Secure Sharing of Personal Health Records on Cloud Using Key-Aggregate Cryptosystem

Anilkumar K B[1], Mr. Prasanna Kumar M J[2],
[1] *PG Student,* [2]*Asst. Prof., Dept. of CS&E,*
BGSIT, B.G Nagar, Mandya -571418

Abstract— In modern health care environments, personal health record (PHR) owners store and share their PHR data via a cloud because of its on-demand resource access, measured service and rapid elasticity. To enable secure and flexible data sharing in the cloud, efficient management of encryption keys is required. Selective data sharing requires different documents to be encrypted with different keys, which implies data users like doctors to securely store the received keys and submit an equal number of keyword trapdoors (encrypted queries) to the cloud server to perform a keyword search over authorized encrypted files. The current work focuses on reducing key-size by generating a single aggregate key, but does not provide searchable encryption, which is required for flexible data sharing. Our proposed scheme addresses this issue by enabling a patient to distribute a single constant-size aggregate key to a data user for sharing a large number of documents and then user submits a single aggregate trapdoor to the cloud for searching over authorized encrypted documents. The novelty of this scheme lies in submitting a single trapdoor for keyword search over documents encrypted with different keys as opposed to traditional methods requiring submission of multiple trapdoors. Performance evaluation confirms that our proposed scheme is practically efficient and also reduces storage overhead by reducing both the number of keys and key-size without affecting security-level, which is highly desired in the resource constraint devices like smartphones.

Keywords—Personal Health Records, secure data sharing, key-aggregate encryption, searchable encryption, cloud storage

I. INTRODUCTION

Cloud systems can be used for providing data sharing functionality because of its ubiquity, convenient and on demand access facilities [1]. So, it is used by Personal Health Record (PHR) owners to store their PHR data on the cloud [2] and remove the geographical dependence between health care provider and patient [3]. The cloud, because of its data outsourcing feature has many privacy and security issues. So PHR owners encrypt their sensitive data before outsourcing it to the cloud and hence the data remains secure against the cloud provider and other malicious users. But data encryption makes searching and retrieving only the selected data containing given keywords a challenging task. Selective data sharing (e.g., sharing medical information with doctors and nurses, sharing of insurance policy information with insurance broker) demands different encryption keys to be used for different files and thus requiring a large number of keys to be

Distributed to users for searching over encrypted files and decrypting them.

Existing searchable encryption schemes require the client to provide a cloud server with a trapdoor (search query) encrypted with every key that a matching document would be encrypted with, and hence the number of trapdoors increases with the number of documents to search. This implies the need for secure communication, storage and computational complexity, making it inefficient in multitenant, ubiquitous and an elastic model like a cloud.

So our aim is to develop an efficient patient-centric data sharing scheme that enables patients to have complete control over their PHRs and manage the keys efficiently. This scheme also aims to enable data users (doctors, nurses) perform a keyword search over authorized encrypted files using single aggregate trapdoor generated from the single aggregate key. of the PHR owner and it is therefore difficult to correctly verify who signed the PHRs.

Kuo et al. [2] developed a patient centric access control scheme for Personal Health Records in the cloud. To achieve this, PHR owner divided his data into various files and produced different encryption keys for each file category. This scheme is based on symmetric encryption scheme which requires the data owner (patient) to transfer encryption keys every time data user (doctor) wants to access or update the record. This makes the system inefficient in a cloud environment. Besides this, number of encryption keys are as many as the number of file categories creating communication and storage overhead.

Chu et al. [1] developed a scheme to allow encrypting a set of documents with different keys, but be decrypted with a single aggregate key. This scheme is based on a public key cryptosystem and takes into account cipher text

class during encryption. The scheme [1] enables efficient delegation of decryption rights for any set of cipher texts using a single aggregate key and is the main inspiration of our work. But it does not support keyword search over the encrypted data which is required to achieve secure and selective data sharing in the cloud.

So we develop a patient-centric, secure data sharing scheme that enables the patients to efficiently delegate decryption rights for any set of cipher text classes using a single compact key, but encompassing the power of all the keys being aggregated. This proposed scheme is based on an asymmetric key cryptosystem and enables data users like a doctor or nurse to generate single trapdoor by encrypting search query with single compact key used for searching over encrypted data as opposed to traditional system requiring users to submit multiple trapdoors for searching data encrypted with different keys.

