4.2.1 PHR Structure

In the EHR system, a patient's PHR describes detailed personal information of him/her. Table. 1 illustrates an example of three personal information in EHR for Alice. Note that the entry of possible illness can be let blank if the patient does not know it properly.

In practice, a patient may want to share his/her record with a physician familiar with his/her symptoms, but do not want to allow others to read anything about personal information. Therefore, in our EHR system, we require that PHR will be encrypted under the symptoms of patient's disease, which potentially specifies the underlying physicians allowed to access patient's personal information.

4.2.2 Physician Attributes

The physicians' attributes, consisting of many kinds of symptoms of the diseases that the physician is professional in, are obtained from the global authority. For example, if a physician is professional in heart disease and has obtained the qualification certificate for this disease, he/she can make an authentication at authority with the certificate. The authority then assigns him with a private key for the corresponding symptoms including heartache, shortness of breath, dizziness and so on. Using this private key, the physician is able to later access the PHRs within his/her scope.

4.2.3 Access Policy

Suppose all the symptoms of diseases constitute a universe $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ where u_i indicates a symptom of disease such as heartache or dizziness (we then will not distinguish the term of symptom and attribute used in the rest of this paper). Then, each patient's disease and the professional abilities of physician can be described as subsets of symptoms, that is $\hat{\omega} \subseteq \mathcal{U}$ and $\omega \subseteq \mathcal{U}$ respectively.

Next, we specify the policy for accessing PHRs. Recall that we use "symptoms" to describe both patient's disease and physician's professional abilities. Intuitively, if an identical set of symptoms is owned by a patient and a physician, it has a significant probability that the physician specializes in the patient's disease. It is advisable to allow such a physician to access this patient's personal information and provide a rigorous treatment on his/her illness.

Besides the "exact match", we emphasize that we should add error-tolerance property in our EHR system to allow for a private key (derived from a measurement of a disease) to decrypt a PHR encrypted with a slightly different measurement of the same disease. This is because the symptom-based measurement for disease is noisy in both folds. On one side, we cannot always require that the same disease is expressed in an identical set of symptoms. For example the disease pneumonia appears in somebody in the symptoms of headache and dry cough, but for others it may present with whole body aches and having a fever. On the other side, an identical set of symptoms may be derived from different disease. For example, a simple symptom of cough could be the result of catching a cold, but it is also attributed to infecting pneumonia.

Aiming at helping patient receive medical treatment as rigorous as possible, while preventing others not professional in disease accessing the PHRs, we use a threshold function shown below to measure whether a physician with ω can access the PHR encrypted under $\hat{\omega}$ or not (1 indicates access is allowed while 0 otherwise), where d is the fault tolerance allowed.

$$\gamma(\hat{\omega}, \omega) = \begin{cases} 1 & |\omega \cap \hat{\omega}| \geq d \\ 0 & |\omega \cap \hat{\omega}| < d \end{cases}$$

4.3 System Description

Based on the design specification elaborated above, we then provide our basic system in detail as follows. Note that in our system, we utilize the threshold ABE [34] shown in Fig. 2 as underlying primitive.

System Setup. Suppose the attributes in universe $\mathcal{U}$ are taken from $\mathbb{Z}_q$. Choose a security parameter λ and run the procedure $\mathsf{Setup}(\lambda)$ of underlying ABE primitive to obtain the public key pk and the master secret key msk. The public key is then published, while the master key is kept by global authority as a secret.

Physician Authorization. Assuming a new physician wants to join the EHR system, he/she needs to be issued a private key for later accessing patients' PHRs. In concrete, the physician provides the authority with his/her qualification certificate for proving his/her specialized in a certain set of diseases. The global authority then computes a set of symptoms ω which he/she is able to professionally deal with. Finally the global authority runs $\mathsf{KeyGen}(\omega, msk)$ and assigns the corresponding private key sk to this physician.

$\mathsf{Setup}(1^\lambda)$: Select a generator $g \in_R \mathbb{G}$ and an integer $x \in_R \mathbb{Z}_q$, and set $g_1 = g^x$. Then, pick elements $g_2, h_1, \ldots, h_n \in_R \mathbb{G}$. Finally, output the public key $pk = (g, g_1, g_2, d, h_1, \ldots, h_n)$ and the master secret key $msk = x$.

