Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 6, June 2014, pg.1 – 10

RESEARCH ARTICLE

A Novel Patient Centric Framework for Data Access Control in Semi-trusted Cloud Servers

Allamaprabhu G Rudraxi, Mr. Parikshith Nayak S.K
M.Tech Scholar, Dept. of CSE
Assistant Professor, Dept. of CSE
AIET, Mangalore –India
prabhurudraxi@gmail.com, **Pari2sn@gmail.com**

Abstract- Cloud storage is a model of networked online storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Personal health record (PHR) is an emerging patient-centric model of health information exchange, Issues such as risks of privacy exposure, scalability in key management, flexible access, and efficient user revocation, have the most important challenges so we propose a novel patient-centric framework and a suite of mechanisms for data access control to PHRs 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. This project also supports multiple owner scenarios and divides the users in the system into multiple security domains that greatly reduce the key management complexity for owners and users. We implement this system and evaluate atop DriveHQ cloud to demonstrate that our new system provides Secure Data Access over outsourced data.

Keywords: Personal health records, cloud storage, attribute based encryption

1. INTRODUCTION

Cloud computing is now the hot spot of computer business and research., as it offers an abstraction of infinite storage space for clients to host data backups in a pay-as you use manner. It helps enterprises and government agencies significantly reduce their financial overhead. Since they can now archive their data backups remotely to third-party cloud storage providers rather than maintain data centres on their own. Recent years, personal health record (PHR) has emerged as a patient-centric model of health information exchange. A PHR service allows a patient to create, manage, and control her personal health data in one place through the web, which has made the storage, retrieval, and sharing of the medical information more efficient. Especially, each patient is promised the full control of her medical records and

can share her health data with a wide range of users, including healthcare providers, family members or friends. Due to the high cost of building and maintaining specialized data centres, many PHR services are outsourced to or provided by third-party service providers, for example, Microsoft HealthVault. Architectures of storing PHRs in cloud computing have been proposed in [2], [3].

While it is exciting to have convenient PHR services for everyone, there are many security and privacy risks which could impede its wide adoption. The main concern is about whether the patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. On the one hand, although there exist healthcare regulations such as HIPAA which is recently amended to incorporate business associates, cloud providers are usually not covered entities [13]. 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 behaviours which may lead to exposure of the PHI. 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 should decide how to encrypt his/her files and to allow which set of users to obtain access to each file. A PHR file should only be available to the users who are given the corresponding decryption key, while remain confidential to the rest of users. Furthermore, the patient shall always retain the right to not only grant, but also revoke access privileges when they feel it is necessary [7]. However, the goal of patient-centric privacy is often in conflict with scalability in a PHR system. The authorized users may either need to access the PHR for personal use or professional purposes. Examples of the former are family member and friends, while the latter can be medical doctors, pharmacists, and researchers, etc. We refer to the two categories of users as personal and professional users, respectively. The latter has potentially large scale; should each owner herself be directly responsible for managing all the professional users, she will easily be overwhelmed by the key management overhead. In addition, since those users' access requests are generally unpredictable, it is difficult for an owner to determine a list of them. On the other hand, different from the single data owner scenario considered in most of the existing works [8], [9], in a PHR system, there are multiple owners who may encrypt according to their own ways, possibly using different sets of crypto- graphic keys. Letting each user obtain keys from every owner who's PHR she wants to read would limit the accessibility since patients are not always online. An alternative is to employ a central authority (CA) to do the key management on behalf of all PHR owners, but this requires too much trust on a single authority (i.e., cause the key escrow problem).

In this paper, we endeavour 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 nontrivial to solve, and remain largely open up-to-date. To this end, we make the following main contributions:

We propose a novel based framework for patient-centric secure sharing of PHRs in cloud computing environments. User should assign the key to the user to which he/she would like to share the file. To address the key management challenges, we left the use to assign the key. In particular, the majority professional users are managed by attribute, while each owner only needs to manage the keys of a small number of users in her personal domain also. 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 ensures less work to have write access control, handles dynamic policy updates, and provides break-glass access to PHRs under emergence scenarios with the help of Security authority by providing access with the attribute.

Owners directly assign access privileges for personal users and encrypt a PHR file under its data attributes, 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.

2. RELATED WORK

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

A. Symmetric key cryptography (SKC) based solutions:

Symmetric-key algorithms are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link Vimercati et.al.[6] 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.

B. Public key cryptography (PKC) based solutions:

PKC based solutions were proposed due to its ability to separate write and read privileges. To realize fine-grained access control, the traditional public key encryption (PKE) based schemes proposed by J. Benaloh, M. Chase, E.Horvitz, and K. Lauter [8] in their work "Patient controlled encryption: ensuring privacy of electronic medical records" they purpose the solution scenario and shows how public and symmetric based encryption used , disadvantage of their

solution is either incur high key management overhead, or require encrypting multiple copies of a file using different users' keys.