III. PRELIMINARIES

In this section, we describe the key-aggregate cryptosystem [1] developed by Chu et al. for patient-controlled encryption.

This scheme consists of five algorithms as follows:

- Setup (1^k, n):This algorithm is run by the patient to set up the scheme. It takes a security parameter and the number categories of PHR n as input and outputs the public parameter param.

- KeyGen:This algorithm is run by the patient to randomly generate a public/master-secret key pair (pk,
  msk) for each PHR category.

- Encrypt (pk, i, record):This algorithm is run by patient/ data users who wants to encrypt record.It takes a public key pk, an index i (cipher text class), and a record R, and outputs an encrypted record ER.

- Extract (msk, S):This algorithm is run by the patient for generating aggregate key to delegate decryption rights for certain set of ciphertext classes to delegatee. It takes the master secret key msk and a set S of indices□

- corresponding to different categories of PHR as input, and outputs the aggregate key KS . Decrypt (KS, S, i, ER):This algorithm is run by a data user like doctor who have received an aggregate key KS generated by Extract.It takes the KS, set S, an index i denoting ciphertext class of an encrypted record, and.

ER, and outputs the decrypted result R if i∈S.

Although the above scheme [1] allows public -key PCE for flexible hierarchy, it does not support any search over encrypted data which is extremely important for secure and flexible data sharing in pay-as- you use model like a cloud. E.g., consider a scenario where patient delegates decryption rights to 100 different categories to the doctor. But at particular time doctor may be interested in examining a few categories only. So to download only records corresponding to selected categories, keyword search using the aggregate key is extremely important. Our proposed scheme provides this feature and requires a more complex mathematical transformation to support keyword ciphertext encryption, aggregate trapdoor generation and keyword matching.

IV. PROPOSED SCHEME

A. System Overview

Our proposed system is based on multi-key searchable encryption scheme [19] and key-aggregate cryptosystem [1]. To enhance security and flexibility in data sharing, each PHR category is encrypted with different public keys (e.g., category

Medical Info is encrypted with public key pk1,Category

Dental Info is encrypted with public key pk2 etc.) . So in such scenario using traditional approach patient has to send all the

encryption keys (K1, K2, ... Kp) to different data users like doctors, nurses, insurance policy broker etc. depending on their access right. After receipt of these keys, data owner has to store them and then generate keyword trapdoors (encrypted

search query) using all these keys (E (K1, search query), E(K2, search query)... E (Kp, search query)) to perform keyword search.

As shown in the Fig.1 (a), Patient is having a Personal Health Record set $\{recordi\}^n i=1$ and each recordi is encrypted with an encryption key ki.

Suppose Patient wants to share p documents $\{recordi\}^p i=1$ with the doctor. In this scenario, Patient has to send all the encryption keys {ki} to the doctor and when doctor wants to retrieve records containing a keyword w, he has to generateTrapdoori for every recordi encrypted with encryption key ki and submit all the trapdoors $\{Trapdoori\}^p i=1$to the cloud server for query search.

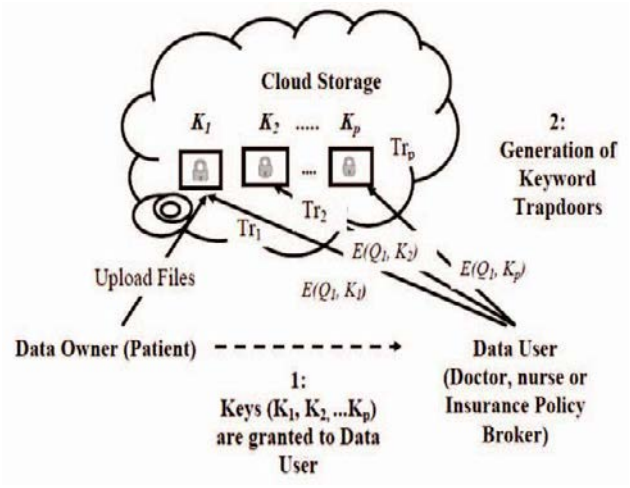

Fig. 1. Traditional approach for secure data sharing on cloud

So in this paper, we propose a novel approach of aggregate trapdoor generation as shown in Fig. 2. where, Patient distributes a single aggregate key for sharing p records with a doctor or nurse and data user (doctor or nurse) will generate a single keyword trapdoor instead of $\{Trapdoori\}^{p}i=1$ for searching over encrypted cloud data. Thus, in proposed scheme delegation right for keyword search and decryption of p records is achieved by sharing a single aggregate key, instead of p keys.

