ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 4, Issue 11, November 2016**

# ABE for Scalable and Secure Sharing of Personal Health Records in Cloud Computing

Shriram Vetal[1], Prof. Zine Datta[2], Prof. Vrushali Gaike Ranmalkar[3]

Research Scholar, Dept. of Computer Engineering, Vishwabharti Academy's Collage of Engineering, A.nagar, Savitribai Phule University of Pune, India[1]

HOD, Dept. of Computer Engineering, Vishwabharti Academy's Collage of Engineering, A.nagar, Savitribai Phule University of Pune, India[2]

Assistant Professor, Dept. of Computer Engineering, Vishwabharti Academy's Collage of Engineering, A.nagar, Savitribai Phule University of Pune, India[3]

**ABSTRACT:** In the recent few years, the usage of the cloud computing has been amazingly increasing among verity of areas, such as e-business, social networks, educations..etc. That transmission is mainly due to valuable and outstanding features that this technology offers. The healthcare industry is continuously developing as well as the health data .Therefore, the need for more storage to keep the data, smart techniques to handle them and remotely access to them is becoming more and more sufficient. Cloud computing give a promising solution to the healthcare industry with it web-based recourse, the huge storage available and expert maintenance. By improving some of the aspects in the health data privacy and more efficient data back-up. Cloud Computing has a promising future to be more and more adapted and intergraded in the healthcare industries. Cloud computing technology provide an almost excellence solution to improve the healthcare industries by decreasing the cost and increasing the usability and reliability

**KEYWORDS**: Personal health records, cloud computing, data privacy, attribute-based encryption

## I. INTRODUCTION

Cloud computing is an internet base computing. It is a model of computing that provides on-demand sources as services over the internet. The implementation details are hidden from the consumer who only concerned about the output services. In the cloud computing instead of owning the physical infrastructure, consumer rent usage from cloud computing service providers. Therefore, consumer will avoid the expenses associated with the physical infrastructure.

Cloud computing is a promising computing paradigm which recently has drawn extensive attention from both academia and industry. By combining a set of existing and new techniques from research areas such as Service-Oriented Architectures (SOA) and virtualization, cloud computing is regarded as such a computing paradigm in which resources in the computing infrastructure are provided as services over the Internet. Along with this new paradigm, various business models are developed[1].

Online personal health record (PHR) enables patients to manage their own medical records in a centralized way, which greatly facilitates the storage, access and sharing of personal health data. With the emergence of cloud computing, it is attractive for the PHR service providers to shift their PHR applications and storage into the cloud, in order to enjoy the elastic resources and reduce the operational cost. However, by storing PHRs in the cloud, the patients lose physical control to their personal health data, which makes it necessary for each patient to encrypt her PHR data before uploading to the cloud servers. Under encryption, it is challenging to achieve fine-grained access control to PHR data in a scalable and efficient way[2].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 4, Issue 11, November 2016**

## II. RELATED WORK

**Traditional Access Control for EHRs**

Traditionally, research on access control in electronic health records (EHRs) often places full trust on the health care providers where the EHR data are often resided in, and the access policies are implemented and enforced by the health providers. Various access control models have been proposed and applied, including role-based (RBAC) and attribute-based access control (ABAC). Each user ' s access right is determined based on his/her roles and the role-specific privileges associated with them. The ABAC extends the role concept in RBAC to attributes, such as properties of the resource, entities, and the environment. Compared with RBAC, the ABAC is more favorable in the context of health care due to its potential flexibility in policy descriptions. A line of research aims at improving the expressiveness and flexibility of the access control policies. However, for personal health records (PHRs) in cloud computing environments, the PHR service providers may not be in the same trust domains with the patients ' . Thus patient-centric privacy is hard to guarantee when full trust is placed on the cloud servers, since the patients lose physical control to their sensitive data. Therefore, the PHR needs to be encrypted in a way that enforces each patient ' s personalized privacy policy, which is the focus of this paper.