C. Attribute Based Encryption based solutions:

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 healthcare records (EHRs). Narayan et al. proposed an attribute-based infrastructure for EHR systems, where each patient's EHR files are encrypted using a broadcast variant of Cipher Text-ABE (CP-ABE)[4] .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. Ibraimi et.al. [5] applied cipher text policy ABE (CP-ABE) [11] to manage the sharing of PHRs, and introduced the concept of social/professional domains but they do not use multi-authority ABE In [3], 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. Drawback is device dependency and revocation is not supported. Other Common drawback of all above solutions is problem of key-escrow as they consider single trusted authority.

3. PROPOSED SOLUTION

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

3.1 Problem Definition

We consider a PHR system where there are multiple PHR owners and PHR users. The owners refer to patients who have full control over their own PHR data, i.e., they can create, manage, and delete it. There is a central server belonging to the PHR service provider that stores all the owners' PHRs. The users may come from various aspects; for example, a friend, a caregiver or a researcher. Users access the PHR documents through the server in order to read or write to someone's PHR, and a user can simultaneously have access to multiple owners' data.

3.2 Security Model

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

3.3 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. 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 [12]. 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.

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.

The data access policies should be flexible, i.e. Dynamic changes to the predefined policies shall be allowed, and especially the PHRs should be accessible under emergency scenarios.

3.4 Architecture

Fig.1 depicts the architecture of proposed system for secure sharing of the personal Health records. The system is split into two 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, medical researchers and insurance agents. For each PSD, its users are personally associated with a data owner (such as family members or close friends), and they make accesses to medical records based on access rights assigned by the owner. Here we consider Data owner who possess the medical record, data reader as who can read the encrypted medical record. In PSD, the owner used key-policy attributed based encryption and generates secret key for their PSD users and in PUD the multi-authority attribute based encryption is preferred. Secret Key for PUD users are generated by Multiple authority (For this paper we consider Specialization Authority and Medical Authority) depending on their specialization and profession in combine.

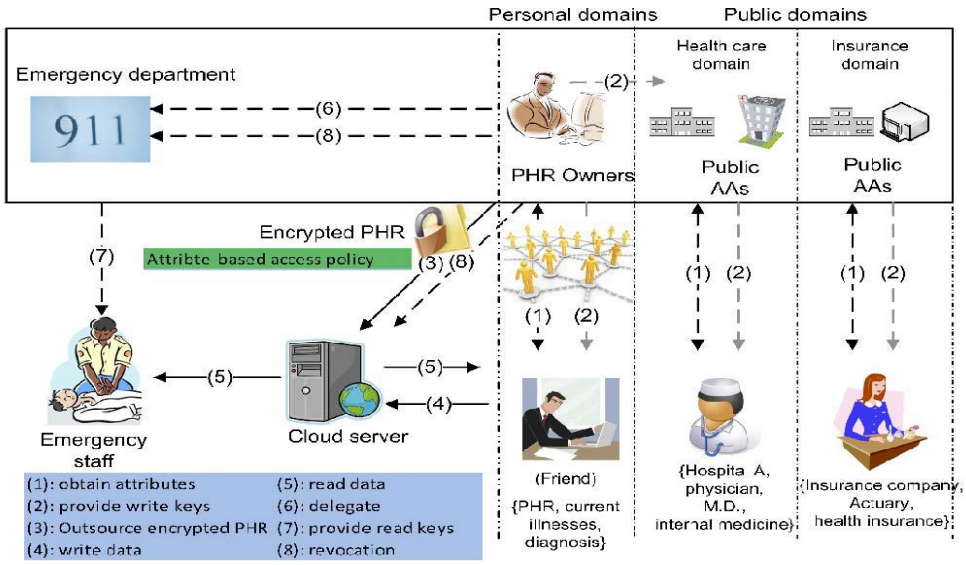

Figure 1. The Proposed Framework for patient-centric secure PHR sharing

For example, in Fig. 2, an "allergy" file's attributes are PHR; medical history; allergy. 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).