$\mathsf{KeyGen}(\omega, msk)$: Upon receiving a private key request on ω, randomly pick a $(d-1)$-degree polynomial $poly(\cdot)$ with $poly(0) = msk$. Then, for each attribute $i \in \omega$, compute $d_{i0} = g_2^{poly(i)} \cdot (g_1 h_i)^{r_i}$ and $d_{i1} = g^{r_i}$, where $r_i \in_R \mathbb{Z}_q$. Finally return $sk = (\{(d_{i0}, d_{i1})\}_{i \in \omega})$.

$\mathsf{Encrypt}(\hat{\omega}, m)$: To encrypt a message $m \in \mathbb{G}$ under $\hat{\omega}$, select an integer $s \in_R \mathbb{Z}_q$. Then, compute $c_0 = m \cdot e(g_1, g_2)^s$, $c_1 = g^s$ and $e_i = (g_1 h_i)^s$ for $i \in \hat{\omega}$. Finally, publish the ciphertext as $ct = (\hat{\omega}, c_0, c_1, \{e_i\}_{i \in \hat{\omega}})$.

$\mathsf{Decrypt}(sk, ct)$: Suppose that a ciphertext ct is encrypted under an attribute set $\hat{\omega}$ and physician is assigned with a private key sk for attribute set ω, which satisfies the restriction that $\gamma(\hat{\omega}, \omega) = 1$. Then, the decryption proceeds as follows. Firstly, an arbitrary d-element subset set $S \subseteq \hat{\omega} \cap \omega$ is selected. Then, the ciphertext is decrypted as $m = \frac{\prod_{i \in S} e(c_1, d_{i0})^{\Delta_{i,S}(0)}}{\prod_{i \in S} e(d_{i1}, e_i)^{\Delta_{i,S}(0)}}$, where $\Delta_{i,S}(0) = \prod_{j \in S, j \neq i} \frac{-j}{i-j}$ is the Lagrange polynomial at zero point.

Fig. 2 Underlying ABE Primitive

PHR Storage. Whenever a patient suffers from some illness, and wants to upload his/her PHR to the cloud servers to call for medical help from physicians. He/She firstly summarize his/her symptoms represented by $\hat{\omega}$ according to the published parameters. For example, if a patient suffers from heart disease, he/she may specify the symptoms "heartache", "shortness of breath" and "dizziness" in $\hat{\omega}$. Note that all these symptoms are expressed in $\mathbb{Z}_q$. Then, the patient randomly chooses a symmetric key k from the key space and encrypts the PHR f with k using standard symmetric key algorithm such as AES. Later on, he/she runs the algorithm $\mathsf{Encrypt}(\hat{\omega}, k)$ and obtains the ciphertext ct which is the encryption of the symmetric key k with respect to the access policy $\hat{\omega}$. Finally, the patient uploads the encrypted PHR ct as well as the symptoms $\hat{\omega}$ to the cloud server.

PHR Access. Suppose a physician wants to access and retrieve PHRs of his/her interests, he/she proceeds as follows: He/She firstly submits his/her professional abilities expressed as a set of symptoms ω to the cloud server. The cloud server then calculates and returns all the records $(\hat{\omega}, ct)$ satisfying $|\hat{\omega} \cap \omega| \geq d$. The physician finally runs $\mathsf{Decrypt}(sk, ct)$ and symmetric key decryption to obtain the plain PHRs of patients, with which he/she can contact the patients and provide medical treatment in time.

4.4 Security Analysis

Recall that two kinds of adversaries are considered, that is, 1) security against external attackers (i.e., cloud server): The security definition for this kind of attacker is captured by confidentiality. It means that, given access to a set of encrypted data, the external attackers, including the server, are not able to

learn any partial information about the underlying PHRs; 2) security against internal attackers (i.e., some "curious" physicians colluding with the cloud server) : The security definition that is used to capture the internal attackers is accessing PHRs not intended for them. It means that even if a set of "curious" physicians collaborate together they still cannot access the encrypted PHRs legal for none of them. Next, we analyze the security of our system in more details.

