Secure and Efficient Way of Handling Medical Records in Cloud

M. Devi Mareeswari[1], A. Shenbaga Bharatha Priya[2] and J.Ganesh[3]

[1]M-Tech(IT), Anna University, Chennai, Dr.Sivanthi Adhithanar College of Engineering, Tiruchendur,Tuticorin-628215, Tamil Nadu,India.

[2]M-Tech(IT), Anna University, Chennai, Dr.Sivanthi Adhithanar College of Engineering, Tiruchendur,Tuticorin-628215, Tamil Nadu,India

[3]Department of IT, Anna University, Chennai, Dr.Sivanthi Adhithanar College of Engineering, Tiruchendur, Tuticorin-628215, Tamil Nadu,India

Abstract

Personal health records (PHRs) are touted as a new convenience technology for consumers. It enables the patients to create a health information of their own in a centralized way, which alleviate the storage, access and sharing of health data in the cloud environment. By storing the health information in the cloud various security issues should arise such us authorization, key management and efficient user revocation, therefore, before outsourcing the PHR in the cloud, it is a promising method to encrypt the PHR using Attribute Based Encryption. Existing cryptographic schemes are planned for single owner settings, here, dealt with multiple owner scenarios which reduce the key management complexity for owners and users. Enhancing the MA-ABE scheme to handle efficient and on-demand user revocation, and prove its security. The experimental results will show the security and efficiency of the proposed system.

Keywords: *Virtualization, ,Personal health record, HealthCare Social Network., Attribute Based Encryption*

1. 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[1] 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 Health Vault. 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[1] of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. A feasible and promising approach would be to encrypt [3] 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 [2] 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.

2. Background

2.1Types of cloud

There are four main type of cloud:
1) **Public cloud**: The cloud computing resource is shared outside, anyone can use it and some payment maybe need.
2) **Private cloud**: It is opposite to public cloud, private cloud’s resource is limit to a group of people, like a staff of a company etc.
3) **Hybrid cloud**: this is a mixture of previous two clouds, some cloud computing resource is shared outside but some don’t.
4) **Community cloud**: this is a special cloud to make use of cloud computing features. More than one community shares a cloud to share and reduce the cost of computing system.

2.2 Virtualization

Virtualization technology lets a single PC or server simultaneously run multiple operating systems or multiple sessions of a single OS. This lets users put numerous applications even those that run on different operating systems on a single PC or server instead of having to host them on separate machines as in the past. The approach is thus becoming a common way for businesses and individuals to optimize their hardware usage by maximizing the number and kinds of jobs a single CPU can handle.

2.3Cryptographic Technique

2.3.1Attribute-based Encryption

It is a type of public key Encryption, in which the secret key of a user and the cipher text are dependent about attributes. In a such as a System the decryption [4] of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text.

2.3.2Multi-Authority Attribute Based Encryption

In a desired Multi-Authority CP-ABE(MA-CP-ABE) [4] system,different domains of attributes are managed by different authorities.An encryptor can encrypt messages with any access policy over the entire attribute universe.

2.3.3key-Policy Attribute based Encryption

KP-ABE is a public key cryptography primitive for one-to-many communications.In KP-ABE [7] data are associated with attributes for each of which a public key component is defined.The encrypt are associates the set of attributes to the message by encrypting it with the corresponding public key components.

2.3.4Ciphertext-Policy AttributeBased Encryption

CP-ABE is a tool for implementing fine-grained access control [8] over encrypted data, and is conceptually similar to traditional access control methods such as Role-Based Access Control[21].

3. Related Works

In the past, health care providers (such as the family doctor) have stored medical records of their patients on paper locally. This allowed a controlled environment with easy management of data privacy and security: keeping the paper records in a locked cabin at the doctor's practice. Even the increasing use of personal computers and modern information technology in medical institutions allowed for a moderate effort to manage privacy and confidentiality of individual medical records. This was due to the decentralized and locally managed infrastructure of each institution. But nowadays outsourcing [6] of IT infrastructure (e.g., cloud computing) and other services (e.g., billing processing and accounting for medical practices) leads to a complex system where privacy-sensitive data are stored and processed at many different places. Hence, it becomes attractive to store and process healthcare data in the cloud (at outsourced data providers that can be accessed via the Internet). While such e-health systems promise a more cost-efficient service and improved service quality, the complexity to manage data security and privacy increases, too.