**Cryptographically Enforced Access Control for Outsourced Data**

For access control of outsourced data, partially trusted servers are often assumed. With cryptographic techniques, the goal is trying to enforce that who has (read) access to which parts of a patient ' s PHR documents in a *fine-grained* way.

*Symmetric key cryptography (SKC) based solutions*. Vimercati *et.al*. Proposed a solution for securing outsourced data on semi-trusted servers based on symmetric key derivation methods, which can achieve fine-grained access control. Unfortunately, the complexities of file creation and user grant/revocation operations are linear to the number of authorized users, which is less scalable. Files in a PHR are organized by hierarchical categories in order to make key distribution more efficient. However, user revocation is not supported. An owner ' s data is encrypted block-by-block, and a binary key tree is constructed over the block keys to reduce the number of keys given to each user.

*Public key cryptography (PKC) based solutions*. PKC based solutions were proposed due to its ability to separate write and read privileges. Benaloh *et. al.* proposed a scheme based on hierarchical identity based encryption (HIBE), where each category label is regarded as an identity. However, it still has potentially high key management overhead. In order to deal with the multi-user scenarios in encrypted search, Dong *et.al*. Proposed a solution based on proxy encryption. Access control can be enforced if every write and read operation involves a proxy server. However, it does not support fine-grained access control, and is also not collusion-safe.

*Attribute-based encryption (ABE).* The SKC and traditional PKC based solutions all suffer from low scalability in a large PHR system, since file encryption is done in an one-to-one manner, while each PHR may have an unpredictable large number of users. To avoid such inconveniences, novel one-to-many encryption methods such as *attribute-based encryption* can be used. In the seminal paper on ABE, data is encrypted to a group of uses characterized by a set of attributes, which potentially makes the key management more efficient. Since then, several works used ABE to realize fine-grained access control for outsourced data. However, they have not addressed the multiple data owner settings, and there lacks a framework for patient-centric access control in multi-owner PHR systems. Note that, a single authority for all users and patients is adopted. However, this suffers from the key escrow problem, and patients ' privacy still cannot be guaranteed since the authority has keys for all owners. Recently Ibraimi *et.al*. applied ciphertext policy ABE (CP-ABE) to manage the sharing of PHRs. However, they still assume a single public authority, while the challenging key-management issues remain largely unsolved.

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

## III. PROPOSED FRAMEWORK

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

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.

Security Model

we consider the server to be semitrusted, i.e., honest but curious. 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.

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.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. There is also user revocation, where all of a user's access privileges are revoked.
- . Write access control. We shall prevent the unauthorized 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 unpredictable, 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.

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

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 patientcentric 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.

