Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-Based Encryption

CHITRAM RAMESH1, D. KRISHNA2, G. DEEPTHI3

1PG Scholar, Dept of CSE, Jawaharlal Nehru Institute of Technology, Hyderabad, India, E-mail: ramesh5545@gmail.com.
2HOD & Associate Professor, Dept of CSE, Jawaharlal Nehru Institute of Technology, Hyderabad, India.
3Assistant Professor, Dept of CSE, Jawaharlal Nehru Institute of Technology, Hyderabad, India.

Abstract: Personal health record is maintain in the centralize server to maintain patient’s personal and diagnosis information. Personal health record (PHR) is an emerging patient-centric model of health information exchange, which is often outsourced to be 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. The security schemes are used to protect personal data from public access. To assure the patients’ control over access to their own PHRs, it is a promising method to encrypt the PHRs before outsourcing. In this paper we propose novel patient-centric framework and suite of mechanism for data access control to PHR’s stored in semi trusted 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. Data owner update the personal data into third party cloud data centers. Multiple data owners can access the same data values. A high degree of patient privacy is guaranteed simultaneously by exploiting multi-authority 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.

Keywords: Personal Health Records, Cloud Computing, Data Privacy, Fine-Grained Access Control, Attribute-Based Encryption.

I. INTRODUCTION

Cloud computing is an emerging computing technology where applications and all the services are provided via Internet. It is a model for enabling on-demand network access to pool resources. Cloud computing can be considered as a computing paradigm with greater flexibility and availability at lower cost. In recent year, Personal Health Record (PHR) has developed as the emerging trend in the health care technology and by which the patients are efficiently able to create, manage and share their personal health information. This PHR is now a day’s stored in the clouds for the cost reduction purpose and for the easy sharing and access mechanism. The main concern about this PHR is that whether the patient is able to control their data or not. It is very essential to have the fine grained access control over the data with the semi-trusted server. But in this the PHR system, the security, privacy and health data confidentiality are making challenges to the users when the PHR stored in the third party storage area like cloud services. The PHR data should be secured from the external attackers and also it should be protect from the internal attackers such that from the cloud server organization itself. When the PHR owner upload the PHR data to the cloud server, the owner is losing the physical control over the data and thus the cloud server will obtain the access on the plain text data and it will make lots of security challenges to the PHR privacy and confidentiality.

The encryption of data before outsourcing it to the third party is consider as the promising approach towards data security and confidentiality towards the third party storage. Privacy threats experienced by users of services offered by Apple Inc. Google Inc., Amazon Inc are clear indications that cloud is intrinsically insecure from a user’s view point. Because users don’t have access to cloud service providers internal operations preserving privacy of user in cloud environment is a challenge for researchers. Cloud computing services benefit from economies of scale achieved through versatile use of resources, specialization, and other efficiencies. The Internet has grown into a world of its own, and its huge space now offers capabilities that could support Physicians in their duties in numerous ways. In recent years, is an emerging trend and PHR is a patient-centric model of health information exchange and management. Generally, PHR service allows a user 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.

In this paper, we endeavor to study the patient centric, secure sharing of PHRs stored on semi-trusted servers, and focus on addressing the complicated and challenging key management issues. In order to protect the personal health data stored on a semi-trusted 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 non-trivial to solve, and remain largely open up-to-date. To this end, we make the following main contributions:

- We propose a novel 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. In particular, the majority professional users are managed distributive 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 emergency scenarios.

- In the public domain, we 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.

- 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. Furthermore, 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. MULTI ATTRIBUTE BASED TECHNIQUE

In the prevailing system, a user’s identity must be validated by the authority, in distributed system; it is a complex task to manage numerous user identities. Also, all users must trust the central authority, if the authority is malicious; he can impersonate any user without being detected. Hence we are facing a major issue with Key-Escrow problem. The secret key is generated in a single place. In turn, the system can be easily attacked by attacking the single space. Keys were generated randomly and it is decided by the key generation center and the user doesn’t have any control/preferences or specification of deciding the key based on user centric purpose. The key generation center (KGC) can decrypt every cipher text addressed to specific users by generating their attribute keys. This could be a potential risk to the data confidentiality or privacy in the data sharing systems. The revocation of any attribute or any single user in an attribute group would affect all users in the group.