In existing system, the PHRs are stored on a server of a third party in the cloud. The PHR server provider is responsible for ensuring data protection. Typically, patients can define role-based access rights [3] for individual health professionals.

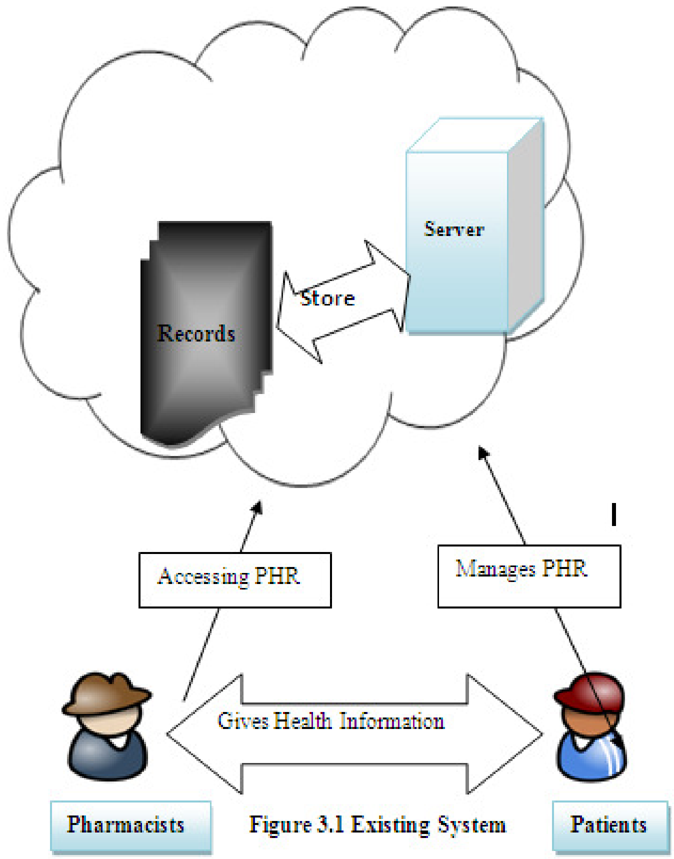


Figure 3.1 Existing System

For example, they can define full access to their family doctor, but only restricted access to some data to their fitness trainer or health coach. The advantages of such an approach are that the PHR is accessible from everywhere because of the centralized management [10] (IT outsourcing). The patient can easily give one doctor access to data and test results that were determined by another doctor, when the data is stored in the PHR. This can help to avoid double examination. Moreover, due to the individual management of PHRs by the patients, it is expected that people are more aware of their own health. This could reduce the healthcare costs in the long term as well. However, from a technical perspective this model has a great disadvantage regarding patients' privacy. On the one hand, patients need to manage complex access rights and need to understand their implications. On the other hand, they need to rely on the robustness and correctness of the security mechanisms implemented at the PHR server provider. In general, it may be possible for the server provider to gain access to the data stored in PHRs.

4. Proposed Works

4.1Considerations

Personal health record is an model for exchanging the health information which is outsourced to be stored at a third party, such as cloud providers. 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. A feasible and promising approach would be to encrypt the data before outsourcing[20]. 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. Yet, issues such as risks of privacy exposure, scalability in key management, flexible access, and efficient user revocation, have remained the most important challenges toward achieving fine-grained, cryptographically enforced data access control. The proposed framework describes patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semi trusted servers. To achieve fine-grained data access control for PHRs, we leverage attribute-based encryption (ABE) techniques to encrypt each patient's PHR file.