Security Against External Adversaries. We need to prove that, given access to a set of encrypted data, the external attackers are not able to learn any partial information about the underlying files. Recall that the main external adversary considered in this paper is the cloud server. For cloud servers, they do not have attribute private keys on any attribute set, which means that they cannot get any information of the secret key k used in the symmetric encryption. Therefore, they are not able to decrypt and get any information of the underlying PHRs if the symmetric encryption is secure.

Security Against Internal Adversaries. To protect the privacy of PHRs from internal adversaries, an ABE scheme is utilized. Specifically, to guarantee security, it is required that even if a set of "curious" physicians (not in the sharing scope of the PHRs) collude together they could not learn anything useful from the encrypted PHRs. This coincides with the security goal of message indistinguishability in ABE, which demands that user is unable to decrypt a ciphertext if his/her attributes does not match the access policy. Based on this analysis, the key point of the security against internal adversaries is the underlying ABE primitive. Actually, this is naturally true due to the work [34].

Based on the analysis above, we draw the conclusion that if the underlying primitives ABE and symmetric key encryption are secure, our proposed EHR system is secure in terms of the adversary model we consider in Section 3.2.

5 Extensions

In this section, based on our basic EHR system, we provide several extensions to facilitate its practical utilization.

5.1 Adding Searchability

Note that one efficiency bottleneck of our basic system is that it does not support searchability at the cloud server. Specifically, upon receiving physician's ω, the cloud server has to scan all the stored PHRs, and find out the PHRs satisfying that the number of overlapped symptoms (i.e., the symptoms associated with found PHR and physician's submitted symptoms) is beyond a predefined threshold value. In this case, the efficiency cost at cloud server for

a single time search is $O(l)$ where l is the number of PHRs having been stored at the cloud server.

Aiming at improving the searching efficiency, we attempt to utilize the fuzzy keyword search technique [27] and build a trie-based index for facilitate the computation at cloud server. The challenge for this purpose is filling the gap between fuzzy keyword search and the set overlaps decision (i.e., decide whether the number of overlaps of two sets is beyond a specified value or not) in this paper. More precisely, unlike the fuzzy keyword search, set has the property that it does not have order and each member appears not more than once, which results that we cannot impose the edit distance to determine the overlaps of set.

In order to transform set overlaps decision to fuzzy keyword search, we number each symptoms in universe from 1 to n, which allows us to express each set as an n-length binary string: if a symptom belongs to a set, the corresponding bit of this symptom is set to 1; otherwise set to 0. After this operation, the problem of set overlap decision is reduced to examine whether two binary strings both have 1 in at least d bits, where d is the predefined threshold value. It is slightly next to the fuzzy keyword search, but there still exists a challenge that we cannot trivially use the edit distance here because 0 cannot take account. That means the fact that two sets ω and $\hat{\omega}$ simultaneously do not have an identical item should not contribute for increasing the number of overlaps.

To tackle this challenge, we use "template match". More precisely, we use wildcard to generate a template set Ω of ω, and use Ω for the search at cloud server. For example, suppose the threshold $d = 3$ and the numbered universe $\mathcal{U} = \{1, 2, \ldots, 10\}$. A set $\omega = \{1, 3, 7, 9\}$ can be expressed as a 10-length binary string "1010001010". It is clear that the sets $\hat{\omega}$ consisting any three items in ω will match it (it means the physician with ω can successfully decrypt the ciphertext encrypted under $\hat{\omega}$). We build the template set $\hat{\Omega} = \{1?1???1???, 1?1?????1?, ??1???1?1?, 1?????1?1?\}$ and use $\hat{\Omega}$ to perform search with a trie-based index like [27], where ? is a wildcard symbol for matching arbitrary bit (0 or 1). The improved stage of PHR storage and access is described as follows.

