International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org Volume 2 Issue 5 ǁ May. 2013 ǁ PP.17-26

Patient–Centric Secure Data Sharing Frame Work for Cloud-Based PHR Systems

Shaik.Musthafa[1], M.Tech Student, Dora Babu.Sudarsa[2], M.Tech.,(Ph.D), Associate Professor

[1](CSE, Audisankara College of Engineering & Technology, Gudur, Andhrapradesh, India)
[2](CSE, Audisankara College of Engineering & Technology, Gudur, Andhrapradesh, India)

ABSTRACT: *In this paper, we propose a novel patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semitrusted servers. To achieve fine-grained and scalable data access control for PHRs, we leverage attribute-based encryption (ABE) techniques to encrypt each patient's PHR file. Different from previous works in secure data outsourcing, we focus on the multiple data owner scenario, and divide the users in the PHR system into multiple security domains that greatly reduces the key management complexity for owners and users. A high degree of patient privacy is guaranteed simultaneously by exploiting multiauthority ABE. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios. Extensive analytical and experimental results are presented which show the security, scalability, and efficiency of our proposed scheme. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In this paper we present a system for realizing complex access control on encrypted data that we call Ciphertext-Policy Attribute-Based Encryption. We develop a new methodology for utilizing the prior techniques to prove selective security for functional encryption systems as a direct ingredient in devising proofs of full security. This deepens the relationship between the selective and full security models and provides a path for transferring the best qualities of selectively secure systems to fully secure systems. In particular, we present a Ciphertext-Policy Attribute-Based Encryption scheme that is proven fully secure while matching the efficiency of the state of the art selectively secure systems.*

Keywords: *attribute-based encryption , ciphertext-policy, data privacy, key management complexity, Personal health records*

I. INTRODUCTION

In recent years, personal health record (PHR) has emerged as a patient-centric model of health information exchange.A PHR service allows a patient to create, manage, and control her personal health data in one place through the web, which has made the storage, retrieval, and sharing of the medical information more efficient. Especially, each patient is promised the full control of her medical records and can share her health data with a wide range of users, including healthcare providers, family members or friends. Due to the high cost of building and maintaining specialized data centers, many PHR services are outsourced to or provided by third-party service providers, for example, Microsoft HealthVault.[1] Recently, architectures of storing PHRs in cloud computing have been proposed in [2], [3]. While it is exciting to have convenient PHR services for everyone, there are many security and privacy risks which could impede its wide adoption. The main concern is about whether the patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. On the one hand, although there exist healthcare regulations such as HIPAA which is recently amended to incorporate business associates [4], cloud providers are usually not covered entities [5]. On the other hand, due to the high value of the sensitive PHI, the third-party storage servers are often the targets of various malicious behaviors which may lead to exposure of the PHI. As a famous incident, a Department of Veterans Affairs database containing sensitive PHI of 26.5 million military veterans, including their social security numbers and health problems was stolen by an employee who took the data home without authorization [6]. To ensure patient-centric privacy control over their own PHRs, it is essential to have fine-grained data access control mechanisms that work with semitrusted servers.A feasible and promising approach would be to encrypt the data before outsourcing. Basically, the PHR owner herself should decide how to encrypt her files and to allow which set of users to obtain access to each file. A PHR file should only be available to the users who are given the corresponding decryption key, while remain confidential to the rest of users. Furthermore, the patient shall always retain the right to not only grant, but also revoke access privileges when they feel it is necessary [7]. However, the goal of patient-centric privacy is often in conflict with scalability in a PHR system. The authorized users may either need to access the PHR for personal use or professional purposes. Examples of the former are family member and friends, while the latter can be medical doctors, pharmacists, and researchers, etc. We refer to the two

categories of users as personal and professional users, respectively. The latter has potentially large scale; should each owner herself be directly responsible for managing all the professional users, she will easily be overwhelmed by the key management overhead. In addition, since those users' access requests are generally unpredict-able, it is difficult for an owner to determine a list of them. On the other hand, different from the single data owner scenario considered in most of the existing works [8], [9], in a PHR system, there are multiple owners who may encrypt according to their own ways, possibly using different sets of crypto-graphic keys. Letting each user obtain keys from every owner whose PHR she wants to read would limit the accessibility since patients are not always online. An alternative is to employ a central authority (CA) to do the key management on behalf of all PHR owners, but this requires too much trust on a single authority (i.e., cause the key escrow problem).

In this paper, we endeavor to study the patient-centric, secure sharing of PHRs stored on semitrusted servers, and focus on addressing the complicated and challenging key management issues. In order to protect the personal health data stored on a semitrusted server, we adopt attribute-based encryption (ABE) as the main encryption primitive. Using ABE, access policies are expressed based on the attributes of users or data, which enables a patient to selectively share her PHR among a set of users by encrypting the file under a set of attributes, without the need to know a complete list of users. The complexities per encryption, key generation, and decryption are only linear with the number of attributes involved. However, to integrate ABE into a large-scale PHR system, important issues such as key management scalability, dynamic policy updates, and efficient on-demand revocation are nontrivial to solve, and remain largely open up-to-date. To this end, we make the following main contributions:

1. We propose a novel ABE-based framework for patient-centric secure sharing of PHRs in cloud computing environments, under the multiowner settings. To address the key management challenges, we conceptually divide the users in the system into two types of domains, namely public and personal domains (PSDs). In particular, the majority professional users are managed distributively by attribute authorities in the former, while each owner only needs to manage the keys of a small number of users in her personal domain. In this way, our framework can simultaneously handle different types of PHR sharing applications' requirements, while incurring minimal key management overhead for both owners and users in the system. In addition, the framework enforces write access control, handles dynamic policy updates, and provides break-glass access to PHRs under emergence scenarios.

