Secure Sharing of Personal Health Record in Cloud Computing Using Encryption Techniques

Anjusha K K, Aswathi P V

***Abstract*— A personal health record (PHR) is an electronic application used by patients to maintain and manage their health information in a private, secure, and confidential environment, and which is stored at a third party, such as cloud providers. The data to be encrypt by the patient before outsourcing for the assurance of data access. The main issues are risks of privacy exposure, scalability in key management, flexible access, and efficient user revocation, have remained the most important challenges toward achieving fine-grained, and cryptographically enforced data access control. So here we have 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 secure data access control for PHRs, Attribute Based Encryption (ABE) and RSA techniques to encrypt each patient's PHR file. And also focus on the multiple data owner scenario, and divide the users in the PHR system into multiple security domains that greatly reduces the key management complexity for owners and users. By the use of these two encryption methods high degree of patient privacy is guaranteed. This scheme also enables break-glass access under emergency scenarios. Extensive analytical and experimental results are presented which show the security and efficiency of the scheme.**

***Index Terms*— Personal health records, cloud computing, data privacy, fine-grained access control, attribute-based encryption, RSA algorithm.**

I. INTRODUCTION

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 their 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 their medical records and can share the health data with a wide range of users, including healthcare providers, family members or friends. Patient-centric privacy control over their own PHRs are ensured and it is essential to have fine-grained data access control mechanisms that work with semi-trusted servers.

The most promising approach would be to encrypt the data before outsourcing. The PHR owner itself should decide how to encrypt the files and to allow which set of users to obtain access to each file. Those who are giving the corresponding decryption key, who can view the PHR file of a patient. The patient shall always retain the right to not only grant, but also revoke access privileges when they feel it is necessary. One of the main concerns is to divide the system into multiple domains, which contain two domains: public domain and personal domain. Public domain contains professional users of the PHR, for example, doctors, nurses etc and the personal domain contains the persons who have direct contact with the patient, which might be their friends or relatives.

The studies of the PHRs stored on semi-trusted servers are focus on addressing the complicated and challenging key management issues. In order to protect the personal health data stored on a semi-trusted server, the methods of attribute based encryption and RSA are adopted. Using these encryption primitives, access policies are expressed based on the attributes of users or data, which enables a patient to selectively share their 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.

II. LITERATURE SURVEY

The most relevant work is the work done by Goyal et al.'s seminar paper on ABE [1], data are encrypted under a set of attributes so that multiple users who possess proper keys can decrypt. This potentially makes encryption and key management more efficient. A fundamental property of ABE is preventing against user collusion.

In the work of Narayan et al.[2] proposed an attribute-based infrastructure for EHR systems, where each patient's PHR files are encrypted using a broadcast variant of ABE that allows direct revocation. However, the ciphertext length grows linearly with the number of unrevoked users.

However, it has several drawbacks. First, it usually assumes the use of a single trusted authority (TA) in the system. This not only may create a load bottleneck, but also suffers from the key escrow problem since the TA can access all the encrypted files, opening the door for potential privacy exposure. In addition, it is not practical to delegate all attribute management tasks to one TA, including certifying all users' attributes or roles and generating secret keys. In fact, different organizations usually form their own sub domains and become suitable authorities to define and certify different sets of attributes belonging to their sub domains (i.e., divide and rule). For example, a professional association would be responsible for certifying medic al specialties, while a regional health provider would certify the job ranks of its staffs. Second, there still lacks an efficient and on-demand user revocation mechanism for ABE with the support for dynamic policy updates/changes, which are essential parts of

Manuscript received March 27, 2014.
Anjusha K K, Department of Computer Science, KMCT College of Engineering, Calicut, India, 09645119073,
Aswathi P V Department of Computer Science, KMCT College of Engineering, Calicut, India, 08129837233.

secure PHR sharing. Finally, most of the existing works do not differentiate between the personal and public domains (PUDs), which have different attribute definitions, key management requirements, and scalability issues. Our idea of conceptually dividing the system into two types of domains is similar with that in; however, a key difference is in [3] a single TA is still assumed to govern the whole professional domain.

In the YWRL scheme [4], the data owner is also a trusted authority (TA) at the same time. It would be inefficient to be applied to a PHR system with multiple data owners and users, because then each user would receive many keys from multiple owners, even if the keys contain the same sets of attributes.