PHR Storage. It is identical to the homonymic stage in Section 4.3 except the cloud server builds an trie-based index for storing the symptom set for each PHR. Specifically, upon receiving $(\hat{\omega}, ct)$, the cloud server parses $\hat{\omega}$ into a n-length string $\alpha_1\alpha_2 \ldots \alpha_v$ following the principle elaborated above where $\alpha_i \in \{0, 1\}^{n/v}$, and then runs Algorithm 1 to adaptively add this string into the trie index $\mathcal{T}$. Note that for reducing storage cost, we combine v bits together stored at each node in trie.

PHR Access. Suppose a physician wants to access and retrieve PHRs of his/her interests, he/she proceeds as follows: He/She firstly parses his/her professional abilities expressed as a set ω of symptoms into the template set $\hat{\Omega}$ and then submits $\hat{\Omega}$ to the cloud server. The cloud server runs an algorithm of SearchTree to find out all the encrypted PHRs, of which the associated

Algorithm 1: Trie Update

```
Procedure TrieUpdate(T, α1α2...αv, P)
Input: trie T
       binary string α1α2...αv
       pointer P pointing to the encrypted PHR
Output: updated trie T
begin
    set current node as root of T ;
    for i ← 1 to v do
        if there exists a child node of current node containing αi then
            set this child node as β;
        else
            create a new child node β containing αi;
        end
        set β as current node;
    end
    append P with current node;
    return T;
end
```

symptoms set ω matches $\hat{\Omega}$. The key idea behind this algorithm is that all sets may have overlaps if their corresponding strings share a common prefix. Then, then all the sets matching $\hat{\Omega}$ can be found with a depth-first search of the trie-based index. Since the Search algorithm used is similar to that in [26], we omit it here and suggest the readers to refer [26] for formal description of this algorithm. After utilizing the search algorithm based on trie and collecting all the $\hat{\omega}$ satisfying $|\hat{\omega} \cap \omega| \geq d$, the cloud server continues to read its corresponding encrypted PHR ct and returns the result $\{(\hat{\omega}, ct)\}_{|\hat{\omega} \cap \omega| \geq d}$ back to physician. Then the following workflow is the same to that in Section 4.3.

5.2 Physician Revocation

Besides the searchability, another functionality needed to be supported by our EHR system is physician revocation. For example, if the certificate of physician authenticated has been out of date, it desires an efficient and secure way for global authority to revoke the decryptabilities of these physicians.

For this purpose, we will utilize the idea of "updating attribute public key" [39] to achieve physician revocation. Generally, this idea works by updating the public keys of the attributes associated with revoked physicians, such that they cannot utilize their private keys generated from the previous attribute public keys to perform decryption. We then provide the details for physician revocation in our setting as follows.

Physician Revocation. Whenever the global authority finds that a physician it manages has an overdue certificate, then it attempts to revoke the decryptability of this physician. To achieve this goal, the following steps are proceeded.

- The global authority firstly defines a minimal set of attributes $\mathcal{S}$ such that all the attributes of the physicians to be revoked exist in $\mathcal{S}$. Next, the it attempts to update all the attribute public keys corresponding to the attributes in $\mathcal{S}$. This work is done by running the $\mathsf{PubUpdate}$ algorithm of underlying ABE primitive shown in Fig. 3 to obtain pk_{new}, rk and msk_{new}[1]. We emphasize that the ABE primitive used here is a bit different from which we elaborate in Fig. 2. Specifically, 1) The attribute public keys $h_1, h_2, \ldots, h_n$ are required to be selected as $h_1 = g^{t_1}, h_2 = g^{t_2}, \ldots, h_n = g^{t_n}$ where $t_1, t_2, \ldots, t_n \in_R \mathbb{Z}_q$. Since t_i are random values, this change will not affect the randomness of h_i; 2) The master secret key used here is described as $msk = (x, t_1, t_2, \ldots, t_n)$. The authority continues to publish pk_{new} as a new version of public key and sending the re-encryption key rk and $\mathcal{S}$ to cloud server.
- The cloud server then re-encrypts the "affected" ciphertext (i.e., are encrypted with attributes involving in $\mathcal{S}$) with the re-encryption key. Specifically, for each "affected" ciphertext ct_{old}, it runs $\mathsf{ReEncrypt}(ct_{\text{old}}, rk)$ to get the updated ciphertext ct_{new}.
- Meanwhile, the the other "affected" physicians (i.e., share attributes with the revoked physicians) should request for the newest version of private keys. Specifically, they take their private keys in old version to authority for key-update. The global authority then re-issues private keys to them through running KeyGen with the newest version of public keys.