2. In the public domain, we use multiauthority ABE (MA-ABE) to improve the security and avoid key escrow problem. Each attribute authority (AA) in it governs a disjoint subset of user role attributes, while none of them alone is able to control the security of the whole system. We propose mechanisms for key distribution and encryption so that PHR owners can specify personalized fine-grained role-based access policies during file encryption. In the personal domain, owners directly assign access privileges for personal users and encrypt a PHR file under its data attributes. Furthermore, we enhance MA-ABE by putting forward an efficient and on-demand user/ attribute revocation scheme, and prove its security under standard security assumptions. In this way, patients have full privacy control over their PHRs.

We provide a thorough analysis of the complexity and scalability of our proposed secure PHR sharing solution, in terms of multiple metrics in computation, communication, storage, and key management. We also compare our scheme to several previous ones in complexity, scalability and security. Further-more, we demonstrate the efficiency of our scheme by implementing it on a modern workstation and performing experiments/simulations.

Compared with the preliminary version of this paper [1], there are several main additional contributions: 1) We clarify and extend our usage of MA-ABE in the public domain, and formally show how and which types of user-defined file access policies are realized. 2) We clarify the proposed revocable MA-ABE scheme, and provide a formal security proof for it. 3) We carry out both real-world experiments and simulations to evaluate the performance of the proposed solution in this paper.

II. RELATED WORK

This paper is mostly related to works in cryptographically enforced access control for outsourced data and attribute based encryption. To realize fine-grained access control, the traditional public key encryption (PKE)-based schemes [8], [10] either incur high key management overhead, or require encrypting multiple copies of a file using different users' keys. To improve upon the scalability of the above solutions, one-to-many encryption methods such as ABE can be used. In Goyal et al.'s seminal paper on ABE [11], data are encrypted

under a set of attributes so that multiple users who possess proper keys can decrypt. This potentially makes encryption and key management more efficient [12]. A fundamental property of ABE is preventing against user collusion. In addition, the encryptor is not required to know the ACL.

2.1 ABE for Fine-Grained Data Access Control

A number of works used ABE to realize fine-grained access control for outsourced data [13], [14], [9], [15]. Especially, there has been an increasing interest in applying ABE to secure electronic healthcare records (EHRs). Recently, Narayan et al. proposed an attribute-based infrastructure for EHR systems, where each patient's EHR files are encrypted using a broadcast variant of CP-ABE [16] that allows direct revocation. However, the ciphertext length grows linearly with the number of unrevoked users. In [17], a variant of ABE that allows delegation of access rights is proposed for encrypted EHRs. Ibraimi et al. [18] applied ciphertext policy ABE (CP-ABE) [19] to manage the sharing of PHRs, and introduced the concept of social/professional domains. In [20], Akinyele et al. investigated using ABE to generate self-protecting EMRs, which can either be stored on cloud servers or cellphones so that EMR could be accessed when the health provider is offline.

However, there are several common drawbacks of the above works. First, they usually assume the use of a single trusted authority (TA) in the system. This not only may create a load bottleneck, but also suffers from the key escrow problem since the TA can access all the encrypted files, opening the door for potential privacy exposure. In addition, it is not practical to delegate all attribute management tasks to one TA, including certifying all users' attributes or roles and generating secret keys. In fact, different organizations usually form their own (sub)domains and become suitable authorities to define and certify different sets of attributes belonging to their (sub)domains (i.e., divide and rule). For example, a professional association would be responsible for certifying medical specialties, while a regional health provider would certify the job ranks of its staffs. Second, there still lacks an efficient and on-demand user revocation mechanism for ABE with the support for dynamic policy updates/changes, which are essential parts of secure PHR sharing. Finally, most of the existing works do not differentiate between the personal and public domains (PUDs), which have different attribute definitions, key management requirements, and scalability issues. Our idea of conceptually dividing the system into two types of domains is similar with that in [18]; however, a key difference is in [18] a single TA is still assumed to govern the whole professional domain.

2.2 Revocable ABE

It is a well-known challenging problem to revoke users/ attributes efficiently and on-demand in ABE. Traditionally, this is often done by the authority broadcasting periodic key updates to unrevoked users frequently [13], [22], which does not achieve complete backward/forward security and is less efficient. Recently, [23] and [24] proposed two CP-ABE schemes with immediate attribute revocation capability, instead of periodical revocation. However, they were not designed for MA-ABE.