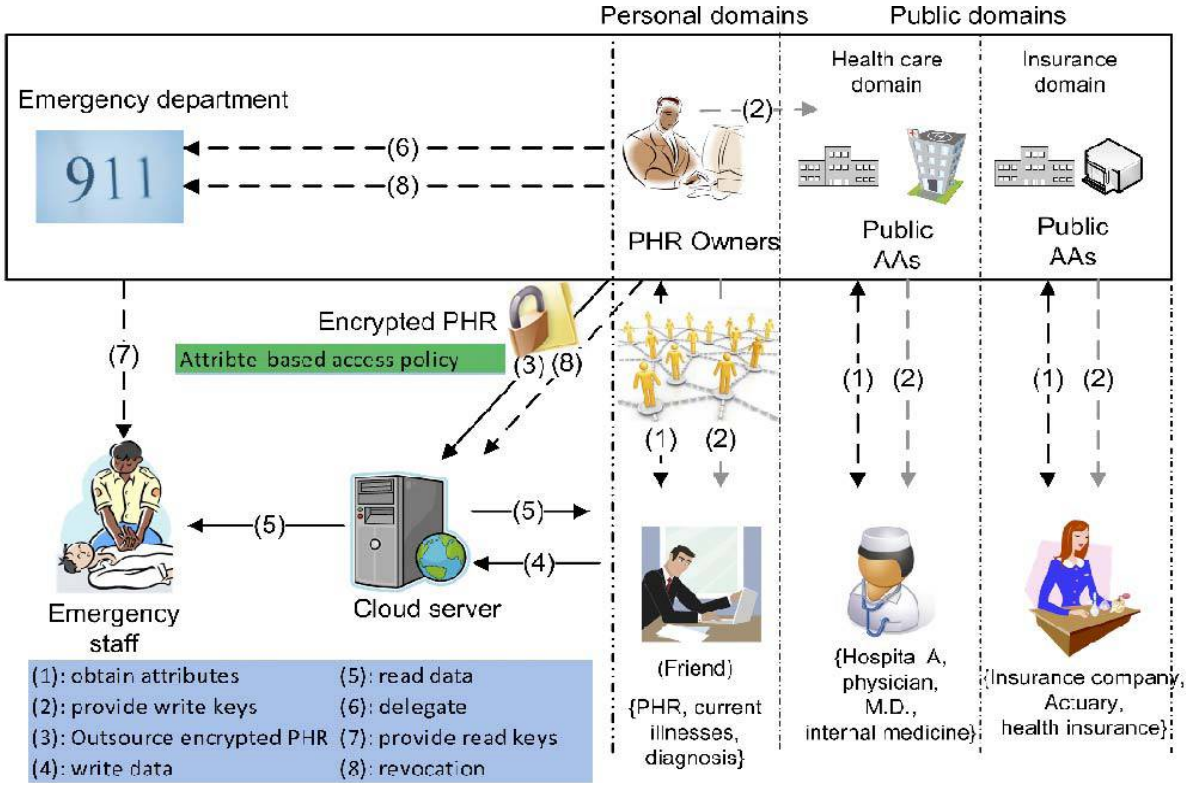


Fig. 1. The proposed framework for patient-centric, secure and scalable PHR sharing on semitrusted storage under multiowner settings.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

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 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.

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). 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.

PHR encryption and access:

The owners upload ABEencrypted PHR files to the server (3). Each owner's PHR file is encrypted both under a certain fine-grained and rolebased 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, in Fig. 2, 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:

1. revocation of one or more role attributes of a public domain user;
2. revocation of a public domain user which is equivalent 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).
3. Revocation of a personal domain user's access privileges;
4. 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

Using 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 distributively. 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 of flexibility, i.e., it functions similar to CP-ABE.

In order to allow the owners to specify an access policy for each PHR document, we exploit the fact that the basic CC MA-ABE works in a way similar to fuzzy-IBE, where the threshold policies (e.g., k out of n) are supported. Since the threshold gate has an intrinsic symmetry from both the encryptor and the user's point of views, we can predefine the formats of the allowed document policies as well as those of the key-policies, so that an owner can enforce a file access policy through choosing which set of attributes to be included in the ciphertext.

Enhancing MA-ABE for User Revocation

The original CC MA-ABE scheme does not enable efficient and on-demand user revocation. To achieve this for MAABE, we combine ideas from YWRL's revocable KP-ABE, and propose an enhanced MA-ABE scheme. In particular, an authority can revoke a user or user's attributes immediately by reencrypting the ciphertexts and updating users' secret keys, while a major part of these operations can be delegated to the server which enhances efficiency.