Most of the existing ABE (Attribute-based encryption) schemes are constructed on the architecture where a single trusted authority or KGC has the power to generate the whole private keys of users with its master secret information. Thus, the key escrow problem is inherent such that the KGC can decrypt every cipher text addressed to users in the system by generating their secret keys at any time. The major drawbacks of the prevailing system is that the data sharing is not much secure and any other user can easily access the data in data store. In addition to it, the system won’t distribute the data based on the attributes of the user. Hence in the proposed system, the key issuing protocol generates and issues user secret keys by using the multiple attributes obtained from the user. The proposed scheme delegates most laborious tasks of membership management to the data center. The major advantage of the proposed system is that the data is shared between the data owner and the users based on the attributes.

The user will be allowed to specify their details in which the keys will be generated using the multiple attributes of the user (Fig 1). Now the key generation system will generate the keys and provide them to the user. Key is used to access the application center and for accessing the confidential data. After the user makes a successful validation to the system, user will get access with their confidential data from the data center. The accessing of the data is done. The user will be allowed to define their attributes in which the user’s key will be generated. After the key generation, the user will be provided with keys.
Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-Based Encryption

When the user’s key matches, then the data will get decrypted and the user can view their confidential data. The user’s parameters params and a security parameter l are taken as input during the process of new user creation. Based on the parameters obtained from the user, an unique key is generated. The keys are generated using the multiple attributes of the user and are stored in a data center. The Keys that are generated are distributed to the user. The corresponding keys of the user are used to perform encryption before it is stored in the database. During the process of verifying credentials or process of login, the security parameter l and the input parameter params is authenticated. If the result of authentication is 1, the user can make use of corresponding key to view the original data.

Fig.1. Architecture Diagram.

A. Cipher text Policy Attribute-Based Encryption (CB-ABE)

In a cipher text policy attribute-based encryption scheme, each user’s key is related with a set of attributes representing their capabilities, and a cipher text is encrypted.

B. MD 5

The MD5 Message-Digest Algorithm is a widely used cryptographic hash function that produces a 128-bit (16-byte) hash value. Takes as input a message of arbitrary length and produces as output a 128 bit “message digest” of the input. MD5 has been employed in a wide variety of security applications, commonly used to check data integrity.

C. Triple DES Algorithm

Triple DES Algorithm is the common name for the Triple Data Encryption Algorithm (TDEA or Triple DEA) block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. The original DES cipher's key size of 56 bits was generally sufficient when that algorithm was designed, but the availability of increasing computational power made brute-force attacks feasible. Triple DES provides a relatively simple method of increasing the key size of DES to protect against such attacks, without the need to design a completely new block cipher algorithm.

D. Ant Colony Optimization Algorithm

The basic philosophy of the algorithm involves the movement of a colony of ants through the different states of the problem influenced by two local decision policies. In our project, we are trying to identify/predict and assign the possible access to be provided to the particular user based on user details.

E. Multi Attribute Based Technique in Key Generation System

Commit: (II) The Commitment scheme consists of the following three algorithms C= (Setup, Commit, Decommit) output params as the systems output parameter.

Decommit: This algorithm takes as input the parameters param, a message M and a commitment comm. The de-commitment decom is used to de-commit the commitment comm.

The KP-ABE consists of a set of parties {P_1, P_2, …, P_N}. An access structure (resp., monotonic access structure) is a collection (resp., monotonic collection) A of non-empty subsets of {P_1, P_2, …, P_N}, namely A \subseteq 2^{\{P_1,P_2,\ldots,P_N\} \setminus \{\phi\}}. The sets in A are called the authorized sets, and the sets outside of A are called the unauthorized sets.