In addition, Ruj et al. [25] proposed an alternative solution for the same problem in our paper using Lewko and Waters's (LW) decentralized ABE scheme [26]. The main advantage of their solution is, each user can obtain secret keys from any subset of the TAs in the system, in contrast to the CC MA-ABE. The LW ABE scheme enjoys better policy expressive-ness, and it is extended by [25] to support user revocation. On the downside, the communication overhead of key revocation is still high, as it requires a data owner to transmit an updated ciphertext component to every nonrevoked user. They also do not differentiate personal and public domains.

In this paper, we bridge the above gaps by proposing a unified security framework for patient-centric sharing of PHRs in a multidomain, multiauthority PHR system with many users. The framework captures application-level requirements of both public and personal use of a patient's PHRs, and distributes users' trust to multiple authorities that better reflects reality. We also propose a suite of access control mechanisms by uniquely combining the technical strengths of both CC MA-ABE [21] and the YWRL ABE scheme [9]. Using our scheme, patients can choose and enforce their own access policy for each PHR file, and can revoke a user without involving high overhead. We also implement part of our solution in a prototype PHR system.

III. FRAMEWORK FOR PATIENT-CENTRIC, SECURE AND SCALABLE PHR SHARING

In this section, we describe our novel patient-centric secure data sharing framework for cloud-based PHR systems.

3.1 Problem Definition

We consider a PHR system where there are multiple PHR owners and PHR users. The owners refer to patients who have full control over their own PHR data, i.e., they can create, manage, and delete it. There is a central server belonging to the PHR service provider that stores all the owners' PHRs. The users may come from

various aspects; for example, a friend, a caregiver or a researcher. Users access the PHR documents through the server in order to read or write to someone's PHR, and a user can simultaneously have access to multiple owners' data.

A typical PHR system uses standard data formats. For example, continuity-of-care (CCR) (based on XML data structure), which is widely used in representative PHR systems including Indivo [27], an open-source PHR system adopted by Boston Children's Hospital. Due to the nature of XML, the PHR files are logically organized by their categories in a hierarchical way [8], [20].

3.1.1 Security Model

In this paper, we consider the server to be semitrusted, i.e., honest but curious as those in [28] and [15]. That means the server will try to find out as much secret information in the stored PHR files as possible, but they will honestly follow the protocol in general. On the other hand, some users will also try to access the files beyond their privileges. For example, a pharmacy may want to obtain the prescriptions of patients for marketing and boosting its profits. To do so, they may collude with other users, or even with the server. In addition, we assume each party in our system is preloaded with a public/private key pair, and entity authentication can be done by traditional challenge-response protocols.

3.1.2 Requirements

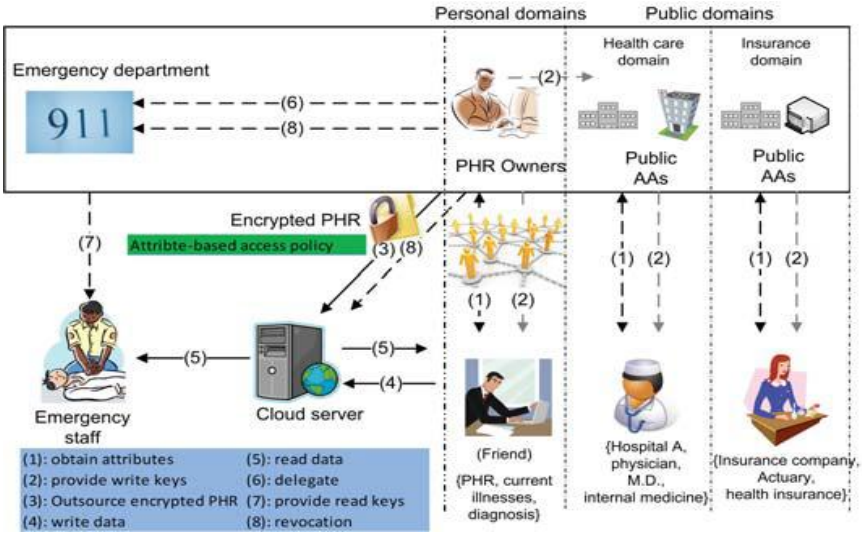
To achieve "patient-centric" PHR sharing, a core requirement is that each patient can control who are authorized to access to her own PHR documents. Especially, user-controlled read/write access and revocation are the two core security objectives for any electronic health record system, pointed out by Mandl et al. [7] in as early as 2001. The security and performance requirements are summarized as follows:

- Data confidentiality. Unauthorized users (including the server) who do not possess enough attributes satisfying the access policy or do not have proper key access privileges should be prevented from decrypting a PHR document, even under user collusion. Fine-grained access control should be enforced, meaning different users are authorized to read different sets of documents.
- On-demand revocation. Whenever a user's attribute is no longer valid, the user should not be able to access future PHR files using that attribute. This is usually called attribute revocation, and the corresponding security property is forward secrecy [23]. There is also user revocation, where all of a user's access privileges are revoked.
- Write access control. We shall prevent the unauthor-ized contributors to gain write-access to owners' PHRs, while the legitimate contributors should access the server with accountability.
- The data access policies should be flexible, i.e., dynamic changes to the predefined policies shall be allowed, especially the PHRs should be accessible under emergency scenarios.
- Scalability, efficiency, and usability. The PHR system should support users from both the personal domain and public domains. Since the set of users from the public domain may be large in size and unpredict-able, the system should be highly scalable, in terms of complexity in key management, communication, computation and storage. Additionally, the owners' efforts in managing users and keys should be minimized to enjoy usability.

