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. 2, February 2014, pg.617 – 625

RESEARCH ARTICLE

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

Prof. Y. B. Gurav[1], Manjiri Deshmukh[2]

[1]Computer&Pune university, India

[2]Computer&Pune university, India

1 *ybgurav@gmail.com* **2** *manjirideshmukh15@gmail.com*

Abstract*: Personal health record is maintain in the centralize server to maintain patient's personal and diagnosis information. Personal health record (PHR) is an emerging patient-centric model of health information exchange, which is often outsourced to be stored at a third party, such as cloud providers. However, there have been wide privacy concerns as personal health information could be exposed to those third party servers and to unauthorized parties. The security schemes are used to protect personal data from public access. To assure the patients' control over access to their own PHRs, it is a promising method to encrypt the PHRs before outsourcing .In this paper we propose novel patient-centric framework and suite of mechanism for data access control to PHR's stored in semi-trusted servers 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. To achieve fine-grained and scalable data access control for PHRs, we leverage attribute based encryption (ABE) techniques to encrypt each patient's PHR file. Data owner update the personal data into third party cloud data centers. Multiple data owners can access the same data values. Different from previous works in secure data outsourcing, we 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. A high degree of patient privacy is guaranteed simultaneously by exploiting multi-authority ABE. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios.***

Keyword*—Personal health records; cloud computing; data privacy; fine-grained access control; attribute-based encryption*

I. Introduction

In recent years, personal health record (PHR) has e- merged 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 centers, many PHR services are outsourced to or provided by third-party service providers, for example, Microsoft HealthVault1. Recently, architectures of storing PHRs in cloud computing have been proposed in .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. On the other hand, due to the high value of the sensitive personal health information (PHI), the third-party storage servers are often the targets of various malicious behaviours which may lead to expo- sure 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 en- crypt 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. Furthermore, the patient shall always retain the right to not only grant, but also revoke access privileges when they feel it is necessary. 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 in a PHR system, there are multiple owners who may encrypt according to their own ways, possibly using different sets of cryptographic keys. Letting each user obtain keys from every owner whose 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 endeavor to study the patient- centric, secure sharing of PHRs stored on semi-trusted servers, and focus on addressing the complicated and challenging key management issues. In order to protect the personal health data stored on a semi-trusted server, we adopt attribute-based encryption (ABE) as the main encryption primitive. Using ABE, access policies are expressed based on the attributes of users or data, which enables a patient to selectively share her PHR among a set of users by encrypting the file under a set of attributes, without the need to know a complete list of users. The complexities per encryption, key generation and decryption are only linear with the number of attributes involved. However, to integrate ABE into a large-scale PHR system, important issues such as key management scalability, dynamic policy updates, and efficient on-demand revocation are non-trivial to solve, and remain largely open up-to-date. To this end, we make the following main contributions: (1) We propose a novel ABE-based framework for patient-centric secure sharing of PHRs in cloud computing environments, under the multi-owner settings. To ad- dress the key management challenges, we conceptually divide the users in the system into two types of domains, namely public and personal domains. In particular, the majority professional users are managed distributively by attribute authorities in the former, while each owner only needs to manage the keys of a small number of users in her personal domain. In this way, our framework can simultaneously handle different types of PHR sharing applications' requirements, while incurring minimal key management overhead for both owners and users in the system. In addition, the framework enforces write access control, handles dynamic policy updates, and provides break-glass access to PHRs under emergence scenarios. (2) In the public domain, we use multi-authority ABE (MA-ABE) to improve the security and avoid key escrow problem. Each attribute authority (AA) in it governs a disjoint subset of user role attributes, while none of them alone is able to control the security of the whole system. We propose mechanisms for key distribution and encryption so that PHR owners can specify personalized fine-grained role-based access policies during file encryption. In the personal domain, owners directly assign access privileges for personal users and encrypt a PHR file under its data attributes. (3)Furthermore, we enhance MA-ABE by putting forward an efficient and on-demand user/attribute revocation scheme, and prove its security under standard security assumptions. In this way, patients have full

privacy control over their PHRs. We provide a thorough analysis of the complexity and scalability of our proposed secure PHR sharing solution, in terms of multiple metrics in computation, communication, storage and key management. We also compare our scheme to several previous ones in complexity, scalability and security. Furthermore, we demonstrate the efficiency of our scheme by implementing it on a modern workstation and performing experiments/simulations.