Remark. During the procedure of revocation, one of the efficiency bottleneck is at the authority. Specifically, it has to re-issue private keys to all the other "affected" physicians, which introduces an overhead computation at authority. This issue can be fixed by using outsourced key-issuing protocol and delegating the attribute-related computation to other cloud service provider [21]. With this technique [21], during a single re-issuing of private key, the authority just needs to nearly three single-based modular exponentiations at local, and its complexity is minimized.

5.3 Reducing Physician's Local Computations

The final observation is the computation at physician sides. Specifically, as the popularity of mobile computing, typically in practice the physician may use mobile device to query the cloud server and be returned with a number of encrypted PHRs matching him/her, which requires a large computation for complete decryption. Even if today's mobile device has theoretically provided computing resources not weaker than personal computers, it will consume a lot battery power, which still does not suggest imposing large overhead computation at mobile devices. In this case, one choice is to temporarily store the encrypted PHRs at mobile devices and transfer them to computer for

[1] For simplicity, we just re-publish the attribute public key component in $\mathsf{PubUpdate}$ because the other components are not be changed.

$\mathsf{PubUpdate}(pk_{\text{old}}, \mathcal{S}, msk_{\text{old}})$: To update all the attribute public keys corresponding to the attributes in $\mathcal{S}$, the authority uses the old master secret key $msk_{\text{old}} = (x, \{t_i\}_{i\in\mathcal{U}})$ to proceed as follows. For each $i \in \mathcal{U}$, it computes $h'_i = h_i^{t'_i/t_i}$ where $t'_i \in_R \mathbb{Z}_q$ if $i \in \mathcal{S}$, and sets $h'_i = h_i$ and $t'_i = t_i$ if $i \notin \mathcal{S}$. Finally output the updated public keys $pk_{\text{new}} = (\{h'_i\}_{i\in\mathcal{U}})$, re-encryption key $rk = (\{\frac{x+t'_i}{x+t_i}\}_{i\in\mathcal{U}})$ and keep the adaptively updated master secret key $msk_{\text{new}} = (x, \{t'_i\}_{i\in\mathcal{U}})$ local.

$\mathsf{ReEncrypt}(ct, rk)$: In order to transform a ciphertext encrypted with an old version of public keys to that under the current public keys, the cloud server computes $e'_i = e_i^{\frac{x+t'_i}{x+t_i}}$ for all the $i \in \hat{\omega}$, $c'_1 = c_1$ and $c'_0 = c_0$. Finally the ciphertext is updated as $ct_{\text{new}} = (c'_0, c'_1, \{e'_i\}_{i\in\hat{\omega}})$.

Fig. 3 Algorithms for Physician Revocation

decryption when come back home. But this will result in delay for medical treatment, possibly affecting the life risk of patients. Therefore, it desires a secure way for physicians to efficiently and immediately decrypt a large number of encrypted PHRs even if they are with resource-constrained devices like mobile devices. As with the remark in the previous, we attempt to resolve this issue with outsourcing computation.

Suppose there exists an additional entity namely decryption cloud service provider (D-CSP) which works for facilitating physicians with resource-constrained devices do a large number of decryptions. Actually, such type of service provider is common in today's cloud computing such as Amazon EC2.

Generally, there are two alternatives for secure delegating decryption: one is based on blinding private keys [12] and the other is to introduce a trivial attribute and use it to control local decryption [21]. In this paper, we just focus the former technique and attempt to extend it to our setting for facilitate physicians' decryption.

In the outsourced decryption workflow, at a high level, after authorization, physicians with attributes Ω can compute his/her blinded private key $\widetilde{sk}$ with a randomly picked blinding factor t, and deliver $\widetilde{sk}$ to the newly introduced D-CSP to be stored. Then, after receiving the ciphertext from cloud server, the physician directly forwards the result to D-CSP which uses his/her corresponding $\widetilde{sk}$ to compute the partially decrypted ciphertext (compared with the final plaintext in our setting, it is blinded by the factor t). The physicians then completely this type of ciphertext with his/her t. Then final improved stages for physician authorization and PHRs access are described as follows.