3.2 Overview of Our Framework

The main goal of our framework is to provide secure patient-centric PHR access and efficient key management at the same time. The key idea is to divide the system into multiple security domains (namely, public domains and personal domains) according to the different users' data access requirements. The PUDs consist of users who make access based on their professional roles, such as doctors, nurses, and medical researchers. In practice, a PUD can be mapped to an independent sector in the society, such as the health care, government, or insurance sector. For each PSD, its users are personally associated with a data owner (such as family members or close friends), and they make accesses to PHRs based on access rights assigned by the owner.

In both types of security domains, we utilize ABE to realize cryptographically enforced, patient-centric PHR access. Especially, in a PUD multiauthority ABE is used, in which there are multiple "attribute authorities" (AAs), each governing a disjoint subset of attributes. Role attributes are defined for PUDs, representing the professional role or obligations of a PUD user. Users in PUDs obtain their attribute-based secret keys from the AAs, without directly interacting with the owners. To control access from PUD users, owners are free to specify role-based fine-grained access policies for her PHR files, while do not need to know the list of authorized users when doing encryption. Since the PUDs contain the majority of users, it greatly reduces the key management overhead for both the owners and users.



Each data owner (e.g., patient) is a trusted authority of her own PSD, who uses a KP-ABE system to manage the secret keys and access rights of users in her PSD. Since the users are personally known by the PHR owner, to realize patient-centric access, the owner is at the best position to grant user access privileges on a case-by-case basis. For PSD, data attributes are defined which refer to the intrinsic properties of the PHR data, such as the category of a PHR file. For the purpose of PSD access, each PHR file is labeled with its data attributes, while the key size is only linear with the number of file categories a user can access. Since the number of users in a PSD is often small, it reduces the burden for the owner. When encrypting the data for PSD, all that the owner needs to know is the intrinsic data properties.

The multidomain approach best models different user types and access requirements in a PHR system. The use of ABE makes the encrypted PHRs self-protective, i.e., they can be accessed by only authorized users even when storing on a semitrusted server, and when the owner is not online. In addition, efficient and on-demand user revocation is made possible via our ABE enhancements.

3.3 Details of the Proposed Framework

In our framework, there are multiple SDs, multiple owners, multiple AAs, and multiple users. In addition, two ABE systems are involved: for each PSD the YWRL's revocable KP-ABE scheme [9] is adopted; for each PUD, our proposed revocable MA-ABE scheme (described in Section 4) is used. The framework is illustrated in Fig. We term the users having read and write access as data readers and contributors, respectively.

System setup and key distribution. The system first defines a common universe of data attributes shared by every PSD, such as "basic profile," "medical history," "allergies," and "prescriptions." An emergency attribute is also defined for break-glass access. Each PHR owner's client application generates its corresponding public/master keys. The public keys can be published via user's profile in an online healthcare social-network (HSN) (which could be part of the PHR service; e.g., the Indivo system [27]). There are two ways for distributing secret keys. First, when first using the PHR service, a PHR owner can specify the access privilege of a data reader in her PSD, and let her application generate and distribute corresponding key to the latter, in a way resembling invitations in GoogleDoc. Second, a reader in PSD could obtain the secret key by sending a request (indicating which types of files she wants to access) to the PHR owner via HSN, and the owner will grant her a subset of requested data types. Based on that, the policy engine of the application automatically derives an access structure, and runs keygen of KP-ABE to generate the user secret key that embeds her access structure. In addition, the data attributes can be organized in a hierarchical manner for efficient policy generation, see Fig. 2. When the user is granted all the file types under a category, her access privilege will be represented by that category instead.

For the PUDs, the system defines role attributes, and a reader in a PUD obtains secret key from AAs, which binds the user to her claimed attributes/roles. For example, a physician in it would receive "hospital A, physician, M.D., internal medicine" as her attributes from the AAs. In practice, there exist multiple AAs each governing a different subset of role attributes. For instance, hospital staffs shall have a different AA from pharmacy specialists. This is reflected by (1) in Fig. MA-ABE is used to encrypt the data, and the concrete mechanism will be presented in Section 4. In addition, the AAs distribute write keys that permit contributors in their PUD to write to some patients' PHR (2).

PHR encryption and access. The owners upload ABE-encrypted PHR files to the server (3). Each

owner's PHR file is encrypted both under a certain fine-grained and role-based access policy for users from the PUD to access, and under a selected set of data attributes that allows access from users in the PSD. Only authorized users can decrypt the PHR files, excluding the server. For improving efficiency, the data attributes will include all the intermediate file types from a leaf node to the root. For example, an "allergy" file's attributes are fPHR; medical history; allergyg. The data readers download PHR files from the server, and they can decrypt the files only if they have suitable attribute-based keys (5). The data contributors will be granted write access to someone's PHR, if they present proper write keys (4).