II. Literature survey

This paper is mostly related to works in cryptograph-icily enforced access control for outsourced data and attribute based encryption. To realize fine-grained access control, the traditional public key encryption (PKE) based schemes either incur high key manage-men overhead, or require encrypting multiple copies of a file using different users' keys. To improve upon the scalability of the above solutions, one-to-many en-crypt ion methods such as ABE can be used. In Goya let. all's seminal paper on ABE data is 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. Fundamental property of ABE is preventing against user collusion. In addition, the encrypt or is not required to know the ACL.

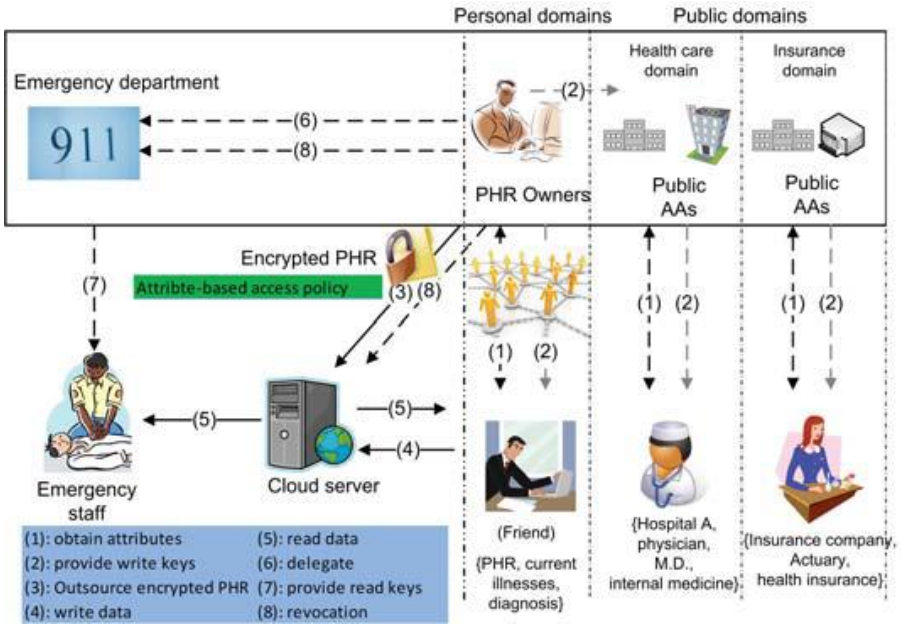
A. ABE for Fine-grained Data Access Control

A number of works used ABE to realize fine-grained access control for outsourced data [9]. Especially, there has been an increasing interest in applying ABE to secure electronic healthcare records (EHRs).Recently, Narayan et al. proposed an attribute-based infrastructure for EHR systems, 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 unrevoked users. In a variant of ABE that allows delegation of access rights is proposed for encrypted EHRs. Ibraimi et.al . applied ciphertext policy ABE (CP-ABE) to manage the sharing of PHRs, and introduced the concept of social/professional domains. In , Akinyele et al. investigated using ABE to gen-erate self-protecting EMRs, which can either be stored on cloud servers or cellphones so that EMR could be accessed when the health provider is offline. 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. 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 medical 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 secure PHR sharing. Finally, most of the existing works do not differentiate between the personal and public do-mains, which have different attribute dentitions, 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 a single TA is still assumed to govern the whole professional domain. Recently, Yu et al . (YWRL) applied key-policy ABE to secure outsourced data in the cloud where a single data owner can encrypt her data and share with multiple authorized users, by distributing keys to them that contain attribute-based access privileges. They also propose a method for the data owner to revoke a user efficiently by delegating the updates of affected ciphertexts and user secret keys to the cloud server. Since the key update operations can be aggregated over time, their scheme

achieves low amortized overhead. However, in the YWRL scheme, the data owner is also a 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. On the other hand, Chase and Chow proposed a multiple-authority ABE (CC MA-ABE) solution in which multiple TAs, each governing a different subset of the system's users' attributes, generate user secret keys collectively. A user needs to obtain one part of her key from each TA. This scheme prevents against collusion among at most N − 2 TAs, in addition to user collusion resistance. However, it is not clear how to realize efficient user revocation. In addition, since CC MA-ABE embeds the access policy in users' keys rather than the ciphertext, a direct application of it to a PHR system is non-intuitive, as it is not clear how to allow data owners to specify their file access policies.

III. Overview of a 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 (PUDs) and personal domains (PSDs)) according to the different users' data access requirements. In both types of security domains, we utilize ABE to realize cryptographically enforced, patient-centric PHR access. Especially, in a PUD multi-authority ABE is used. Each data owner 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.



A. 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) . In RBAC , 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 environ- ments, 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.

