Authenticated Sharing of Personal Health Records in Cloud

Murali N.S.1, Dr. D.Thilagavathy2
1PG Scholar, 2HOD, Department of CSE, Adhiyamaan College of Engineering, Hosur, India
muralimano21@gmail.com

Abstract
Personal health record (PHR) is an emerging patient-centric model of health information exchange, which stored at a third party, such as cloud providers. However, there have been wide privacy concerns as personal health information could be exposed to those third party servers and to unauthorized parties. Different from previous works in secure data outsourcing, 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. Proposed a novel patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semi trusted servers. Extensive analytical and experimental results are presented which show the security, scalability, and efficiency.

Keywords: Personal health records, data privacy, fine-grained access control, attribute-based encryption

1. Introduction
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 centres, many PHR services are outsourced to or provided by third-party service providers, for example, Microsoft Health Vault. Recently, architectures of storing PHRs in cloud computing have been proposed. 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 patient 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, cloud providers are usually not covered entities. 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. To ensure patient-centric privacy control over their own PHRs, it is essential to have fine-grained data access control mechanisms that work with semi trusted 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. However the goal of patient-centric privacy is in conflict with scalability in a PHR system. 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. The following main contribution


[214-219]
1. ABE-based framework for patient-centric secure sharing of PHRs in cloud computing environments, under the multi owner 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, use multi authority 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.

II. Related Work

This 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, 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 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. A fundamental property of ABE is preventing against user collusion. In addition, the encryption is not required to know the ACL.

A. ABE for Fine-Grained Data Access Control

A number of works used ABE to realize fine-grained access control for outsourced data. Especially, there has been an increasing interest in applying ABE to secure electronic health care records (EHRs). An attribute-based infrastructure for EHRs, with each patient’s EHR file are encrypted using a broadcast variant of CP-ABE that allows direct revocation. However, the cipher text length grows linearly with the number of unrevoked users. A variant of ABE that allows delegation of access rights is proposed for encrypted EHRs. Ibrahim et al. applied cipher text policy ABE (CP-ABE) to manage the sharing of PHRs, and introduced the concept of social/professional domains. In, Akinyele et al. investigated using ABE to generate self-protecting EMRs, which can either be stored on cloud servers or cell phones 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 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 ; however, a key difference is in a single TA is still assumed to govern the whole professional domain.

B. 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 which does not achieve complete backward/forward security and is less efficient. Recently, and proposed two CP-ABE schemes with immediate attribute revocation capability, instead of periodical revocation. However, they were not designed for MA-ABE.

In addition, Raj et al. proposed an alternative solution for the same problem in our paper using Lewko and Waters’s (LW) decentralized ABE scheme. 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 expressiveness, and it is extended by 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 cipher text component to every non revoked user. They also do not differentiate personal and public domains.

By proposing a unified security framework for patient-centric sharing of PHRs in a multi domain, multi authority PHR system with many users. The framework 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 and ABE scheme. 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.

would certify the job ranks of its

III. Framework for Patient-Centric, Secure and Scalable PHR Sharing

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

A. Problem Definition

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 user can simultaneously 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, 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.

B. Security Model

Consider the server to be semi trusted, i.e., honest but curious as those. 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.

C. 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, the security and performance requirements are summarized as follows:

http://www.ijesrt.com
D. Details of the Proposed Frame

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 is adopted; for each PUD, our proposed revocable MA-ABE scheme (described in Section 4) is used. The framework is illustrated in Fig. 1. We term the users having read and write access as data readers and contributors, respectively. 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). 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.

The owners upload ABE-encrypted PHR files to the server. 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.

IV. MA-ABE in the Public Domain

For the PUDs, our framework delegates the key management functions to multiple attribute authorities. In order to achieve stronger privacy guarantee for data owners, the Chase-Chow (CC) MA-ABE scheme is used, where each authority governs a disjoint set of attributes distributive. It is natural to associate the ciphertext of a PHR document with an owner-specified access policy for users from PUD. However, one technical challenge is that CC MA-ABE is essentially a KP-ABE scheme, where the access policies are enforced in users’ secret keys, and those key-policies do not directly translate to document access policies from the owners’ points of view. By our design, we show that by agreeing upon the formats of the key-policies and the rules of specifying which attributes are required in the ciphertext, the CC MA-ABE can actually support owner-specified document access policies with some degree staff authenticates herself to the ED, requests and obtains the corresponding patient’s skEM, and
then decrypts the PHR documents using skEM. After the patient recovers from the emergency, she can revoke the break-glass access by computing a rekey: rkEM, submit it to the ED and the server to update her skEM and CT to their newest versions, respectively.