Physician Authorization. This stage works the same as that elaborated in Section 4.3 except a key-blinding procedure is required. Specifically, after receiving the sk from the global authority, the physician continues to run the algorithm shown in Fig. 4 to exponentially blind his/her private key with t, and forwards and stores $\widetilde{sk}$ at D-CSP for future decryption. The final private key includes sk issued by authority and the blinding factor t, and we will abuse it as sk as well in our following elaboration.

$\mathsf{KeyBlind}(sk)$: Upon receiving a private key $sk = (\{d_{i0}, d_{i1}\}_{i\in\omega})$ generated from the authority, physician picks $t \in_R \mathbb{Z}_q$ and computes $d'_{i0} = d^t_{i0}$ and $d'_{i1} = d^t_{i1}$ for each $i \in \omega$. Finally output $\widetilde{sk} = (\{d'_{i0}, d'_{i1}\}_{i\in\omega})$ and the redefined private key $sk = (\{d_{i0}, d_{i1}\}_{i\in\omega}, t)$

$\mathsf{Decrypt}_{\mathrm{out}}(\widetilde{sk}, ct)$: Suppose that a ciphertext $ct = (c_0, c_1, \{e_i\}_{i\in\hat{\omega}})$ is encrypted under an attribute set $\hat{\omega}$ and D-CSP is assigned with a blinded private key $\widetilde{sk}$ for attribute set ω, which satisfies the restriction that $\gamma(\hat{\omega}, \omega) = 1$. Then, D-CSP proceeds as follows. Firstly, an arbitrary d-element subset set $S \subseteq \hat{\omega} \cap \omega$ is selected. Then, the partial decryption is computed as $c'_0 = \frac{\prod_{i\in S} e(c_1, d'_{i0})^{\Delta_{i,S}(0)}}{\prod_{i\in S} e(d'_{i1}, e_i)^{\Delta_{i,S}(0)}} = e(g_1, g_2)^{st}$. Finally D-CSP returns $ct' = (c_0, c'_0)$.

$\mathsf{Decrypt}(sk, ct')$: . Upon receiving the partially decrypted ciphertext ct' from D-CSP, physician computes $c_0/(c'_0)^{1/t} = m$.

Fig. 4 Algorithms for Reducing Computation for Physicians

PHRsAccess. Instead of directly decrypting ciphertext by themselves, the physicians firstly forward the searched encrypted PHRs to D-CSP (for simplicity our description starts from receiving the results and we omit the workflow of search which is detailedly elaborated in the same stage shown in Section 5.1). The D-CSP reads the corresponding stored $\widetilde{sk}$ for this physician and runs the outsourced decryption algorithm $\mathsf{Decrypt}_{\mathrm{out}}$ shown in Fig. 4 for each forwarded ciphertext. After performing partial decryption, D-CSP then sends all the partially decrypted ciphertexts to the physician, which runs the local decryption algorithm $\mathsf{Decrypt}$ on each of them to obtain symmetric key. With these keys, physician finally does the symmetric decryption and get the plain PHRs.

6 Conclusion

In this paper, aiming at allowing for efficient storing and sharing PHRs in today's cloud computing, we design a secure cloud-based EHR system using ABE. Our system allows the patients to selectively share their PHRs with physicians by performing encryption under their current symptoms but without knowing the precise description of their illnesses or the departments of physicians needed in medical treatment. Furthermore, in order to fill the gap between theoretical proposal and practical application, we provide three extensions including adding searchability, supporting physician revocation functionality, allowing for efficient local decryption, on our basic EHR system. We believe our final system meets the practical requirements in today's EHR service.

Acknowledgement

This work is supported by National Natural Science Foundation of China (Grant No.61100224, No.61272455), Guangdong Natural Science Foundation

(No.S2013010013671), Guangzhou Research Infrastructure Development Fund (No. 201222412), Guangzhou Zhujiang Science and Technology Future Fellow Fund (No. 2011J2200089), and the MOE-China Mobile Research Fund (No. MCM20121051).

