International Journal of Research (IJR) Vol-1, Issue-11 December 2014 **ISSN 2348-6848**

Scalable and Secure Sharing of PHR using Distributed Attribute-Based Encryption in Cloud

[1] P V B Sivappa , [2]G.Rama Swamy

[1]M.Tech Research Scholar, Department of CSE,
[2] Professor, Department of CSE
Priyadarshini Institute of Technology & Science, Chintalapudi, India

Abstract— *Cloud computing has emerged as one of the most influential paradigms in the IT industry for last few years. Normally data owners and service providers are not in the same trusted domain in cloud computing. 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 the information could be exposed to those third party servers and to unauthorized parties. In the existing system, 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, leverage attribute based encryption (ABE) techniques to encrypt each patient's PHR file. A high degree of patient privacy is guaranteed simultaneously by exploiting multi-authority ABE. In proposed System, introduce the concept of Distributed Attribute-Based Encryption (DABE), where an arbitrary number of parties can be present to maintain attributes and their corresponding secret keys. Also two-level access control model introduced, that combines fine-grained access control, which supports the precise granularity for access rules, and coarse-grained access control, which allows the storage provider to manage access requests while learning only limited information from its inputs. This is achieved by arranging outsourced resources into units called access blocks and enforcing access control at the cloud only at the granularity of blocks.*

Index Terms— Attribute Based Encryption, Distributed Attributed Based Encryption, PHR, Cloud Computing, Coarse Grain

1. INTRODUCTION

Cloud computing is a computing model, where resources such as computing power, storage, network and software are abstracted and provided as services on the internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (Iaas), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Cloud computing provides on-demand self service, in which the different business units are allowed to get the computing

resources as they need without having to go through IT for equipment .It supports broad network access, which allows applications to be built in ways that align with how businesses operate today in mobile, multi-device, etc. It allows resource pooling, which allows for pooling of different computing resources to deliver the services to multiple users. It is highly elastic, which allows for quick scalability of resources depending on the demand.

In recent years, personal health record (PHR) has emerged as a patient-centric model of health information exchange. In the existing system they propose a novel ABE-based framework for patient-centric secure sharing of PHRs in cloud computing environments, under the multi-owner settings. To address the key management challenges, they conceptually divide the users in the system into two types of domains, namely public and personal domains. In the public domain, they 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. They propose mechanisms for key distribution and encryption so that PHR owners can specify personalized fine-grained role-based access policies during file encryption. But it has some security issues. In our proposed system we introduce a two-level access control model that combines fine-grained access control, which supports the precise granularity for access rules, and coarse-grained access control, which allows the storage provider to manage access requests while learning only limited information from its inputs. This is achieved by arranging outsourced resources into units called access blocks and enforcing access control at the cloud only at the granularity of blocks. And also our solution handles the read and writes access control. In recent years, personal health record (PHR) has emerged as a patient-centric model of health information exchange. In the existing system they propose a novel ABE-based framework for patient-centric secure sharing of PHRs in cloud computing environments, under the multi-owner settings. To address the key management challenges, they conceptually divide the users in the system into two types of domains, namely public and personal domains. In the public domain, they use multi-authority

In recent years, personal health record (PHR) has emerged as a patient-centric model of health information exchange. In the existing system they propose a novel ABE-based framework for patient-centric secure sharing of PHRs in cloud computing environments, under the multi-owner settings. To address the key management challenges, they conceptually divide the users in the system into two types of domains, namely public and personal domains. In the public domain, they 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. The mechanisms for key distribution and encryption so that PHR

owners can specify personalized fine-grained role-based access policies during file encryption. But it has some security issues. It introduced a two-level access control model that combines fine-grained access control, which supports the precise granularity for access rules, and coarse-grained access control, which allows the storage provider to manage access requests while learning only limited information from its inputs. This is achieved by arranging outsourced resources into units called access blocks and enforcing access control at the cloud only at the granularity of blocks. And also our solution handles the read and writes access control.