User revocation. Here, we consider revocation of a data reader or her attributes/access privileges. There are several possible cases:

- revocation of one or more role attributes of a public domain user
- revocation of a public domain user which is equiva-lent to revoking all of that user's attributes. These operations are done by the AA that the user belongs to, where the actual computations can be delegated to the server to improve efficiency (8).
- Revocation of a personal domain user's access privileges
- revocation of a personal domain user. These can be initiated through the PHR owner's client application in a similar way.

Policy updates. A PHR owner can update her sharing policy for an existing PHR document by updating the attributes (or access policy) in the ciphertext. The supported operations include add/delete/modify, which can be done by the server on behalf of the user.

Break-glass. When an emergency happens, the regular access policies may no longer be applicable. To handle this situation, break-glass access is needed to access the victim's PHR. In our framework, each owner's PHR's access right is also delegated to an emergency department (ED, (6)). To prevent from abuse of break-glass option, the emergency staff needs to contact the ED to verify her identity and the
emergency situation, and obtain temporary read keys (7). After the emergency is over, the patient can revoke the emergent access via the ED.

An example. Here, we demonstrate how our framework works using a concrete example. Suppose PHR owner Alice is a patient associated with hospital A. After she creates a PHR file F1 (labeled as "PHR; medical history; allergy; emergency" in Fig. 2), she first encrypts it according to both F1's data labels (under the YWRL KP-ABE), and a role-based file access policy P1 (under our revocable MA-ABE). This policy can be decided based on recommended settings by the system, or Alice's own preference. It may look like

P1 :¼ ''ðprofession ¼ physicianÞ

^ðspecialty ¼ internal medicineÞ

^ðorganization ¼ hospital AÞ'':

She also sends the break-glass key to the ED. In addition, Alice determines the access rights of users in her PSD, which can be done either online or offline. For example, she may approve her friend Bob's request to access files with labels fpersonal infog or fmedical historyg. Her client application will distribute a secret key with the access structure ðpersonal info _ medical historyÞ to Bob. When Bob wants to access another file F2 with labels "PHR—me-dical history—medications," he is able to decrypt F2 due to the "medical history" attribute. For another user Charlie who is a physician specializing in internal medicine in hospital B in the PUD, he obtains his secret key from multiple AAs such as the American Medical Association (AMA), the American Board of Medical Specialties (ABMS), and the American Hospital Association (AHA). But he cannot decrypt F1, because his role attributes do not satisfy P1. Finally, an emergency room staff, Dorothy who temporarily obtains the break-glass key from ED, can gain access to F1 due to the emergency attribute in that key.

Remarks. The separation of PSD/PUD and data/role attributes reflects the real-world situation. First, in the PSD, a patient usually only gives personal access of his/her sensitive PHR to selected users, such as family members and close friends, rather than all the friends in the social network. Different PSD users can be assigned different access privileges based on their relationships with the owner. In this way, patients can exert fine-control over the access for each user in their PSDs. Second, by our multi-domain and multiauthority framework, each public user only needs to contact AAs in its own PUD who collabora-tively generates a secret key for the user, which reduces the workload per AA (since each AA handles fewer number of attributes per key issuing).

In addition, the multiauthority ABE is resilient to compromise of up to N _ 2 AAs in a PUD, which solves the key-escrow problem. Furthermore, in our framework user's role verification is much easier. Different organizations can form their own (sub)domains and become AAs to manage and certify different sets of attributes, which is similar to divide and rule.

IV. SECURITY ANALYSIS

In this section, we analyze the security of the proposed PHR sharing solution. First we show it achieves data confidenti-ality (i.e., preventing unauthorized read accesses), by proving the enhanced MA-ABE scheme (with efficient revocation) to be secure under the attribute-based selective-set model [21], [34]. We have the following main theorem .

Theorem 2. The enhanced MA-ABE scheme guarantees data confidentiality of the PHR data against unauthorized users and the curious cloud service provider, while maintaining the collusion resistance against users and up to N _ 2 AAs addition, our framework achieves forward secrecy, and security of write access control. For detailed security analysis and proofs, please refer to the online supplementary material, available online, of this paper.

We also compare the security of our scheme with several existing works, in terms of confidentiality guarantee, access control granularity, and supported revocation method, etc. We choose four representative state-of-the-art schemes to compare with:

1. the VFJPS scheme [28] based on access control list (ACL);
2. the BCHL scheme based on HIBE [8] where each owner acts as a key distribution center;
3. the HN revocable CP-ABE scheme [23], where we adapt it by assuming using one PUD with a single authority and multiple PSDs to fit our setting;
4. the NGS scheme in [16] which is a privacy-preserving EHR system that adopts attribute-based broadcast encryption to achieve data access control;
5. The RNS scheme in [25] that enhances the Lewko-Waters MA-ABE with revocation capability for data access control in the cloud.