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D.: A View of Cloud Computing. Communications of the ... **53**(4), 50–58 (2010)
2. Benaloh, J., Chase, M., Horvitz, E., Lauter, K.: Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records. In: CCSW '09 Proceedings of the 2009 ACM workshop on Cloud computing security, pp. 103–114 (2009)
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Computer Society (2007)
4. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search. In: Advances in Cryptology-Eurocrypt 2004, pp. 506–522. Springer (2004)
5. Chase, M.: Multi-Authority Attribute Based Encryption. Theory of Cryptography pp. 515–534 (2007)
6. Chase, M., Chow, S.: Improving Privacy and Security in Multi-Authority Attribute-Based Encryption. In: CCS '09 Proceedings of the 16th ACM conference on Computer and communications security, pp. 121–130 (2009)
7. Cheung, L., Newport, C.: Provably Secure Ciphertext Policy ABE. In: CCS '07: Proceedings of the 14th ACM conference on Computer and communications security, pp. 456–465. ACM Request Permissions (2007)
8. Elger, B.S., Iavindrasana, J., Lo Iacono, L., Müller, H., Roduit, N., Summers, P., Wright, J.: Strategies for Health Data Exchange for Secondary, Cross-Institutional Clinical Research. Computer Methods and Programs in Biomedicine **99**(3), 22–22 (2010)
9. Farzandipour, M.M., Sadoughi, F.F., Ahmadi, M.M., Karimi, I.I.: Security Requirements and Solutions in Electronic Health Records: Lessons Learned From a Comparative Study. Journal of Medical Systems **34**(4), 629–642 (2010)
10. Fernández-Alemán, J.L., Señor, I.C., Lozoya, P.Á.O., Toval, A.: Security and Privacy in Electronic Health Records: a Systematic Literature Review. Journal of biomedical informatics **46**(3), 541–562 (2013)
11. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In: Proceedings of the 13th ACM conference on Computer and communications security, pp. 89–98. ACM (2006)
12. Green, M., Hohenberger, S., Waters, B.: Outsourcing the Decryption of ABE Ciphertexts. In: SEC'11: Proceedings of the 20th USENIX conference on Security, pp. 34–49. USENIX Association (2011)
13. Haas, S., Wohlgemuth, S., Echizen, I., Sonehara, N.: Aspects of Privacy for Electronic Health Records. international journal of Medical Informatics **80**(2), e26–31 (2011)
14. Hu, Chen, Hou: A Hybrid Public Key Infrastructure Solution (HPKI) for HIPAA Privacy/Security Regulations. Computer Standards & Interfaces **32**(5-6), 7–7 (2010)
15. Lee, W.B.W., Lee, C.D.C.: A Cryptographic Key Management Solution for HIPAA Privacy/Security Regulations. Information Technology in Biomedicine, IEEE Transactions on **12**(1), 34–41 (2007)
16. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: EUROCRYPT'10: Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques, pp. 62–91. Springer-Verlag (2010)
17. Lewko, A., Waters, B.: New Techniques for Dual System Encryption and Fully Secure HIBE with Short Ciphertexts. Theory of Cryptography pp. 455–479 (2010)