2 RELATED WORKS

This paper is mostly related to works in cryptographically enforced access control for outsourced data and attribute based encryption. To realize fine-grained access control, the traditional public key encryption (PKE) based schemes [8], [10] 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 ABE can be used. In Goyal et. al's seminal paper on ABE [11], 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 [12]. A fundamental property of ABE is preventing against user collusion. In addition, the encryptor is not required to know the ACL.

A. Personal Health Record Using ABE

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. To assure the patients' control over access to their own PHRs, it is a promising method to encrypt the PHRs before outsourcing. 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. Ming Li, Shucheng Yu , Yao Zheng and Kui Ren [1] 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, and leverage attribute based encryption (ABE) techniques to encrypt each patient's PHR file. Different from previous works in secure data outsourcing, it focuses 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. This 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.

B. Securing Personal Health Records

M. Li, S. Yu, K. Ren, and W. Lou [14] proposes a novel and practical framework for fine-grained data access control to PHR data in cloud computing environments, under multi owner settings. To ensure that each owner has full control over her PHR data, they leverage Attribute-Based Encryption (ABE) as the encryption primitive, and each owner generates her own set of ABE keys. his way, a patient can selectively her PHR among a set of users by encrypting the file according to a set of attributes, and her encryption and user management complexity is linear to the number of attributes rather than the number of authorized users in the system.

C. Securing Personal Health Records

M. Li, S. Yu, K. Ren, and W. Lou [14] proposes a novel and practical framework for fine-grained data access control to PHR data in cloud computing environments, under multi owner settings. To ensure that each owner has full control over her PHR data, they leverage Attribute-Based Encryption (ABE) as the encryption primitive, and each owner generates her own set of ABE keys. his way, a patient can selectively her PHR among a set of users by encrypting the file according to a set of attributes, and her encryption and user management complexity is linear to the number of attributes rather than the number of authorized users in the system. To avoid from high key management complexity for each owner and user, they divide the system into multiple Security Domains (SDs), where each of them is associated with a subset of all the users. Each owner and the users having personal connections to her belong to a personal domain, while for each public domain they rely on multiple auxiliary Attribute Authorities (AA) to manage its users and attributes. Each AA distributive governs a disjoint subset of attributes, while none of them alone is able to control the security of the whole system. In addition, they discuss methods for enabling efficient and on-demand revocation of users or attributes, and break-glass access under emergence scenarios.

D. Securing The E-Health Cloud

H. Lohr, A.-R. Sadeghi, and M. Winandy [4] proposes general problems of e-health systems and provide a technical solution for the protection of privacy-sensitive data, which has not been appropriately addressed yet for end-user systems. In particular, Their contributions are as follows: They describe an abstract model of e-health clouds, which comprehends the common entities of healthcare telemetric infrastructures. Based on this model, they outline three main problem areas for security and privacy, namely (i) data storage and processing, (ii) management of e-health infrastructures, and (iii) usability aspects of end-users. They present security architecture for privacy domains in e-health systems which leverages on modern security technology of commodity platforms. This architecture extends the protection of privacy-sensitive data from centrally managed secure networks to the client platforms of the end-users. For each application area a separate privacy domain is

established and it is enforced both centrally and locally on each platform.

3 FRAMEWORKS FOR PATIENT-CENTRIC, SECURE AND SCALABLE PHR SHARING

In this section, we describe our novel patient-centric secure data sharing framework for cloud-based PHRsystems. The main notations are summarized in Table 1.

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. A typical PHR system uses standard data formats. For example, continuity-of-care (CCR) (based on XML data structure), which is widely used in representative PHR systems including Indivo [27], an open-source PHR system adopted by Boston Children's Hospital. Due to the nature of XML, the PHR files are logically organized by their categories in a hierarchical way [8], [20].