IV. Encryption method

In cloud computing, there are different existing schemes that provide security, data confidentiality and access control. Users need to share sensitive objects with others based on the recipients ability to satisfy a policy in distributed systems. One of the encryption schemes is Attribute Based Encryption (ABE) which is a new paradigm where such policies are specified and cryptographically enforced in the encryption algorithm itself. Hence, the existing ABE schemes are of two types. They are Key-Policy ABE (KP-ABE) scheme and Ciphertext-Policy ABE (CP- ABE) scheme. Encryption techniques for personal health records in cloud computing literature review as follows.

A. Attribute-Based Encryption

Attribute-Based Encryption (ABE), a generalization of identity-based encryption that incorporates attributes as inputs to its cryptographic primitives. Data is encrypted using a set of attributes so that multiple users who possess proper can decrypt. Attribute-Based Encryption (ABE) not only offers fine-grained access control but also prevents against collusion. J. Benaloh [2], has proposed a scheme in which a file can be uploaded without key distribution and it is highly efficient. But it is a single data owner scenario and thus it is not easy to add categories. C. Dong [5] has explored that the data encryption scheme does not require a trusted data server. The server can perform encrypted searches and updates on encrypted data without knowing the plaintext or the keys to decrypt. But in this scheme the server knows the access pattern of the users which allows it to infer some information about the queries. To realize fine grained access control, the traditional public key encryption based schemes and either incur high key management overhead, or require encrypting multiple copies of a file using different users keys. To improve upon the scalability of the above solutions, one-to-many encryption methods such as attribute based encryption (ABE) can be used. Sahai and Waters [7] first introduced the attribute based encryption (ABE) for enforced access control through public key cryptography. The main goal for these models is to provide security and access control. The main aspects are to provide flexibility, scalability and fine grained access control. In classical model, this system can be achieved only when user and server are in a trusted domain. So, the new access control scheme that is 'Attribute Based Encryption (ABE)' scheme was introduced which consist of key policy attribute based encryption (KP-ABE). As compared with classical model, KP-ABE provided fine grained access control. However it fails with respect to flexibility and scalability when authorities at multiple levels are considered. In ABE scheme both the user secret key and the ciphertext are associated with a set of attributes. ABE is implemented for one-to many encryption in which cipher-texts are not necessarily encrypted to one particular user, it may be for more than one number of users. Akinyele et al investigated using ABE to generate self-protecting EMRs, which can either be stored on cell phones or cloud servers so that EMR could be accessed when health provider is in offline also.

Limitations of ABE: The use of a single trusted authority (TA) in the system. Single trusted authority (TA) not only creates a load bottleneck, but also have key escrow problem since the TA can access all the encrypted files. This opens the door for potential privacy exposure.

B. Key Policy Attribute Based Encryption

V. Goyal, O. Pandey, A. Sahai, and B. Waters [5] proposed a key-policy attribute-based encryption (KP-ABE) scheme. It is the modified form of the classical model of ABE. Exploring KP-ABE scheme, attribute policies are associated with keys and data is associated with attributes. The keys only associated with the policy that is to be satisfied by the attributes that are associating the data can decrypt the data. Key Policy Attribute Based Encryption (KP-ABE) scheme is a public key encryption technique that is designed for one-to-many communications. This scheme enables a data owner to reduce most of the computational

overhead to cloud servers. The use of this encryption scheme KP-ABE provides fine-grained access control. Each file or message is encrypted with a symmetric data encryption key (DEK), which is again encrypted by a public key corresponding to a set of attributes in KPABE, which is generated corresponding to an access structure. The data file that is encrypted is stored with the corresponding attributes and the encrypted DEK. Only if the corresponding attributes of a file or message stored in the cloud satisfy the access structure of a user's key, then the user is able to decrypt the encrypted DEK, which is used to decrypt the file or message.

Limitations of KP- ABE: The main disadvantage in the scheme is that the data owner is also a Trusted Authority (TA) at the same time. If this scheme is applied to a PHR system with multiple data owners and users, it would be inefficient because then each user would receive many keys from multiple owners, even if the keys contain the same set of attributes.

C. Expressive Key Policy Attribute Based Encryption

Y. Zheng proposed Expressive Key-Policy ABE ,the encryption methods in clouds Attribute-sbased encryption (ABE), allows fine grained access control on encrypted data. In the key policy Attribute based encryption, the primitive enables senders to encrypt messages with a set of attributes and private keys are associated with access tree structure that specifies which all the cipher texts the key holder is allowed to decrypt. In most ABE systems, the ciphertext size grows linearly with the number of ciphertext attributes and the only known exceptions only support restricted forms of threshold access policies. This expressive key-policy attribute based encryption (KP-ABE) schemes allowing for non-monotonic access and with constant ciphertext size. The private keys have quadratic size in the number of attributes. On the other hand, they reduce the number of pairing evaluation size to a constant, which appears to be a unique feature among expressive KP-ABE schemes. This is more efficient than KP-ABE.