III. CIPHERTEXT POLICY ATTRIBUTE BASED ENCRYPTION (CP-ABE)

Cipher text Policy Attribute based Encryption (CP-ABE), similar with role-based access control system, can be widely applied to realize access control in many applications including medical systems and education systems. For example, the sensitive medical records, tightly related to patients’ privacy, must be accessed only if the users are authorized with patients’ consent; solutions of exams in the education online system also should be only read by professors or specified teaching assistants. The CP-ABE scheme deals with those situations, by encrypting the target information with expressive access policies, such as “Medical” and “Physician”, “Professor” or (“Computer Science” and “Teaching Assistant”). In fact, CP-ABE can provide a perfect solution to an access control system by considering, efficient distributing, expressive access control and data confidentiality. In the traditional CP-ABE scheme, once users obtain the credentials from a system manager at the beginning of setup phase, the access ability is always valid for those who may even break the confidential rules by abusing this private information. Upon detecting those malicious adversaries, without any revocation mechanism
embedded, the system manager has to rebuild up the whole system. Therefore, revocation mechanism should be designed into the system from the beginning rather than being added after the other issues are addressed, as it requires careful planning on where functionality should be placed and how to reduce the computational and communication costs. In this paper, we aim at developing the CP-ABE scheme with efficient revocation.

Designing a revocation mechanism for CP-ABE is not a simple task while considering the following aspects: first, system manager only associates user secret keys with different sets of attributes instead of individual characteristics. The fuzzy identities therefore encumber the system’s revocation on one specified user; second, users’ individuality are taken place by several common attributes, and thus revocation on attributes or attribute sets can not accurately exclude the users with misbehaviors; third, the system must be secure against collusion attack from revoked users even though they share some common attributes with non-revoked users. To consider the revocation problem in a traditional CPABE scheme, limited choices are available. One is the revocation of a single attribute, which is not in connection with users’ behaviors but more likely to be periodical update of universal attribute set of the whole system. Another possible solution is to revoke one attribute set corresponding to one specific set of users. In this way, all the users’ access abilities will be revoked if they share the same attribute set with the malicious user, which is inappropriate in the real application.

As a solution to accurately and efficiently revoke the users’ access abilities, we modify the traditional CP-ABE model to CP-ABE-R (Cipher text Policy Attribute Based Encryption with Revocation) model in which each user is identified by a unique identifier. However, the encryption and decryption algorithms are completed without the involvement of these unique identifiers. The system manager assigns user secret keys along different paths in the revocation tree according to different unique identifiers, and publishes the revocation information according to a time stamp. A sender, without any knowledge of the receivers’ unique identifiers, encrypts the data with an access policy and a time stamp initialized by the system manager. The update information is the primary trigger controlling the user’s access ability. If the user’s unique identifier is not in the revocation list during he/she possesses the access ability corresponding to his/her attribute set; otherwise, he/she would be deprived of all access abilities. Therefore, this CP-ABE-R model leverages the expressive access control ability with accurate and efficient revocation.

In our construction, the binary tree technique is adopted to reduce communication and computational costs during the update phase. To avoid the need for maintaining secure channels between the system manager and the non-revoked users, the update information is generated corresponding to the key update nodes, a minimum set providing enough information to the non-revoked users and no useful information to the revoked users. In our design, a user secret key consists of three components: a unique secret key \( k \) mixed with personalized factors including attribute related personalized factor \( t \) and revocation related personalized factor \( t' \), as shown to Fig.2 the factor \( t \) and \( t' \) are two independent random variables among different users, and those personalized factors are key components to resist collusion attack, which will be elaborated. In our decryption algorithm, non-revoked users select the available update information to depersonalize the factor \( t' \), extract the access structure components in cipher text to depersonalize the factor \( t \), and eventually use unique secret key \( k \) to decrypt the message. To the best of our knowledge, this work is the first attempts to address the revocation issue in cipher text policy attribute based encryption.