4.2 Architecture

4.2.1 Patient Centric Framework

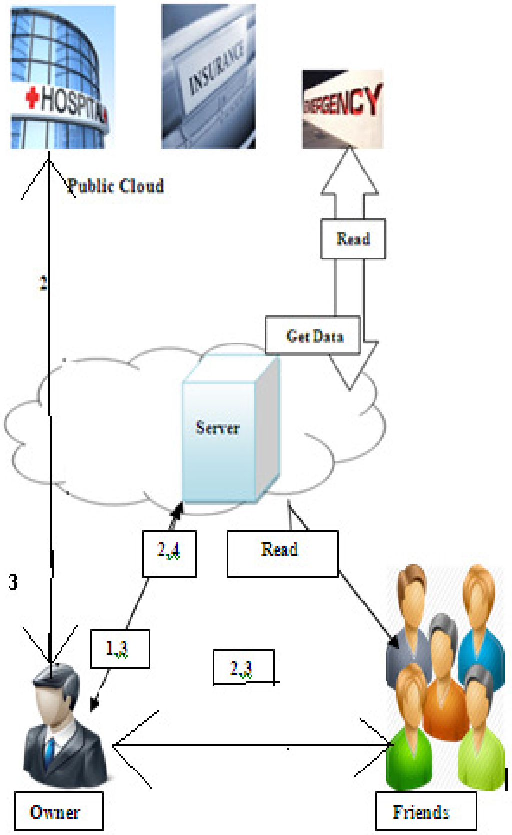
The main goal of our framework is to providesecure patient-centric PHR access and efficient key management at the same time. The key idea is to divide the system into multiple security domains (namely, public domains (PUDs) and personal domains (PSDs)) according to the different users' data access requirements. The PUDs consist of users who make access based on their professional roles, such as doctors, nurses and medical researchers. In practice, a PUD [5] 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. 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[19], 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 [11], it greatly reduces the key management overhead for both the owners and users.

4.2.2 Key Distribution- PHR Encryption and 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) [12] (which could be part of the PHR service. There are two ways for distributing secret keys [9]. 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, Second, a reader in PSD could obtain the secret key by sending a request (indicating which types of files she wants to access) , 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 [15] 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, when the user is granted all the file types under a category, her access privilege will be represented by that category instead. 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[13], excluding the server. For improving efficiency, the data attributes will include all the intermediate file types from a leaf node to the root. The data readers download PHR files from the server, and they can decrypt the files only if they have suitable attribute based keys. The data contributors will be granted write access to someone's PHR, if they present proper write keys.

4.2.3 ABE for Fine-grained Data Access Control

In this module ABE to realize fine-grained [14] access control for outsourced data Especially, there has been an increasing interest in applying ABE to secure electronic healthcare records (EHRs). An attribute-based infrastructure for EHR systems[18], where each patient's EHR files are encrypted using a broadcast variant of CP-ABE that allows direct revocation. However, the cipher text length grows linearly with the number of un revoked users. In a variant of ABE that allows delegation of access rights is proposed for encrypted EHRs applied cipher text policy ABE (CP-ABE) [16] to manage the sharing of PHRs, and introduced the concept of social/professional domains 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.



4.1 Proposed System

1.Stores data in Cloud.
2.Get Attributes.
3.Write Access.
4.Revoke.

4.2.4 Break-glass module

In this module 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 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[17]. After the Emergency is over; the patient can revoke the emergent access via the ED.

4.3Technical Layout

4.3.1Introducing Web Application

Organizations are increasingly becoming dependent on the Internet for sharing and accessing information. This Internet boom has changed the focus of application development from stand-alone applications to distributed Web applications. Web applications are programs that can be executed either on a web server or in a web browser. They enable you to share and access information over the Internet and operate intranets.

4.3.2Introduction to ASP.NET

ASP.NET is a part of the .NET Framework, a new computing platform from Microsoft optimized for creating applications that are highly distributed across the Internet. Highly distributed, here means that the components of the applications, as well as the data, may reside anywhere in the Internet rather than all being contained inside one software program somewhere. It is a programming framework, and one of the primary differences between it and traditional ASP is that it uses a common language runtime (CLR) capable of running compiled code on a web server to deploy powerful wed-based applications.