3.1.1 Security Model

In this paper, we consider the server to be semi-trusted, i.e., honest but curious as those in [28] and [15]. 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.

3.1.2 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. [7] 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 [23]. 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.

3.2 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 (PUDs) and personal domains (PSDs)) according to the different users' data access requirements. The PUDs consist of users who make access based on their professional roles, such as doctors, nurses and medical researchers. In practice, a PUD 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. 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

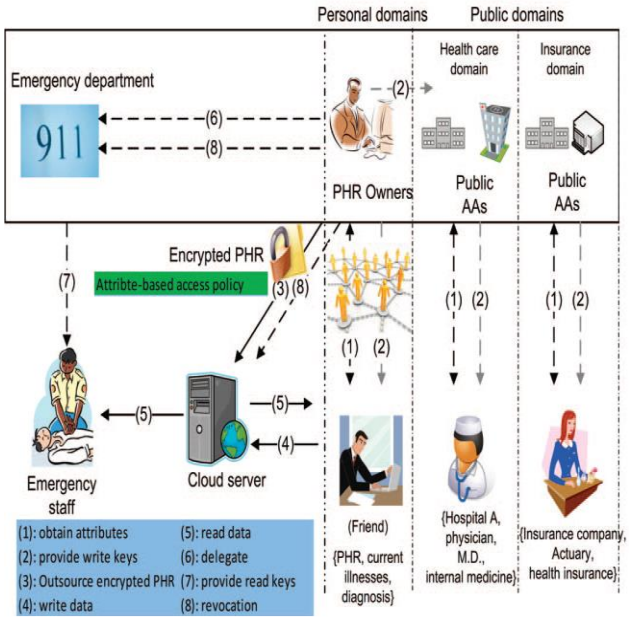

Fig. 1. The proposed framework for patient-centric, secure and scalable PHR sharing on semi-trusted storage under multi-owner settings.

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 patient-centric 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 now is the intrinsic data properties.

3.3 PROPOSED SYSTEM

In this proposed system we introduce the concept of Distributed Attribute-Based Encryption (DABE), i.e., a fully distributed version of CP-ABE, where multiple attribute authorities may be present and distribute secret attribute keys. Furthermore, we give the first construction of a DABE scheme, which supports policies written in DNF; the cipher texts grow linearly with the number of conjunctive terms in the policy. Our scheme is very simple and efficient, demonstrating the practical viability of DABE. We furthermore provide a proof of security in the generic group model; even though this proof is weaker than the proofs of some more recent CP-ABE schemes, our scheme is much more efficient, requiring only O(1) pairing operations during encryption and decryption. The following diagrams shows that the authority to each users. The basic idea behind it is to provide two levels of access control: coarse-grained and fine-grained. The coarse grained level access control will be enforced explicitly by the cloud provider and it would also represent the granularity at which he will learn the access pattern of users. Even though the cloud provider will learn the access pattern over all user requests, he will not be able to distinguish requests from different users, which would come in the form of anonymous tokens. The fine-grained access control will be enforced obliviously to the cloud through encryption and would prevent him from differentiating requests that result in the same coarse-grained access control decision but have different fine-grained access pattern. The mapping between files and access blocks is transparent to the users in the sense that they can submit file requests without knowing in what blocks the files are contained. While most existing solutions focus on read request, we present a solution that provides both read and write access control. Choosing the granularity for the access blocks in the read and write access control schemes affects the privacy guarantees for the scheme as well as its efficiency performance. Advantages of Proposed System

- It provides data confidentiality by implementing a fine-grained and coarse grained cryptographic access control mechanism;
- It supports practical and flexible data sharing scheme by handling both read

and write operations in the access control model.