### V. Security Analysis

In this section, we analyze the security of the proposed PHR sharing solution. First we show it achieves data confidentiality (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. We have the following main theorem.

Theorem: The enhanced MA-ABE scheme guarantees data confidentiality of the PHR data against unauthorized users and the curious cloud service provider, while maintaining the collision resistance against users and up to N - 2 AAs.

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

1. The VFJPS scheme based on access Control list (ACL);
2. The BC HL scheme based on HI BE, where each owner acts as a key distribution center;
3. The HN scheme in which is a CP-ABE scheme, 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 which is a privacy-preserving EHR system that adopts attribute-based broadcast encryption to achieve data access control;
5. The RNS scheme in which is a Lewko- Waters MA-ABE with revocation capability for data access control in the cloud.

The conjunctive policy restriction only applies for PUD, while in PSD a user’s access of flexibility i.e., it functions similar to CP-ABE Achieving More Expressive File Access Policies By enhancing the key-policy generation rule, we can enable more expressive encryption’s access policies. We exploit an observation that in practice, a user’s attributes/roles belonging to different types are often correlated ated with respectively to a primary attribute type. In the following, an attribute tuple refers to the set of attribute values over needed by one AA (each of a different type) that are correlated with each other.

#### A. Enforce Write Access Control

If there is no restriction on write access, anyone may write to someone’s PHR using only public keys, which is undesirable. By granting write access, we mean a data contributor should obtain proper authorization from the organization she is in (and/or from the targeting owner), which shall be able to be verified by the server who grants/rejects write access. A naive way is to let each contributor obtain a signature from her organization every time she intends to write. Yet this requires the organizations be always online. The observation is that, it is desirable and practical to authorize according to time periods whose granularity can be adjusted. For example, a doctor should be permitted to write only during her office hours; on the other hand, the doctor must not be able to write to patients that are not treated by her. Therefore, we combine signatures with the hash chain technique to achieve our goals.

#### B. Deal with Break-Glass Access

For certain parts of the PHR data, medical staffs need to have temporary access when an emergency happens to a patient, who may become unconscious and is unable to change her access policies beforehand. The medical staffs will need to delegate her emergency key to an emergency department. Specifically, in the beginning, each owner defines an “emergency” attribute and builds it into the PSD part of the cipher text of each PHR document that she allows break-glass access. She then generates an emergency key skEM using the single-node key-policy “emergency,” and delegates it to the ED who keeps it in a database of patient directory. Upon emergency, a medical staff 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 framework 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. The RNS scheme only applies to the PUD.

<table>
<thead>
<tr>
<th>Notations for Efficiency Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_k$</td>
</tr>
<tr>
<td>$s_1$</td>
</tr>
<tr>
<td>$s_T$</td>
</tr>
<tr>
<td>$s_2$</td>
</tr>
<tr>
<td>$s_p$</td>
</tr>
<tr>
<td>$N_0$</td>
</tr>
<tr>
<td>$N_u$</td>
</tr>
<tr>
<td>$N_v$</td>
</tr>
<tr>
<td>$N_a$</td>
</tr>
<tr>
<td>$m$</td>
</tr>
<tr>
<td>$l_o, l_u$</td>
</tr>
<tr>
<td>$l$</td>
</tr>
</tbody>
</table>

VI. Conclusion

In this paper, I have proposed a framework of secure sharing of personal health records in cloud computing. Considering partially trustworthy cloud servers, patients shall have complete control of their own privacy through encrypting their PHR files to allow fine-grained access. 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. The framework addresses the unique challenges brought by multiple PHR owners and users, in that greatly reduce the complexity of key management while enhance the privacy guarantees compared with previous works. Through implementation and simulation, showed that my solution is both scalable and efficient.

VII. References