18. Lewko, A., Waters, B.: Decentralizing Attribute-Based Encryption. EUROCRYPT'11 Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology pp. 568–588 (2011)
19. Lewko, A., Waters, B.: Unbounded HIBE and Attribute-Based Encryption. EUROCRYPT'11 Proceedings of the 30th Annual international conference on Theory and applications of cryptographic techniques: advances in cryptology pp. 547–567 (2011)
20. Lewko, A., Waters, B.: New Proof Methods for Attribute-Based Encryption: Achieving Full Security Through Selective Techniques. Advances in Cryptology–CRYPTO 2012 pp. 180–198 (2012)
21. Li, J., Chen, X., Li, J., Jia, C., Ma, J., Lou, W.: Fine-Grained Access Control System Based on Outsourced Attribute-Based Encryption. In: Computer Security–ESORICS 2013, pp. 592–609 (2013)
22. Li, J., Huang, X., Li, J., Chen, X., Xiang, Y.: Securely Outsourcing Attribute-based Encryption with Checkability. IEEE Transactions on Parallel and Distributed Systems p. http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.271 (2013)
23. Li, J., Jia, C., Li, J., Chen, X.: Outsourcing Encryption of Attribute-Based Encryption with Mapreduce. ICICS'12: Proceedings of the 14th international conference on Information and Communications Security pp. 191–201 (2012)
24. Li, J., Li, J., Chen, X., Liu, Z., Jia, C.: Privacy-preserving data utilization in hybrid clouds. Future Generation Computer Systems (2013)
25. Li, J., Li, J., Liu, Z., Jia, C.: Enabling efficient and secure data sharing in cloud computing. Concurrency and Computation: Practice and Experience pp. n/a–n/a (2013)
26. Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W.: Enabling Efficient Fuzzy Keyword Search over Encrypted Data in Cloud Computing. IACR Cryptology ePrint Archive pp. 1–16 (2009)
27. Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W.: Fuzzy Keyword Search Over Encrypted Data in Cloud Computing. In: INFOCOM'10: Proceedings of the 29th conference on Information communications, pp. 1–5. IEEE Press (2010)
28. Li, M., Yu, S., Cao, N., Lou, W.: Authorized Private Keyword Search over Encrypted Data in Cloud Computing. In: ICDCS '11: Proceedings of the 2011 31st International Conference on Distributed Computing Systems, pp. 383–392. IEEE Computer Society (2011)
29. Menachemi, N., Collum, T.H.: Benefits and Drawbacks of Electronic Health Record Systems. Risk management and healthcare (4), 47–55 (2011)
30. Narayan, S., Gagné, M., Safavi-Naini, R.: Privacy Preserving EHR System Using Attribute-Based Infrastructure. In: CCSW '10 Proceedings of the 2010 ACM workshop on Cloud computing security workshop, pp. 47–52. ACM Request Permissions (2010)
31. Neubauer, T., Heurix, J.: A Methodology for the Pseudonymization of Medical Data. International Journal of Medical Informatics **80**(3) (2011)
32. Riedl, B., Grascher, V., Neubauer, T.: Applying a Threshold Scheme to the Pseudonymization of Health Data. In: PRDC '07: Proceedings of the 13th Pacific Rim International Symposium on Dependable Computing, pp. 397–400. IEEE Computer Society (2007)
33. Rodríguez-Vera, F.J., Marin, Y., Sanchez, A., Borrachero, C.: Illegible Handwriting in Medical Records. Journal of the Royal Medical Records **95**(11), 545–546 (2002)
34. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: EUROCRYPT'05: Proceedings of the 24th annual international conference on Theory and Applications of Cryptographic Techniques, pp. 457–473. Springer-Verlag (2005)
35. Shamir, A.: How to Share a Secret. Communications of the ACM **22**(11), 612–613 (1979)
36. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on, pp. 44–55 (2000)
37. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: an Expressive, Efficient, and Provably Secure Realization. In: Public Key Cryptography–PKC 2011, pp. 53–70. Springer (2011)

38. Winslow, E.H.E., Nestor, V.A.V., Davidoff, S.K.S., Thompson, P.G.P., Borum, J.C.J.: Legibility and Completeness of Physicians' Handwritten Medication Orders. Heart & Lung: The Journal of Acute and Critical Care **26**(2), 158–164 (1997)
39. Yu, S., Wang, C., Ren, K., Lou, W.: Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing. In: 2010 Proceedings IEEE INFOCOM, pp. 1–9. IEEE (2010)
40. Zhang, R.Z.R., Liu, L.L.L.: Security Models and Requirements for Healthcare Application Clouds. 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD) pp. 268–275 (2010)
41. Zhou, Z., Huang, D.: Efficient and Secure Data Storage Operations for Mobile Cloud Computing. In: 2012 8th International Conference on Network and Service Management (CNSM), pp. 37–45. IEEE (2012)