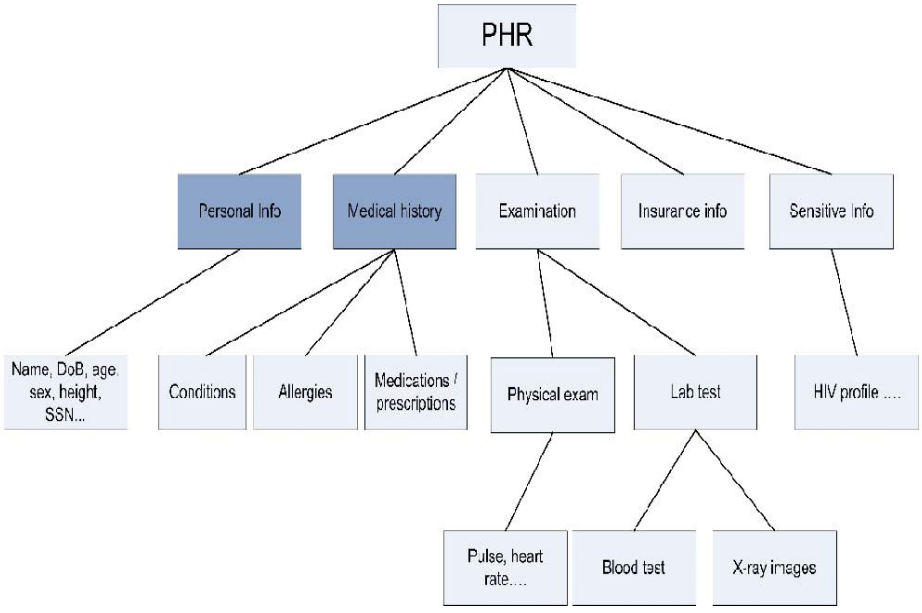

Figure 2. The Attribute Hierarchy of Files

3.5 System Implementation

This stage focuses on specific tools such as programming languages, libraries and components which allow to quickly producing software of high quality. Implementation is the stage of project when theoretical design is turned down into a working system. Thus it can be

considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that new system will work and be effective.

3.5.1 Main Software Requirements are as follows:

Java Paring based Cryptography Library and Eclipse IDE, My-SQL Server as Database Server, Servlets and JSP for GUI development.

3.5.2 Algorithms

1) ABE Algorithm

It is a type of public key Encryption in which the public key of a user and the cipher-text are dependent about attributes (e.g. the country he lives, the kind of subscription he has).
An (Key-Policy) Attribute Based Encryption scheme consists of four steps.

Setup Attributes: This algorithm is used to set attributes for users. This is a randomized algorithm that takes no input other than the implicit security parameter. It defines a bilinear group G1 of prime order p with a generator g, a bilinear map $e : G1 \times G1 \rightarrow G2$ which has the properties of bi-linearity, computability, and non-degeneracy. From these attributes public key and master key for each user can be determined. The attributes, public key and master key are denoted as
Attributes-$U = \{ 1,2. . . N\}$
Public key-$PK = (Y, T_1, T_2, . . . , T_N)$
Master key-$MK = (y, t_1, t_2, . . . , t_N)$
Where $TI \in G_1$ and $ti \in Zp$ are for attribute i, $1 \leq i \leq N$, and $Y \in G_2$ is another public key component. We have $T_{i=gti}$ and $Y = e(g, g)y$, $y \in Zp$. While PK is publicly known to all the parties in the system, MK is kept as a secret by the authority party.

Encryption: This is a randomized algorithm that takes a message M, the public key PK, and a set of attributes I as input. It outputs the cipher text E with the following format:
$\boldsymbol{E = (I, \tilde{E}, \{E_i \in I)}$
Where $\tilde{E} = MY s, E_i = T is.$ And s is randomly chosen from Z.

Secret key generation: This is a randomized algorithm that takes as input an access tree T, the master key MK, and the public key K. It outputs a user secret key SK as follows. First, it defines a random polynomial $p_i(x)$ for each node I of T in the top-down manner starting from the root node r. For each non-root node j, $p_j(0) = p\ parent\ (j)(idx(j))$ where $parent\ (j)$ represents j's parent and $idx\ (j)$ is j's unique index given by its parent. For the root node r, $pr\ (0) = y$. Then it outputs SK as follows. **SK = {sk i }i ∈ L**
Where L denotes the set of attributes attached to the leaf nodes of T and sk $i = g\ p_i(0)/t_i$.

Decryption: This algorithm takes as input the cipher text **E** encrypted under the attribute set **I**, the user's secret key **SK** for access tree **T**, and the public key **PK**. It first computes e(E i,sk i)=e(g, g)pi(0)s for leaf nodes. Then, it aggregates these pairing results in the bottom-up manner using the polynomial interpolation technique. Finally, it may recover the blind factor Y s =e(g,g) ys and output the message **M** if and only if **I** satisfies **T**

2) RSA Algorithm

The system works on a public and private key system. The public key is made available to everyone. With this key a user can encrypt data but cannot decrypt it, the only person who can decrypt it is the one who possesses the private key.

Algorithm: First of all, two large distinct prime numbers *p* and *q* must be generated. The product of these, we call *n* is a component of the public key. It must be large enough such that the numbers *p* and *q* cannot be extracted from it - 512 bits at least i.e. numbers greater than 10154. We then generate the encryption key *e* which must be co-prime to the number $m = (n) = (p-1)(q-1)$. We then create the decryption key d such that de mod m = 1. We now have both the public and private keys.