It can be seen that, our scheme achieves high privacy guarantee and on-demand revocation. The conjunctive policy restriction only applies for PUD, while in PSD a user's access structure can still be arbitrary monotonic formula. In comparison with the RNS scheme, in RNS the AAs are independent with each other, while in our scheme the AAs issue user secret keys collectively and interactively. Also, the RNS scheme supports arbitrary monotonic Boolean formula as file access policy. However, our user revocation method is more efficient in terms of communication overhead. In RNS, upon each revocation event, the data owner needs to recompute and send new ciphertext components corresponding to revoked attributes to all the remaining users. In our scheme, such interaction is not needed. In addition, our proposed frame-work specifically addresses the access requirements in cloud-based health record management systems by logically dividing the system into PUD and PSDs, which considers both personal and professional PHR users. Our revocation methods for ABE in both types of domains are consistent. The RNS scheme only applies to the PUD.

V. SCALABILITY AND EFFICIENCY

5.1 Storage and Communication Costs

First, we evaluate the scalability and efficiency of our solution in terms of storage, communication, and computa-tion costs. We compare with previous schemes in terms of

Table I: Notations for Efficiency Comparison

S_k	Bit size of a FEK
S_1	Bit size of an element in $\mathbb{G}_1$ / $\mathbb{G}_2$
S_T	Bit size of an element in $\mathbb{G}_T$
S_z	Bit size of an element in $\mathbb{Z}_p^*$
$S_\mathcal{P}$	Bit size of access policy and attribute set in CT
N (or N_i)	Number of AAs in a PUD (or the i-th PUD)
N_o	The number of owners in the system
N_u	The number of data users in the system
N_r	Number of revoked users for a file
N_a	Number of users in an attribute group
m	Number of attribute types in the PUD
t_c, t_u	Total number of attributes appeared in CT, sk_u
l	Depth of file hierarchy of an owner's PHR

Cipher text size, user secret key size, public key/information size, and revocation (rekeying) message size.

The results are given in Table 2. The ciphertext size only accounts for the encryption of FEK. In our scheme, for simplicity we assume there is only one PUD, thus the ciphertext includes m additional wildcard attributes and up to N _ 1 dummy attributes. Our scheme requires a secret key size that is linear with jAuj, the number of attributes of each user, while in the VFJPS and BCHL schemes this is linear with No, since a user needs to obtain at least one key from each owner whose PHR file the user wants to access. For public key size, we count the size of the effective information that each user needs to obtain. The VFJPS scheme requires each owner to publish a directed acyclic graph representing her ACL along with key assignments, which essentially amounts to OðNuÞ per owner. This puts a large burden either in communication or storage cost on the system. For rekeying, we consider revocation of one user by an owner in VFJPS and BCHL. In VFJPS, revoking one user from a file may need overencryption and issuing of new public tokens for all the rest of users in the worst case. The NGS scheme achieves direct user revocation using ABBE, which eliminates the need of rekeying and reencryption; however, attribute revocation is not achieved; and for the revocable ABBE in [32], either the ciphertext size is linear with the number of revoked users, or the public key is linear with the total number of users in the system.4 For the RNS scheme, the main drawback is the large size of revocation messages to be transmitted to nonrevoked users.

These indicate our scheme is more scalable than existing works. To further show the storage and communication costs, we provide a numerical analysis using typical parameter settings in the supplementary material, available online.

5.2 Computation Costs

Next, we evaluate the computational cost of our scheme through combined implementation and simulation. We provide the first implementation of the GPSW KP-ABE scheme [35] (to the best of our knowledge), and also integrated the ABE algorithms into a prototype PHR system, Indivo [27], [36]. The GPSW KP-ABE scheme is tested on a PC with 3.4 GHz processor, using the pairing-based cryptography (PBC) library [37]. The public para-meters are chosen to provide 80 bits security level, and we use a pairing-friendly type-A 160-bit elliptic curve group [37]. This parameter setting has also been adopted in other related works in ABE [19], [38]. We then use the ABE algorithms to encrypt randomly generated XML-formatted files (since real PHR files are difficult to obtain), and implement the user-interfaces for data input and output. Due to space limitations, the details of prototype imple-mentation are reported in [36].

In the supplementary material, available online, we present benchmarks of cryptographic operations and detailed timing results for the two ABE algorithms used by our framework. It is shown that, the decryption operation in our enhanced MA-ABE scheme is quite fast, because it involves only jACPUDj þ 1 pairing operations (in contrast, the RNS scheme involves 2jACPUD j þ 1 pairing operations). The time costs of key generation, encryption, and decryp-tion processes are all linear with the number of attributes. For 50 attributes, they all take less than 0.5 s.

From the system aspect, each data owner (patient) uses the YWRL ABE scheme for setup, key generation and revocation, uses both YWRL and enhanced MA-ABE for encryption. Each PSD user adopts the YWRL scheme for decryption, while each PUD user adopts the enhanced MA-ABE scheme for decryption. Each AA uses enhanced MA-ABE for setup, key generation and revocation. Next, we provide estimations of computation times of each party in the system in Table 6. The values are calculated from the example parameters and benchmark results, where ex-ponentiation times Exp1 ¼ 6:4 ms, ExpT ¼ 0:6 ms, pairing time TP ¼ 2:5 ms.

Finally, we simulate the server's computation cost spent in user revocation to evaluate the system performance of user revocation. Especially, the lazy-revocation method greatly reduces the cost of revocation, because it aggregates multiple ciphertext/key update operations, which amor-tizes the computations over time. The details of the experimental/simulation evaluation results are presented in the supplementary material, available online.