D. Cipher Text Policy Attribute Based Encryption

Sahai et al [7] introduced the concept of another modified form of ABE called CP-ABE that is Ciphertext Policy Attribute Based Encryption. In several distributed systems a user should only be able to access data if a user possess a certain set of credentials or attributes. To store the data and mediate access control a trusted server is the only method for enforcing such policies The confidentiality of the data will be compromised, if any server storing the data is compromised. The storage server is untrusted if the data can be confidential by this technique. Previous Attribute-Based Encryption systems used to the outsourced data can be described and built policies into users keys. While in this system attributes are used to describe a users credentials, and a party encrypting data determines a policy for decrypt. In ciphertext-policy attribute-based encryption (CP-ABE), depends how attributes and policy are associated with cipher texts and users decryption keys. In a CP-ABE scheme, a ciphertext is associated with a monotonic tree access structure and a user's decryption key is associated with set of attributes. In this scheme the ciphertext is encrypted with a tree access policy chosen by an encryptor, while the decryption key is generated with respect to a set of attributes. As long as the set of attributes should satisfy the tree access policy and it can be associated with a decryption key with a given ciphertext, the key can be used to decrypt the cipher text. However, basic CP-ABE schemes are far from enough to support access control in modern enterprise environments, require considerable flexibility and efficiency in specifying policies and managing user attributes.

Limitations of CP-ABE:

Decryption keys only support user attributes that are organized logically as a single set, so users can only use all possible combinations of attributes in a single set issued in their keys to satisfy policies.

E. Cipher Text Policy Attribute Set Based Encryption

S. Jahid, P. Mittal, and N. Borisov et al [6] applied CP- ASBE schemes with immediate attribute revocation capability, instead of periodical revocation. Ciphertext Policy Attribute Set Based Encryption (CP-ASBE)- a new form of CP-ABE. It organizes user attributes into a recursive set based structure and allows users to impose dynamic constraints on how those attributes may be combined to satisfy a policy. In a CP-ABE scheme, decryption keys only support user attributes that are organized logically as a single set, so users can only use all possible combinations of attributes in a single set issued in their keys to satisfy policies.

To solve this problem, ciphertext-policy attribute-set- based encryption is introduced. Thus, by grouping user attributes into sets such that those belonging to a single set have no restrictions on how they can be combined, CP-ASBE can support compound attributes without sacrificing the flexibility to easily specify policies involving the underlying singleton While restricting users to use attributes from a single set during decryption can be thought of as a regular CP-ABE scheme, the challenge in constructing a CP-ASBE scheme is in selectively allowing users to combine attributes from multiple sets within a given key while still preventing collusion.

Limitations of CP-ASBE: Constructing a CP-ASBE scheme is in selectively allowing users to combine attributes from multiple the cloud providers. However, HABE uses disjunctive normal form policy and assumes all attributes in one conjunctive clause are administrated by the same domain master by multiple domain masters. The same attribute may be administrated according to specific policies, which is difficult to implement in practice.

F. Identity Based Encryption (IBE) and Hierarchical Identity Based Encryption (HIBE):

M. Franklin, D.Boneh [3] introduced an identity-based encryption scheme, data is encrypted using an arbitrary string as the key and for decryption; a decryption key is mapped to the arbitrary encryption key by a key authority. Hierarchical Identity Based Encryption (HIBE) is the hierarchical form of a single IBE [3]. The concept of HIBE scheme can help to explain the definition of security. Ina regular IBE (1-HIBE) scheme, there is only one private key generator (PKG) that distributes private keys to each users, having public keys are their primitive ID (PID) arbitrary strings. A two-level HIBE (2-HIBE) scheme consists of a root PKG, domain PKGs and users, all of which are associated with PID's. A users public key consists of their PID and their domains. In a 2-HIBE, users retrieve their private key from their domain PKG. The private key PK is compute by Domain PKGs of any user in their domain, their domain secret key-SK can be provided and previously requested from the root PKG. Similarly, is for number of sub-domains. There also includes a trusted third party or root certificate authority that allows a hierarchy of certificate authorities: Root certificate authority issues certificates for other authorities or users in their respective domains. The original system does not allow for such structure. However, a hierarchy of PKG is reduces the workload on root server and allows key assignment at several levels.

Limitations of IBE: The main disadvantage of this system is key management overhead. Letting each user obtain keys from every owner PHR wants to read would limit the accessibility.