B. Construction of the Proposed Scheme

Our proposed scheme consists of three roles, such as multiple patients, a cloud service provider (CSP), and multiple data users like doctors, nurses and insurance brokers. Following are the assumptions:

The CSP is semi-trusted by the patient (data owner). Fig. 2. below shows the framework of our proposed scheme. It consists of eight algorithms as follows:

- Setup (1^k, n):
  This algorithm is used by a data owner to set up an account on an untrusted cloud server. It takes security parameter (k) and maximum number of categories of PHR that he will upload on cloud server (n) as input to generate set of public keys and master keys. This algorithm also publishes the system parameter param.

  1) System Parameter (param) = $\{public\ keys\}^{n}i=1$
  2) Security Parameter determines the size of public and master secret key (e.g., if k=high, size of public key = 1024 bits and size of master key = 512 bits).
  3) public keys = {p1, p2, ...pn} and pi ∈ {e1, e2, ...en}
  4) master keys={m1, m2, ...mn} and mi∈ {e1, e2,
     ...en} where, 1 < e <

     φ(n)suchthat,
     φ(n) = (p-1) (q-1) and n=p. q

     Here, p and q are both prime numbers.

- Key Generation (pk, msk):

  This algorithm is used by data owner (Patient) to generate the private key as (1). The data owner divides his PHR into different categories to produce category names and private keys for each PHR category as follows:

  1) The private key (Kj) is generated by obtaining the public key (pkj) and master secret key (mskj) as follows:
     Kj= (pkj .mskj) (1)

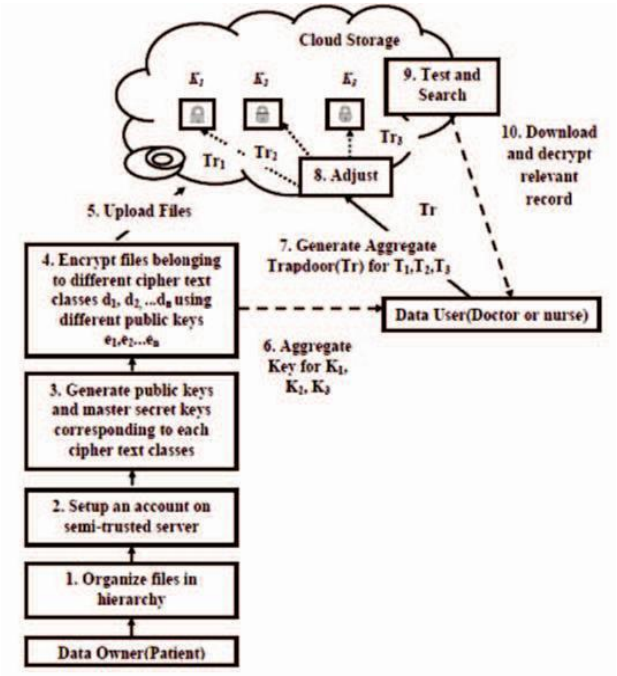

Fig. 2. Proposed framework of secure sharing of personal health records using key-aggregate cryptosystem

- Encryption (pk, msk, Cj):
  This algorithm is used by patient to encrypt a text or image file identified by its category name (Cj) along with its keyword. Data owner generates data cipher text by encrypting it's contents with public key (pkj) and generates keyword cipher text by encrypting it's

Adjust (j, S, Tr):

This algorithm is used by a cloud server to generate right trapdoor Trj as (5).

1) The cloud server obtains set of authorized documents (S) based on the role of data users. E.g. if data user is a doctor, S = {i1, i2, i3} as mentioned in Extract phase.