The idea to revoke one attribute of a user in MA-ABE is as follows: The AA who governs this attribute actively updates that attribute for all the affected unrevoked users. To this end, the following updates should be carried out: 1) the public/master key components for the affected attribute; (2) the secret key component corresponding to that attribute of each unrevoked user; 3) Also, the server shall update all the ciphertexts containing that attribute. In order to reduce the potential computational burden for the AAs, we adopt proxy encryption to delegate operations 2 and 3 to the server, and use lazy-revocation to reduce the overhead. In particular, each data attribute i is associated with a version number veri. Upon each revocation event, if I is an affected attribute, the AA submits a key to the server, who then reencrypts the affected ciphertexts and increases their version numbers. The unrevoked users' secret key components are updated via a similar operation using the rekey. To delegate secret key updates to the server, a dummy attribute needs to be additionally defined by each of N _ 1 AAs, which are always ANDed with each user's key-policy to prevent the server from grasping the secret keys. This also maintains the resistance against up to N _ 2 AA collusion of MA-ABE. Using lazy-revocation, the affected ciphertexts and user secret keys are only updated when an affected unrevoked user logs into the system next time. By the form of the rekey, all the updates can be aggregated from the last login to the most current one.

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 some temporary authorization (e.g., emergency key) to decrypt those data. Under our framework, this can be naturally achieved by letting each patient 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 ciphertext of each PHR document that she allows break-glass access. She then generates an emergency key skEM using the singlenode key-policy "emergency," and delegates it to the ED who keeps it in a database of patient directory. Upon emergency, a medical 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 4, Issue 11, November 2016**

Summary

In this above, we present a method to enforce owner's access policy during encryption, which utilizes the MAABE scheme in a way like CP-ABE. The essential idea is to define a set of key-generation rules and encryption rules. There are two layers in the encryptor's access policy; the first one is across different attribute authorities while the second is across different attributes governed by the same AA. For the first layer, conjunctive policy is enabled; for the second, either k-out-of-n or DNF policy are supported. We exploit the correlations among attribute types under an AA to enable the extended second-level DNF policy.

## IV. CONCLUSION AND FUTURE WORK

Proposed a novel framework for secure sharing of personal health records in cloud computing. Considering partially trustworthy cloud servers, saying 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 greatly reduced the complexity of key management while enhancement of the privacy guarantees compared with previous works. By utilizing ABE to encrypt the PHR data, patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications, and affiliations. Furthermore, enhancement of an existing MA-ABE scheme to handle efficient and on-demand user revocation, and prove its security. Through implementation and simulation, it shows that proposed solution is both scalable and efficient.

In this paper System is improved to support dynamic policy management model. Thus, Personal Health Records are maintained with security and privacy. In future, to provide high security and privacy for Personal Health Record (PHR), the existing Multi authority attribute based encryption could be further enhanced to proactive Multi authority attribute based encryption Data Confidentiality and Integrity is a major concern. We mainly concentrate on business cloud where various organizations store their data about their project in the cloud. We have analyzed the security of our algorithm and also the efficiency.

## REFERENCES

[1] Gaurav Berad, Prof. J.S. Raghatwan, Prof. MayurAknewar" Review on ABE for Scalable and Secure Sharing of Personal Health Records in Cloud Computing "IJIRCCE Vol. 3, Issue 12, December 2015.

[2] Shucheng Yu∗, Cong Wang†, Kui Ren† , and Wenjing Lou, " Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing", IEEE Communications Society IEEE INFOCOM 2010, Technical Program at IEEE INFOCOM 2010.

[3] Ming Li1, Shucheng Yu1, Kui Ren2, and Wenjing Lou1, " Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings", S. Jajodia and J. Zhou (Eds.): SecureComm 2010, LNICST 50, pp. 89–106, 2010. Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2010.

[4] 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.[4] Minu George, Dr. C.Suresh Gnanadhas, Saranya.K "A Survey on Attribute Based Encryption Scheme in Cloud Computing " IJARCCE Vol. 2, Issue 11, November 2013

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

[6] 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.

[7] 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.

## BIOGRAPHY

**Shriram Vetal** is a Research Scholar, Department of Computer Engineering, Vishwabharti Academy's Collage of Engineering,A.nagar, Savitribai Phule University of Pune, India. The research was supported by the senior faculty and Scholars of Vishwabharti Academy's Collage of Engineering,A.nagar Department of Computer Engineering.