Table II: Comparison of Efficiency

Scheme	Ciphertext size	User secret key size	Public key/info. size	Revocation message
VFJPS [28]	S_k	$N_o \cdot S_k$	$O(N_o \cdot N_u)$	$O(N_u)$
BCHL [8]	$l \cdot S_1 + S_k$	$l \cdot N_o \cdot S_1$	$2S_1 \cdot N_o$	N/A
HN [23]	$(2t_c+1)S_1 + S_T + S_p$	$(2t_u+1)S_1 + 2(\log N_u)S_k$	$2(S_1 + S_T)$	$(N_u - N_a)(\log \frac{N_u}{N_h - N_a})S_z$
NGS [16]	$(t_c + 2N_r)S_1 + S_T$	$(t_u + 4)S_1$	$(\bar{m} + \bar{l} + 6)S_1 + S_T$	0
RNS [25]	$t_c(2S_1 + S_T) + S'_p$	$t_u \cdot S_1$	$\|U\|(S_1 + S_T)$	$O((t_u + 1)S_T \cdot (N_u - N_r))$
Our scheme	$(t_c + m + N - 1)S_1 + S_T + S_p$	$(t_u + m + 1)S_1$	$(\|U\| + N - 1)S_1$	$t_u \cdot S_z$

Table III : Computation Complexity for Each Party in the System, and Numerical Estimation of Time Costs Assuming Following Parameters (Also Used in Supplementary Material, Available Online): jUDj ¼ 50, jURj ¼ 100, N ¼ 5 (Number of AAs), jACPSDj ¼ 5, jACPUDj ¼ 35, jAuj ¼ m ¼ 15, jLðT Þj ¼ 10, j_0j ¼ 5 (a Minimal Number of Attributes to Revoke a User)

	Setup	KeyGen. (per user)	Enc. (per file)	Dec. (per file)	User revo.
Owner	$\|\mathcal{U}_D\|\mathsf{Exp}_1 + \mathsf{Exp}_T$	$\|I_A(T)\|\mathsf{Exp}_1$	$(\|\mathbb{A}^C_{PSD}\| + \|\mathbb{A}^C_{PUD}\| + 1)\mathsf{Exp}_1 + 2\mathsf{Exp}_T$	/	$\|\gamma'\|\mathsf{Exp}_1$
Estimate (s)	0.32	0.064	0.264	/	0.032
PSD user	/	/	/	$\sim \|L(T)\|T_P$	/
Estimate (s)	/	/	/	0.025	/
PUD user	/	/	/	$\sim (\|\mathbb{A}^u\| + m + 1)T_P$	/
Estimate (s)	/	/	/	0.078	/
k-th AA	$(\|\mathcal{U}_R\|_k + 1)\mathsf{Exp}_1 + \mathsf{Exp}_T$	$\sim \|\mathbb{A}^u_k\|\mathsf{Exp}_1$	/	/	$\|\gamma'\|\mathsf{Exp}_1$
Estimate (s)	0.135	0.036	/	/	0.032

VI. CONCLUSION

In this paper, we have proposed a novel framework of Patient–Centric Secure Data Sharing Frame Work for Cloud-Based PHR Systems. Considering partially trustworthy cloud servers, we argue that to fully realize the patient-centric concept, patients shall have complete control of their own privacy through encrypting their PHR files to allow fine-grained access. The framework addresses the unique challenges brought by multiple PHR owners and users, in that we greatly reduce the complexity of key management while enhance the privacy guarantees compared with previous works. We utilize ABE to encrypt the PHR data, so that patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications, and affiliations. Further-more, we enhance an existing MA-ABE scheme to handle efficient and on-demand user revocation, and prove its security. Through implementation and simulation, we show that our solution is both scalable and efficient.

In the future, it would be interesting to con-sider attribute-based encryption systems with different types of expressibility. While, Key-Policy ABE and Ciphertext-Policy ABE capture two interesting and complimentary types of systems there certainly exist other types of systems. The primary challenge in this line of work is to find a new systems with elegant forms of expression that produce more than an arbitrary com-bination of techniques.

REFERENCES

[1]. M. Li, S. Yu, K. Ren, and W. Lou, "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings," Proc. Sixth Int'l ICST Conf. Security and Privacy in Comm. Networks (SecureComm '10), pp. 89-106, Sept. 2010.

[2]. H. Lo¨ hr, A.-R. Sadeghi, and M. Winandy, "Securing the E-Health Cloud," Proc. First ACM Int'l Health Informatics Symp. (IHI '10), pp. 220-229, 2010.

[3]. M. Li, S. Yu, N. Cao, and W. Lou, "Authorized Private Keyword Search over Encrypted Personal Health Records in Cloud Computing," Proc. 31st Int'l Conf. Distributed Computing Systems (ICDCS '11), June 2011.

[4]. "The Health Insurance Portability and Accountability Act," http://www.cms.hhs.gov/HIPAAGenInfo/01_Overview.asp, 2012.