ASP.NET still use HTTP to communicate to the browser and back, but it brings added functionality that makes the communication process much richer. If any files have the appropriate extension or contain code, the server routes those files to ASP.NET for processing prior to sending them out to the client. The script or code is then processed and the appropriate content is generated for transmission back to the browser/client. Because processing takes place before the results are delivered to the user, all manner of functionality can be built-in such as database access, component usage and the ordinary programmatic functionality available with scripting languages. Microsoft has introduced ASP. ASP.NET is the .NET version of ASP. ASP.NET is a standard HTML file that contains embedded server-side scripts.

4.3.3ASP.NET in .NET Framework

ASP.NET, which is the .NET version of ASP, is built on Microsoft .NET Framework. Microsoft introduced the .NET Framework to help developers create globally distributed software with Internet functionality and interoperability. The elements of an ASP.NET application include Web service to provide a mechanism for programs to communicate over the Internet.

4.3.4 ADO.NET

ADO.NET is all about data access. Data is generally stored in a relational database in the form of related tables. Retrieving and manipulating data directly from a database requires the knowledge of database commands to access the data.

5. Results

The ScreenShots of our project is captured and listed below.

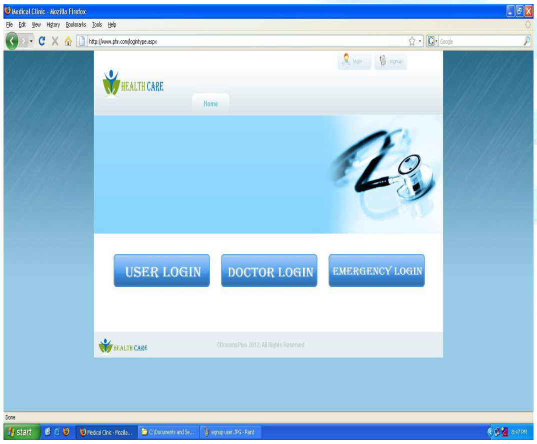
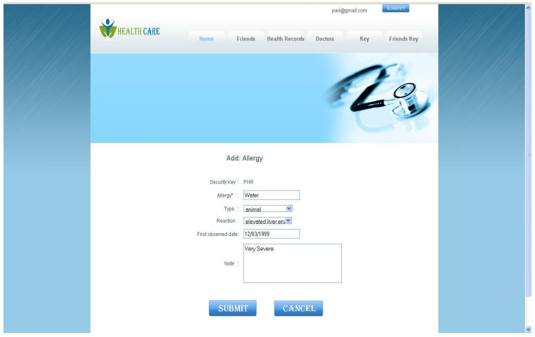


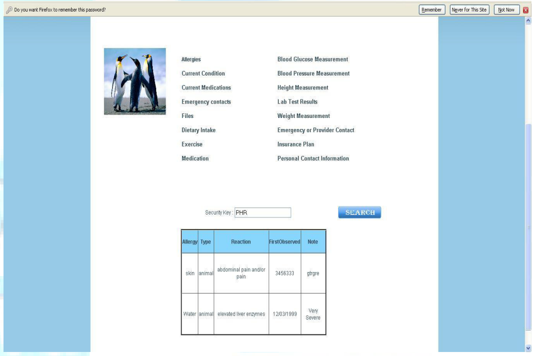
Fig 5.1 Various Login Form



Add: Allergy

Security Key :	PHR
Allergy* :	Water
Type :	animal
Reaction :	elevated liver en:
First observed date:	12/03/1999
Note :	Very Severe