Encryption: We let $y = E(x)$ be the encryption function where *x* is an integer and *y* is the encrypted form of x, y = xe mod n

Decryption: We let $X = D(y)$ be the decryption function where *y* is an encrypted integer and *X* is the decrypted form of y, X = yd mod n

3) DES Algorithm

DES is the archetypal block cipher , an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits.

4. RESULTS

The Experiment measures the different Time comparisons for various internet connections Speed that affects system Performance. We vary the internet connection speed for 512 Kbps it will take longer time as we increase file size gradually so we get nonlinear curve and for 1 Mbps internet connection speed it take lesser time compared with first one so it shows a linear curve as we increase file size gradually and finally for 2 Mbps internet connection speed it takes lesser time compared to both above shown in graph so it shows a linear curve as we increase in file size gradually as shown in Table.

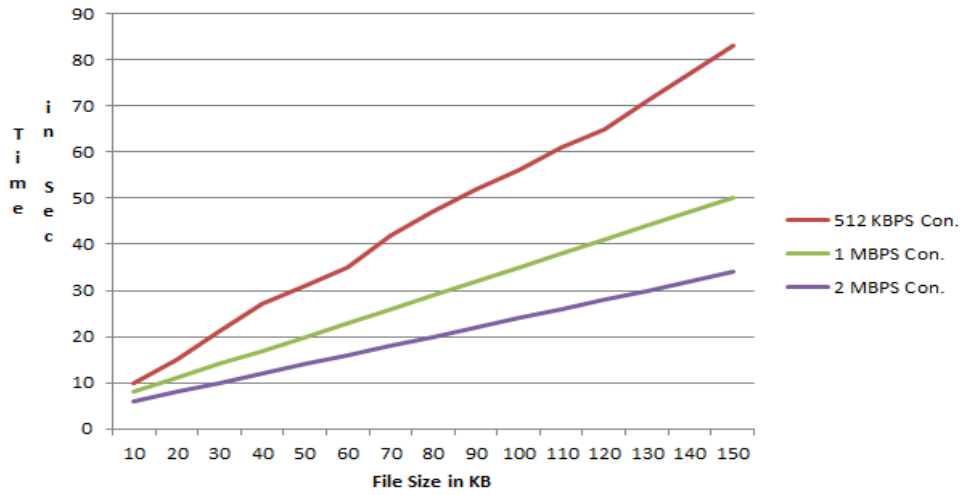

Figure 3. Time Comparison of various Internet connection speed

File Size (KB)	512 Kbps con	1Mbps con	2 Mbps con
10	10	8	6
20	15	11	8
30	21	14	10
40	27	17	12
50	31	21	14
60	35	23	16
70	42	26	18
80	47	29	20
90	52	32	22
100	56	35	24
110	61	38	26
120	65	41	28
130	71	44	30
140	77	47	32
150	83	50	34

Table 1. Various File Size and Different Internet Connections Speed

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel framework of secure sharing of personal health records in cloud computing. The main motto of the patient centric model is that the share the personal health records of the patient with maximum security, as the cloud servers are trustworthy. Patients shall have full control over encrypting their PHR files to allow fine-grained access. We encrypt the PHR files based on the algorithm ABE (Attribute based encryption), 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.

Through implementation we show that our solution is both scalable and efficient. In our Future Work we can upload Video File and Maintaining User record that have all accessed the Data.

REFERENCES

[1] M.Li,S. Yu, K. Ren, and W.Lou, "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings," pp. 89-106, Sept. 2010

[2] H. Lö hr, A.-R. Sadeghi, and M. Winandy, "Securing the E-Health Cloud," pp. 220-229, 2010.

[3]. A. Akinyele, C.U. Lehmann, M.D. Green, M.W. Pagano, Z.N.J. Peterson, and A.D. Rubin. Self-protecting electronic medical records usingattribute-based encryption on mobile device. Technical report, Cryptology ePrint Archive, Report 2010/565, 2010.

[4]. S. Narayan, M. Gagn´e, and R. Safavi-Naini, "Privacy preserving ehr system using attribute-based infrastructure," ser. CCSW '10, 2010, pp. 47–52.

[5]. L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Ciphertext-policy attribute-based threshold decryption with flexible delegation and revocation of user attributes," 2009.

[6] "At Risk of Exposure - in the Push for Electronic Medical Records, Concern Is Growing About How Well Privacy Can Be Safeguarded, Jun 26,2006.

[7] K.D. Mandl, P. Szolovits, and I.S. Kohane, "Public Standards and Patients' Control: How to Keep Electronic Medical Records Accessible but Private," BMJ, vol. 7281, pp. 283-287, Feb. 2001.

[8] J. Benaloh, M. Chase, E.Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," pp. 103-114, 2009.

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

[10] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," (ASIACCS '10), 2010

[11] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption" 2007.

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

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