[5]. "Google, Microsoft Say Hipaa Stimulus Rule Doesn't Apply to Them," http://www.ihealthbeat.org/Articles/2009/4/8/, 2012.

[6]. "At Risk of Exposure - in the Push for Electronic Medical Records, Concern Is Growing About How Well Privacy Can Be Safeguarded," http://articles.latimes.com/2006/jun/26/health/ he-privacy26, 2006.

[7]. K.D. Mandl, P. Szolovits, and I.S. Kohane, "Public Standards and Patients' Control: How to Keep Electronic Medical Records Accessible but Private," BMJ, vol. 322, no. 7281, pp. 283-287, Feb. 2001.

[8]. J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 103-114, 2009.

[9]. S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM '10, 2010.

[10]. C. Dong, G. Russello, and N. Dulay, "Shared and Searchable Encrypted Data for Untrusted Servers," J.Computer Security, vol. 19, pp. 367-397, 2010.

[11]. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.

[12]. M. Li, W. Lou, and K. Ren, "Data Security and Privacy in Wireless Body Area Networks," IEEE Wireless Comm. Magazine, vol. 17, no. 1, pp. 51-58, Feb. 2010.

[13]. A. Boldyreva, V. Goyal, and V. Kumar, "Identity-Based Encryption with Efficient Revocation," Proc. 15th ACM Conf. Computer and Comm. Security (CCS), pp. 417-426, 2008.

[14]. L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Ciphertext-Policy Attribute-Based Threshold Decryption with Flexible Delegation and Revocation of User Attributes," 2009.

[15]. S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS '10), 2010.

[16]. S. Narayan, M. Gagne´, and R. Safavi-Naini, "Privacy Preserving EHR System Using Attribute-Based Infrastructure," Proc. ACM Cloud Computing Security Workshop (CCSW '10), pp. 47-52, 2010.

[17]. X. Liang, R. Lu, X. Lin, and X.S. Shen, "Patient Self-Controllable Access Policy on Phi in Ehealthcare Systems," Proc. Advances in Health Informatics Conf. (AHIC 10), 2010.

[18]. L. Ibraimi, M. Asim, and M. Petkovic, "Secure Management of Personal Health Records by Applying Attribute-Based Encryption," technical report, Univ. of Twente, 2009.

[19]. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy (SP '07), pp. 321-334, 2007.

[20]. J.A. Akinyele, C.U. Lehmann, M.D. Green, M.W. Pagano, Z.N.J.Peterson, and A.D. Rubin, "Self- rotecting Electronic Medical Records Using Attribute-Based Encryption," Cryptology ePrint Archive, Report 2010/565, http://eprint.iacr.org/, 2010.

[21]. M. Chase and S.S. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 121-130, 2009.

[22]. X. Liang, R. Lu, X. Lin, and X.S. Shen, "Ciphertext Policy Attribute Based Encryption with Efficient Revocation," technical report, Univ. of Waterloo, 2010.

[23]. J. Hur and D.K. Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214-1221, July 2011.

[24]. S. Jahid, P. Mittal, and N. Borisov, "Easier: Encryption-Based Access Control in Social Networks with Efficient Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), Mar. 2011.

[25]. S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," Proc. IEEE 10th Int'l Conf. Trust, Security and Privacy in Computing and Comm. (TrustCom), 2011.

[26]. A. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," EUROCRYPT: Proc. 30th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology, pp. 568-588, 2011.

[27]. "Indivo." http://indivohealth.org/, 2012.

[28]. S.D.C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-Encryption: Management of Access Control Evolution on Outsourced Data," Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB '07), pp. 123-134, 2007.

[29]. A. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," EUROCRYPT: Proc. 30th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology, pp. 568-588, 2011.

[30]. A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler, "Spins: Security Protocols for Sensor Networks," Wireless Networking, vol. 8, pp. 521-534, Sept. 2002.

[31]. H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in Mobile Ad Hoc Networks: Challenges and Solutions," IEEE Wireless Comm., vol. 11, no. 1, pp. 38-47, Feb. 2004.

[32]. N. Attrapadung and H. Imai, "Conjunctive Broadcast and Attribute-Based Encryption," Proc. Third Int'l Conf. Palo Alto on Pairing-Based Cryptography-Pairing, pp. 248-265, 2009.

[33]. S. Mu¨ ller, S. Katzenbeisser, and C. Eckert, "Distributed Attribute-Based Encryption," Proc. 11th Int'l Conf. Information Security and Cryptology (ICISC 08), pp. 20-36, 2009.

[34]. S. Chow, "New Privacy-Preserving Architectures for Identity-/ Attribute-Based Encryption," PhD thesis, NYU, 2010.

[35]. Y. Zheng, "Key-Policy Attribute-Based Encryption Scheme Implementation," http://www.cnsr.ictas.vt.edu/resources.html, 2012.

[36]. Y. Zheng, "Privacy-Preserving Personal Health Record System Using Attribute-Based Encryption," master's thesis, Worcester Polytechnic Inst., 2011.

[37]. B. Lynn, "The Pbc Library," http://crypto.stanford.edu/pbc/, 2012.

[38]. M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure Attribute-Based Systems," J. Computer Security, vol. 18, no. 5, pp. 799-837, 2010.