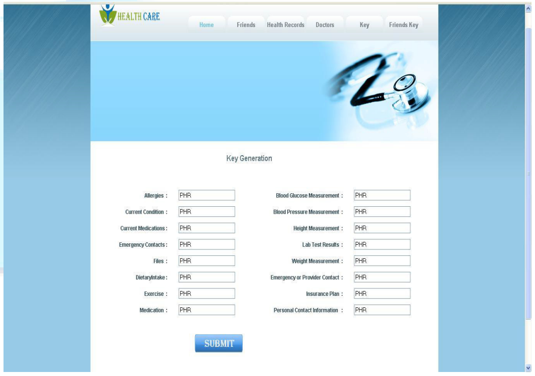
Fig 5.2 Adding medical Detail



Security Key : PHR

Allergy	Type	Reaction	FirstObserved	Note
skin	animal	abdominal pain and/or pain	3456333	gtrgre
Water	animal	elevated liver enzymes	12/03/1999	Very Severe

Fig 5.3 A doctor viewing a patient's information



Key Generation

Allergies :	PHR	Blood Glucose Measurement :	PHR
Current Condition :	PHR	Blood Pressure Measurement :	PHR
Current Medications :	PHR	Height Measurement :	PHR
Emergency Contacts :	PHR	Lab Test Results :	PHR
Files :	PHR	Weight Measurement :	PHR
DietaryIntake :	PHR	Emergency or Provider Contact :	PHR
Exercise :	PHR	Insurance Plan :	PHR
Medication :	PHR	Personal Contact Information :	PHR

Fig 5.4 Key Generation

6. Conclusion and Future Enhancements

6.1Conclusions

In this paper, we have proposed a novel framework of secure sharing of personal health records in cloud computing. 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. Furthermore, we enhance an existing MA-ABE scheme to handle efficient and on-demand user revocation, and prove its security.

6.2 Future Enhancements

As future study, it will be interesting to enhance the HSN with a third party auditor to verify the cloud server that stores and process the PHRs. Homomorphic Split key Encryption can become additional enhancement to verify the trustworthiness of the TPA.

References

[1] 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.

[2] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, “Proposed NIST standard for role-based access control,” *ACM TISSEC.*, vol. 4, no.3,pp. 224–274, 2001.

[3] C. Dong, G. Russello, and N. Dulay, “Shared and Searchable Encrypted Data for Untrusted Servers,” J. Computer Security,vol. 19, pp. 37-397, 2010.

[4] 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.

[5] 104th United States Congress, “Health Insurance Portability and Accountability Act of 1996 (HIPPA),” Online at http://aspe.hhs.gov/admnsimp/pl104191.htm, 1996.

[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” in *Proc. of NDSS’05*, 2005.

[7] 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.

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

[9] 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.

[10] M. J. Atallah, K. B. Frikken, and M. Blanton, “Dynamic and efficient key management
for access hierarchies,” in *CCS '05*, 2005, pp. 190–202.

[11] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, “Over-encryption: Management of access control evolution on outsourced data,” in *Proc. of VLDB’07*, 2007.

[12] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,”
in *Proc. of ESORICS ’09*, 2009.

[13] S. Yu, K. Ren, W. Lou, and J. Li, “Defending against key abuse attacks in kp-abe enabled broadcast systems,” in *Proc. of SECURECOMM’09*,2009.

[14] Z. Yang, S. Zhong, and R. N. Wright. Privacy-preserving queries on encrypted data. In D. Gollmann, J. Meier, and A. Sabelfeld, editors, ESORICS, volume 4189 of Lecture Notes in Computer Science, pages 479–495. Springer, 2006.

[15] Hugh Harney, Andrea Colgrove, and Patrick Drew McDaniel. Principles of policy in secure groups. In *NDSS*, 2001.

[16] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 466{481. Springer, 2002.

[17] M. Ito, A. Saito, and T. Nishizeki. Secret Sharing Scheme Realizing General Access Structure. In *IEEE Globecom*. IEEE, 1987.

[18] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters. Secure Atrribute-Based Systems.In *ACM conference on Computer and Communications Security (ACM CCS)*, 2006.To appear.

[19] Nigel P. Smart. Access control using pairing based cryptography. In *CT-RSA*, pages 111{121, 2003.

[20] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In TCC, pages 535{554, 2007.

[21] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In EUROCRYPT, pages 506{522, 2004.