- It enhances data and user privacy by protecting access control rules and access patterns from the storage provider. It provides data confidentiality by implementing a fine-grained and coarse grained cryptographic access control mechanism
- Benefit from the use of Distributed attribute-based encryption, there is no central authority that is able to maintain all attributes and distribute secret attribute keys.
- It enhances data and user privacy by protecting access control rules and access patterns from the storage provider

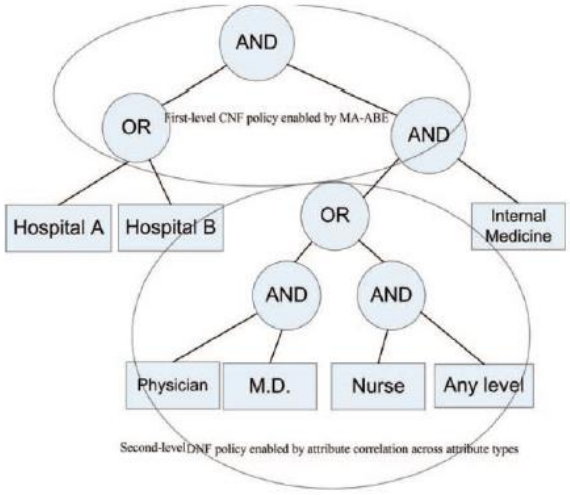

Fig: 2. An example policy realizable using MA-ABE

The DABE Scheme The DABE scheme consists of seven fundamental algorithms: Setup, CreateUser, CreateAuthority, RequestAttributePK, RequestAttributeSK, Encrypt and Decrypt. The description of the seven algorithms is as follows:

- **Setup:** The Setup algorithm takes as input the implicit security parameter 1k. It outputs the public key PK and the master key MK.
- **CreateUser (PK, MK, u):** The CreateUser algorithm takes as input the public key PK, the master key MK, and a user name u. It outputs a public user key PKu, that will be used by attribute authorities to issue secret attribute keys for u, and a secret user key SKu, used for the decryption of ciphertexts.
- **CreateAuthority (PK, a):** The CreateAuthority algorithm is executed by the attribute authority with identifier a once during initialization. It outputs a secret authority key SKa.

RequestAttributePK (PK, A, SKa): The RequestAttributePK algorithm is executed by attribute authorities whenever they receive a request for a public attribute key. The algorithm checks whether the authority identifier aA of A equals a. If this is the case, the algorithm outputs a public attribute key for attribute A, denoted PKA, otherwise NULL.

- **RequestAttributeSK (PK, A, SKa, u, PKu):** The RequestAttributeSK algorithm is executed by the attribute authority with identifier a whenever it receives a request for a secret attribute key. The algorithm checks whether the authority identifier aA of A equals a and whether the user u with public key PKu is eligible of the attribute A. If this is the case, RequestAttributeSK outputs a

secret attribute key SKA,u for user u. Otherwise, the algorithm outputs NULL.

- **Encrypt (PK,M,A,PKA1, . . . ,PKAN):** The Encrypt algorithm takes as input the public key PK, a message M, an access policy A and the public keys PKA1, . . . ,PKAN corresponding to all attributes occurring in the policy A. The algorithm encrypts M with A and outputs the ciphertext CT.
- **Decrypt(PK,CT,A, SKu, SKA1,u, . . . , SKAN,u):** The Decrypt algorithm takes as input a ciphertext produced by the Encrypt algorithm, an access policy A, under which CT was encrypted, and a key ring SKu, SKA1,u, . . . , SKAN,u for user u. The algorithm Decrypt decrypts the ciphertext CT and outputs the corresponding plaintext M if the attributes were sufficient to satisfy A; otherwise it outputs NULL.

Note that this scheme differs from CP-ABE] in that the two algorithms CreateAuthority and RequestAttributePK were added, and CP-ABE's algorithm KeyGen is split up into CreateUser and RequestAttributeSK. It is crucial that RequestAttributeSK does not need any components of the master key MK as input, so that every attribute authority is able to independently create attributes. However, we still require that a trusted central party maintains users (executes CreateUser), as otherwise collusion attacks would be possible.