III. EXISTING SYSTEM

Their sensitive personal health information (PHI), especially when they are stored on a third-party server which people may not fully trust. On the one hand, although there exist healthcare regulations such as HIPAA which is recently amended to incorporate business associates, cloud providers are usually not covered entities. On the other hand, due to the high value of the sensitive PHI, the third-party storage servers are often the targets of various malicious behaviors which may lead to exposure of the PHI. As a famous incident, a Department of Veterans Affairs database containing sensitive PHI of 26.5 million military veterans, including their social security numbers and health problems was stolen by an employee who took the data home without authorization. To ensure patient-centric privacy control over their own PHRs, it is essential to have fine-grained data access control mechanisms that work with semi trusted servers. A feasible and promising approach would be to encrypt the data before outsourcing. Basically, the PHR owner herself should decide how to encrypt her files and to allow which set of users to obtain access to each file. A PHR file should only be available to the users who are given the corresponding decryption key, while remain confidential to the rest of users.

A. *Disadvantages of Existing 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.
- 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.

IV. PROPOSED SYSTEM

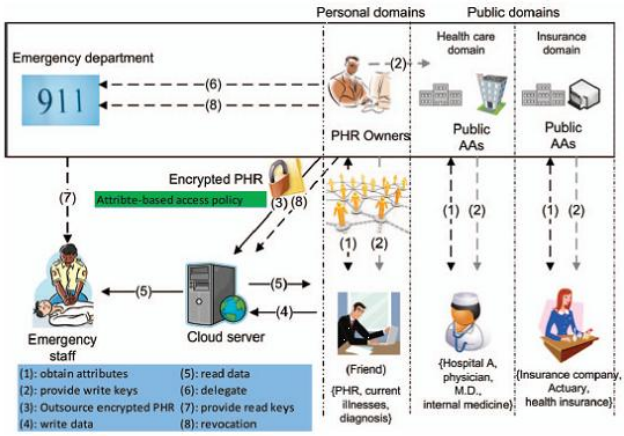
The mostly related to works in cryptographically enforced access control for outsourced data and attribute based encryption. To assure the patients' control over access to their own PHRs, it is a promising method to encrypt the PHRs before outsourcing. In this paper, we propose a novel patient-centric 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) and RSA techniques to encrypt each patient's PHR file.

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.

In order to protect the personal health data stored on a semi-trusted server, we adopt RSA with the patient's attribute 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 their 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.

A. *Advantages of Proposed System*

- 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.
- 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.



Architecture of Personal Health Records

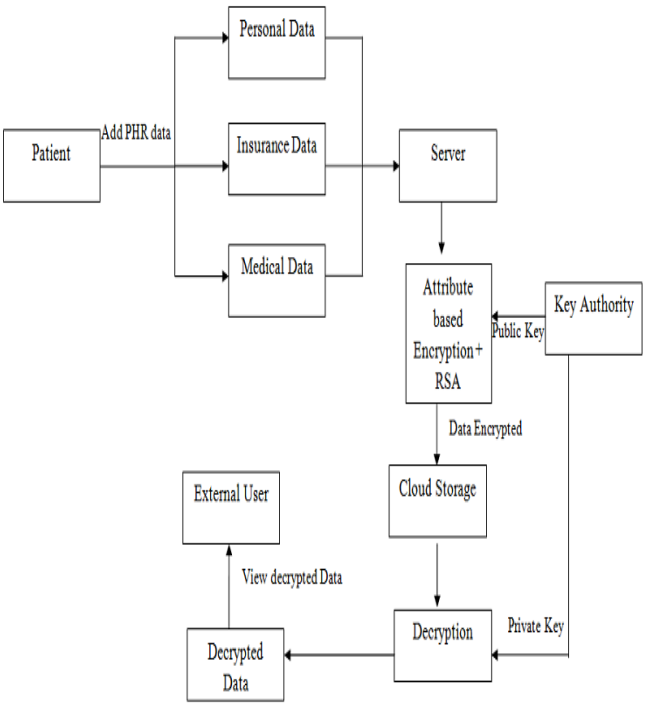
V. ENCRYPTION METHODS

The RSA Algorithm is a form of public key encryption. Public key encryption is a process where each user is given two keys, one which is public and seen by anyone who wishes to see it, and one which is kept strictly private. The thought of making one of your keys public seems bizarre at first, but we will soon see why this is so effective and secure. Each message is encrypted using your recipient's public key. You can get your recipient's public key easily- it's public. However, only your recipient has the corresponding private key to decode it. So suppose someone intercepted your message. They would only have access to the public keys, but almost no way to actually decrypt the message. The only way would be to reverse the algorithm, which is extremely difficult.

The iterative procedure can be shown accordingly:

1. *Key Generation.*
2. *Encryption.*
3. Decryption.

RSA algorithm overview is:
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. We then generate the encryption key e which must be co-prime to the number m = ϕ(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.



Block Diagram of Personal Health Record

A. *RSA ALGORITHM*

RSA encrypts messages through the following algorithm, which is divided into 3 steps:

1. Key Generation

I. Choose two distinct prime numbers p and q.

II. Find n such that n = pq.
n will be used as the modulus for both the public and private keys.

III. Find the totient of n, ϕ (n)
ϕ(n)=(p-1)(q-1).

IV. Choose an e such that 1 < e < ϕ(n), and such that e and ϕ(n) share no divisors other than 1
(e and ϕ(n) are relatively prime).
e is kept as the public key exponent.