![Fig.2. The components of secret keys.](image)

### IV. SCALABILITY AND EFFICIENCY

#### A. Storage and Communication Costs

First, we evaluate the scalability and efficiency of our solution in terms of storage, communication and computation costs. We compare with previous schemes in terms of cipher text size, user secret key size, public key/information size, and revocation (re-keying) message size. Our analysis is based on the worst case where each user may potentially access part of every owners’ data. Table 1 is a list of notations, where in our scheme: \[ |U| = |U_d| + |U_R|, \] \[ tc = |A^C_{PSD}| + |A^C_{PUD}| \] (includes one emergency attribute), and \[ tu = |A^u_{PSD}| + |A^u_{PUD}| \] (a user could be both in a PSD and PUD). Note that, since the HN, NGS and RNS schemes do not separate PSD and PUD, their \[ |U| = |U_R|, \] \[ tc = |A^C_{PUD}|, \] and \[ tu = |A^u_{PUD}|. \] However, they only apply to PHR access in the PUD. In addition, \[ S_P \sim O(t^2) \] in the RNS scheme, while \[ S_P \sim O(t\log t) \] for the rest. The results are given in Table 2. The cipher text size only accounts for the encryption of FEK. In our scheme, for simplicity we assume there is only one PUD, thus the cipher text includes \( n \) additional wildcard attributes and up to \( N - 1 \) dummy attributes. Our scheme requires a secret key size that is linear with \( |A^n| \), the number of attributes of each user, while in the VFJPS and BCHL schemes this is linear with \( N^n \).
Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-Based Encryption

since a user needs to obtain at least one key from each owner who’s 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 O (N_u) per owner. This puts a large burden either in communication or storage cost on the system. For re-keying, we consider revocation of one user by an owner in VFJPS and BCHL.

TABLE I: Notations for Efficiency Comparison

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_k</td>
<td>Bit size of a FEK</td>
</tr>
<tr>
<td>S_1</td>
<td>Bit size of an element in G_1/G_2</td>
</tr>
<tr>
<td>S_T</td>
<td>Bit size of an element in G_T</td>
</tr>
<tr>
<td>S_z</td>
<td>Bit size of an element in Z_p</td>
</tr>
<tr>
<td>S_P</td>
<td>Bit size of access policy and attribute set in CT</td>
</tr>
<tr>
<td>N (or N_l)</td>
<td>Number of AAs in a PUD (or the l-th PUD)</td>
</tr>
<tr>
<td>N_o</td>
<td>The number of owners in the system</td>
</tr>
<tr>
<td>N_d</td>
<td>The number of data users in the system</td>
</tr>
<tr>
<td>N_r</td>
<td>The number of revoked users for a file</td>
</tr>
<tr>
<td>m</td>
<td>Number of attributes in the PUD</td>
</tr>
<tr>
<td>t_s, t_u</td>
<td>Total number of attributes appeared in CT, s_k_u</td>
</tr>
<tr>
<td>l</td>
<td>Depth of file hierarchy of an owner’s PHR</td>
</tr>
</tbody>
</table>

In VFJPS, revoking one user from a file may need over-encryption 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 re-keying and re-encryption; however, attribute revocation is not achieved; and for the revocable ABBE, either the cipher text size is linear with the number of revoked users, or the public key is linear with the total number of users in the system. For the RNS scheme, the main drawback is the large size of revocation messages to be transmitted to non-revoked users. In our scheme, revocation of one user requires revoking a minimum set of data attributes that makes her access structure un-satisfiable. From Table 2, it can be seen that our scheme has much smaller key size compared with VFJPS and BCHL, smaller rekeying message size than VFJPS, and RNS, the size of cipher text is smaller than NGS while being comparable with BCS. The public key size is smaller than VFJPS and BCHL, and is comparable with that of RNS; while it seems larger than those of NGS and BCS note that we can use the large universe constructions to dramatically reduce the public key size. Overall, compared with non-ABE schemes, our scheme achieves higher scalability in key management. Compared with existing revocable ABE schemes, the main advantage of our solution is small re-keying message sizes.