G. Hierarchical Attribute-Base Encryption (HABE) and Hierarchical Attribute Set Based Encryption (HASBE)

This scheme Hierarchical attribute-based encryption (HABE) is derived by Wang et al .It is designed to achieve fine-grained access control in cloud storage services. It is a combination of HIBE and CP-ABE. In the HABE scheme, there are multiple keys with different usages.

Limitations of HASBE: Compared with ASBE, this scheme cannot support compound attributes efficiently and does not support multiple value assignments.

H. Distributed Attribute - Based Encryption

S. Ruj, A. Nayak, and I. Stojmenovic [9] introduced a concept of Distributed Attribute-Based Encryption (DABE). In DABE, there will be an arbitrary number of parties to maintain attributes and their corresponding secret keys. There are three different types of entities in a DABE scheme[9]:1. The master is responsible for the distribution of secret user keys. However, master is not involved in the creation of secret attribute keys.2. Attribute authorities are responsible to verify whether a user is eligible of a specific attribute; in this case they distribute a secret attribute key to the user. An attribute authority generates a public attribute key for each attribute it maintains; this public key will be available to all the users. Eligible users receive a personalized secret attribute key over an authenticated and trusted channel. 3. Users can encrypt and decrypt messages. To encrypt a message, user should formulate the access policy in Disjunctive Normal Form (DNF).To decrypt a ciphertext, a user needs at least access to some set of attributes which satisfies the access policy. The main advantage of the solution is each user can obtain secret keys from any subset of the Trusted Authorities (TAs) in the system.

Limitations of DABE: It requires a data owner to transmit an updated ciphertext component to every non-revoked user. While sharing the information the communication overhead of key revocation is still high.

I. Ciphertext policy ABE

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. 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.

J. Homorphic Encryption

An encryption scheme has algorithm consists of three steps[2].

1. Key Generation - creates two keys i.e. the privacy key prk and the public key puk.

2. Encryption - encrypts the plaintext P with the public key puk to yield ciphertext C.

3. Decryption - decrypts the ciphertext C with the privacy key prk to retrieve the plaintext P.

4. Evaluation - outputs a ciphertext C of f(P) such that Decrypt (prk,P) = f(P).

The scheme becomes homomorphic if f can be any arbitrary function, and the resulting ciphertext of Eval is compact. That means it does not grow too large regardless of the complexity of function f). The Eval algorithm in essence means that the scheme can evaluate its own decryption algorithm. Utilizing Homomorphic Authenticators[11] to significantly reduce the arbitrarily large communication Overhead for public auditability without introducing any online burden on the data owner, we resort to the homomorphic authenticator technique Homomorphic authenticators are unforgeable metadata generated from individual data blocks, which can be securely aggregated in such a way to assure a verifier that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator.

Conclusion

In this paper made a survey on the Improving the Security on Public Health Record System in Cloud Computing. And also made a detailed study about what are the techniques is needed for security the Health Record System. Attribute Based Encryption is the good technique to securing the Health records. It is efficient in the Conjunctive Property. But somewhat limitations on MA-ABE in real time with the property of Disjunctive as well as it had the little bit problem while revocation. Because it can be affect the non-revoked users. So move to the Attribute Based Broadcast Encryption. It satisfies the Disjunctive Property also and handles the revocation perfectly. Identity Based Encryption is the better way to provide the authentication for the Public Health Record System. homomorphic encryption with data auditing is used to verify the trustworthiness of third party auditor.

References

[1] Sahai and B. Waters. “Fuzzy Identity Based Encryption.”, In Advances in Cryptology – Eurocrypt, volume 3494 of LNCS, pages 457–473. Springer, 2005.

[2] Li, M., Lou,W., Ren, K., “ Data security and privacy in wireless body area networks”, IEEE Wireless Communications Magazine (February 2010).

[3] 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.

[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] Ming LiShucheng Yu, Yao Zheng,Kui Ren, and Wenjing Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption", IEEE transactions on parallel and distributed systems, vol. 24, no. 1, january 2013.

[6] Y. Zheng, "Privacy-Preserving Personal Health Record System Using Attribute-Based Encryption", master's thesis, Worcester Polytechnic Inst., 2011.

[7] 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.

[8] 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.

[9] 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.

[10] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption", Proc. IEEE Symp. Security and Privacy (SP '07), pp. 321-334, 2007.

[11] Q.Wang et al.,"Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. ESORICS „09, Sept. 2009, pp. 355–70.

Acknowledgement

I take this opportunity to express my deepest thanks and gratitude to head of the department and guide Prof.Y.B.Gurav sir for his inspiring guidance, constant support encouragement and suggestions to shape and preparation of the paper in a systematic way.