V. Determine d (using modular arithmetic) which satisfies the congruence relation de ≡ 1 (mod ϕ (n)).

In other words, pick d such that de - 1 can be evenly divided by (p-1) (q-1), the totient, or ϕ(n).

This is often computed using the Extended Euclidean Algorithm, since e and ϕ(n) are relatively prime and d is to be the modular multiplicative inverse of e.
d is kept as the private key exponent.

The public key has modulus n and the public (or encryption) exponent e. The private key has modulus n and the private (or decryption) exponent d, which is kept secret.
Equations

2. Encryption

I. Person A transmits his/her public key (modulus n and exponent e) to Person B, keeping his/her private key secret.

II. When Person B wishes to send the message "M" to Person A, he first converts M to an integer such that 0 < m < n by using agreed upon reversible protocol known as a padding scheme.

III. Person B computes, with Person A's public key information, the ciphertext c corresponding to
$c \equiv m^e \pmod n$.

IV. Person B now sends message "M" in ciphertext, or c, to Person A.

3. Decryption

I. Person A recovers m from c by using his/her private key exponent, d, by the computation

$m \equiv c^d \pmod n$.

II. Given m, Person A can recover the original message "M" by reversing the padding scheme.
This procedure works since
$C \equiv m^e \pmod n$,
$c^d \equiv (m^e)^d \pmod n$,
$c^d \equiv m^{de} \pmod n$.

By the symmetry property of mods we have that
$m^{de} \equiv m^{de} \pmod n$.
Since $de = 1 + k\phi(n)$, we can write
$m^{de} \equiv m^{1 + k\phi(n)} \pmod n$,
$m^{de} \equiv m(m^k)^{\phi(n)} \pmod n$,
$m^{de} \equiv m \pmod n$.
From Euler's Theorem and the Chinese Remainder Theorem, we can show that this is true for all m and the original message
$c^d \equiv m \pmod n$, is obtained.

B. *Advantages*

The RSA algorithm is indeed among the strongest, but can it withstand anything? Certainly nothing can withstand the test of time. In fact, no encryption technique is even perfectly secure from an attack by a realistic cryptanalyst. Methods such as brute-force are simple but lengthy and may crack a message, but not likely an entire encryption scheme. We must also consider a probabilistic approach, meaning there's always a chance someone may get the \one key out of a

million". So far, we don't know how to prove whether an encryption scheme is unbreakable. If we cannot prove it, we will at least see if someone can break the code. This is how the NBS standard and RSA were essentially certified. Despite years of attempts, no one has been known to crack either algorithm. Such a resistance to attack makes RSA secure in practice.

We will see why breaking RSA is at least as hard as factoring n. Factoring large numbers is not provably hard, but no algorithms exists today to factor a 200-digit number in a reasonable amount of time. Fermat and Legendre have both contributed to this field by developing factoring algorithms, though factoring is still an age-old math problem. This is precisely what has partially certified RSA as secure.
To show that RSA is secure, we will consider how a cryptanalyst may try to obtain the decryption key from the public encryption key, and not how an intruder may attempt to "steal" the decryption key. This should be taken care of as one would protect their money, through physical security methods. The authors of RSA provide an example: the encryption device (which could be, say, a set of integrated chips within a computer), would be separate from the rest of the system. It would generate encryption and decryption keys, but would not print out the decryption key, even for its owner. It would, in fact, erase the decryption key if it sensed an attempted intrusion.

VI. CONCLUSIONS

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 and RSA 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. Through implementation and simulation, we show that our solution is both scalable and efficient.

REFERENCES

[1] 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.
[2] S. Narayan, M. Gagne´ , and R. Safavi-Naini, "Privacy Preserving EHR System Using Attribute-Based Infrastructure," Proc. ACM Cloud Computing Security Workshop (CCSW '10), pp. 47-52, 2010.
[3] Y L. Ibraimi, M. Asim, and M. Petkovic, "Secure Management of Personal Health Records by Applying Attribute-Based Encryp-tion," technical report, Univ. of Twente, 2009.
[4] 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.

Anjusha K K, she is pursuing M.TECH (CS) in KMCT College of Engineering, Calicut, Kerala, India. She has received B.E in Computer Science and Engineering. His main research interest includes Cryptography, Cloud Computing and Image Processing.

Aswathi P V, she is currently with the Department of Computer Science and Engineering, KMCT college of Engineering, Calicut, Kerala, India. She is having 2 years of teaching experience. Her main research interest includes Cryptography, Image Processing and Neural Networks