To revoke a user, the maximum re-keying message size is linear with the number of attributes in that user’s secret key. 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.

B. 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 (to the best of our knowledge), and also integrated the ABE algorithms into a prototype PHR system. The GPSW KP-ABE scheme is tested on a PC with 3.4 GHz processor, using the pairing based cryptography (PBC) library. The public parameters are chosen to provide 80 bits security level, and we use a pairing-friendly type-A 160-bit elliptic curve group. This parameter setting has also been adopted in other related works in ABE. 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 implementation are reported. In the supplementary material, 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 |A^C_{PSD}| + 1 pairing operations (in contrast, the RNS scheme involves 2|A^C_{PSD}| + 1 pairing operations). The time costs of key generation, encryption and decryption processes are all linear with the number of attributes. For 50 attributes, they all take less than 0.5s.

TABLE III: Computation Complexity For Each Party In The System, And Numerical Estimation Of Time Costs Assuming Following Parameters (Also Used In Supplementary Material): |U| = 50, |U_R| = 100, N = 5 (number of AAs), |A^C_{PSD}| = 5, |A^C_{PSD}| = 35, |A^C_{PSD}| = m = 15, |L (T)| = 10, |n^*| = 5 (a Minimal Number Of Attributes To Revoke A User).

From the system aspect, each data owner (patient) uses the YWRL ABE scheme for setup, key generation and revocation, uses both YWRL and enhanced MAABE for encryption. Each PSD user adopts the YWRL scheme for decryption, while each PUD user adopts the enhanced MAABE scheme for decryption. Each AA uses enhanced MAABE for setup, key generation and revocation. Next we
provide estimations of computation times of each party in the system in Table 3. The values are calculated from the example parameters and benchmark results, where exponentiation times $\text{Exp}_p = 6.4 \text{ms}$, $\text{Exp}_T = 0.6 \text{ms}$, pairing time $\text{TP} = 2.5 \text{ms}$. 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 cipher text/key update operations, which amortizes the computations over time. The details of the experimental/simulation evaluation results are presented in the supplementary material.

**V. CONCLUSION**

In this paper, we have proposed a novel framework of secure sharing of personal health records in cloud computing. The personal health records are now considered as the emerging trend in the personal health information exchange field. And cloud computing storage and sharing service is highly utilized by the users. Cloud computing is increasingly used by healthcare service providers. Privacy is major issue while outsourcing healthcare data on cloud. The data security is the main privacy issue and the attribute based encryptions and its variations are applied for this security purpose. This paper supports efficient on-demand revocation using the CP-ABE technique. We use Attribute Based Encryption to encipher the Personal Health Record data, hence that patients can permit access not only by personal users, but also many users from public domains with different professional roles, affiliations and qualifications. In addition, we enhance an existing Multi Authority Attribute Based Encryption scheme to manage on-demand user revocation, efficient, and prove its security. Through implementation and simulation, we show that our solution is both scalable and efficient.

**VI. REFERENCES**


**Author’s Profile:**

CHITRAM RAMESH, PG Scholar,
Dept of CSE, Jawaharlal Nehru Institute of Technology, Hyderabad, India, E-mail:ramesh5545@gmail.com.

D.KRISHNA, HOD & Associate Professor, Dept of CSE, Jawaharlal Nehru Institute of Technology, Hyderabad, India.

G.DEEPTHI Assistant Professor, Dept of CSE, Jawaharlal Nehru Institute of Technology, Hyderabad, India.