TABLE I. KEY DESCRIPTION OF HEALTH CARE DOMAIN

Key	Description	Usage
PK	Global Key	Input for all operation
MK	Master Key	Creation of user keys
SKa	Secret Key of Attribute authority a	Creation of attribute key
PKa	Public key of attribute a	Encryption
SKu	Secret key of attribute a for user u	Decryption
PKu	Public key of user u	Key request
SKa,u	Secret key of user u	Decryption

Security Model Setup : The challenger runs the Setup algorithm and gives the global key PK to the adversary.

- The challenger runs the Setup algorithm and gives the global key PK to the adversary.
- The adversary asks the challenger for an arbitrary number of user keys. The challenger calls CreateUser for each requested user and returns the resulting public and private user keys to the adversary. For each user the adversary can request an arbitrary number of secret and public attribute keys, that the challenger creates by calling RequestAttributeSK or Request AttributePK, respectively. Whenever the challenger receives a request for an attribute A of authority a, he tests whether he has already created a secret key SKa for a. If not, he first calls CreateAuthority to create the appropriate authority key (note that SKa will not be made available to the adversary).

IV. CONCLUSION

In this paper, we have proposed a novel framework of secure sharing of personal health records in cloud computing by using DABE. Considering partially trustworthy cloud servers, 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 greatly reduce the complexity of key management while enhance the privacy guarantees compared with previous works. We utilize ABE to encrypt the PHR data, so that the patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications and affiliations. Further enhancement could be done on an existing DABE scheme to handle efficient and on-demand user revocation, and prove its security.

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," in *SecureComm'10*, Sept. 2010, pp. 89–106.

[2] H. L¨ohr, A.-R. Sadeghi, and M. Winandy, Securing the e-health cloud," in *Proceedings of the 1st ACM International Health Informatics Symposium*, ser. IHI '10, 2010, pp. 220–229.

[3] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized private keyword search over encrypted personal health records in cloud computing,"in *ICDCS '11*, Jun. 2011.

[4] "The health insurance portability and accountability act." [Online]. Available:http://www.cms.hhs.gov/HIPAAGenInfo/01 Overview.asp

[5] "Google, microsoft say hipaa stimulus rule doesn't apply to them," http://www.ihealthbeat.org/Articles/2009/4/8/ .

[6] "At risk of exposure – in the push for electronic medical records, concern is growing about how well privacy can be safeguarded," 2006. [Online]. vailable:http://articles.latimes.com/2006/jun/26/health/he-privacy26

[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. 322, no. 7281, p. 283, Feb. 2001.

[8] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient controlled encryption: ensuring privacy of electronic medical records," in *CCSW '09*, 2009, pp. 103–114.

[9] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *IEEE INFOCOM'10*, 2010.

[10] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," in *Journal of Computer Security*, 2010.

[11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *CCS '06*, 2006, pp. 89–98.

[12] M. Li, W. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEEWireless Communications Magazine*, Feb. 2010.

[13] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient evocation," in *ACM CCS*, ser. CCS '08, 2008, pp. 417–426.

[14] 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.

[15] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *ASIACCS'10*, 2010.

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

[17] X. Liang, R. Lu, X. Lin, and X. S. Shen, "Patient self-controllable access policy on phi in ehealthcare systems," in *AHIC 2010*, 2010.

[18] L. Ibraimi, M. Asim, and M. Petkovic, "Secure management of personal health records by applying attribute-based encryption," *Technical Report, University of Twente*, 2009.

[19] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE S& P '07*, 2007, pp. 321–334.

[20] J. A. Akinyele, C. U. Lehmann, M. D. Green, M. W. Pagano,Z. N. J. Peterson, and A. D. Rubin, "Self-protecting electronic medical records using attribute-based encryption," Cryptology ePrint Archive, Report 2010/565, 2010, http://eprint.iacr.org/.