2) The cloud server obtains product of all master secret keys other than search query category. E.g. if Aggregate trapdoor is generated by w=Laboratory Test Reports (i1) then product (P) obtained from Data owner (Patient) will be:

$$P = (mski2.\ mski3.\ mski4) \quad (4)$$

V. IMPLEMENTATION DETAILS

A. Platform and Technology

The experimental setup of the proposed system is

configured using OpenStack, an open source cloud platform

using DevStack module. We use the following platform and

technology:

O.S: Ubuntu 12.04

Database: OpenStack MySQL

Web Server: Apache

Tomcat Email Server: Hmail

Email Client: Thunderbird

Hypervisor: KVM

contents with master key (mskj) when uploading j^{th} document on cloud server where $j \in \{1, 2, ...n\}$.

- Extract(msk, S):

This algorithm is used by patient to generate an aggregate encryption key using a master secret key as follows:

1) Patient, based on the roles of different data users (doctors, insurance broker, etc.) generates different sets S1, S2...Sn such that, S1 ∈ Doctor,

S2 ∈Insurance Broker etc.

Here, S represents indices of different PHR categories to which access right is to be granted.

2) Patient obtains all master secret keys corresponding to different categories in S and generates an

aggregate key (Kagg) as in (2). E.g., suppose Patient (data owner) wants doctors to access PHR records corresponding to categories like Laboratory Test Reports (i1), Allergies Info (i2), Current

Medications (i3) and X-rays (i4). Hence, S= {i1, i2, i3, i4}.

In this case, it will first obtain mski1, mski2, mski3, mski4 and generates aggregate key as follows:

$$Kagg = SHA\text{-}256(\Pi i \in S\ (msk)i) \quad (2)$$

This key is used to delegate keyword search and the decryption right to a data user, and is provided to him via email.

- Aggregate Trapdoor (Kagg, w):

This algorithm is used by a data user (doctor or nurse) to generate an aggregate trapdoor to obtain all encrypted files corresponding to a particular file category. This algorithm uses an aggregate key (Kagg) obtained from trusted third party by verifying the identity of data user based on constant-size aggregate key and category name (keyword w) as input.

$$Tr = E\ (Kagg, search\ query\ (w)) \quad (3)$$

Laboratory Test Reports (i1) then product (P) obtained will be as (4).

Although a data user has known P, he cannot get each value of the multiplier (msk) from a product.

2) The data user obtains secret key as follows:

private keyj = (Kagg / Π i□j (msk) i). Pkj (8)

In the above example private key for decrypting category Laboratory Test Reports is obtained by:

private keyj = (Kagg / P). Pkj (9)

The data user obtains plain text as follows:

plain text = (cipher text) / (private keyj) (10)

3) Although Cloud sever has known P, he cannot get each value of the multiplier (msk) from a product. The cloud server then generates right trapdoor by:

Trj = Tr / (Πi ∈S(msk)i□j) (5)

In the above example the cloud server obtains the right (adjusted) trapdoor as follows:

((Search query (w). (Π□msk□i=1)) /(Π□msk□i=2) (6)

This algorithm enables searching of documents encrypted with different keys.

- Test (Trj, cw):

This algorithm is used by a cloud server to perform a keyword search over the j^{th} document. For j^{th} document, this algorithm takes as input the adjusted trapdoor and keyword cipher text and outputs true or false by judging:

cw == Trj (7)

- Decrypt(Kagg, S, j, Cj , Pj):
  This algorithm is used by data user who has received an aggregate key and obtains plain text as (10).

1) Data user (doctor) obtains product of all master secret keys other than file categorytobedecrypted.E.g. if the file category to be decrypted isStorage: OpenStack Swift (Object Storage) Language: Java (J2EE)

Web browser: Mozilla Firefox, Google Chrome.

B. Datasets

We conducted experimental evaluation of the proposed system on a real world personal health record dataset available at [22]. We selected the records and arranged them into a tree structure as per the requirements of different patients.

C. Performance Evaluation

Fig. 3. illustrates the number of keys to be granted for different approaches in the case of 511 records.

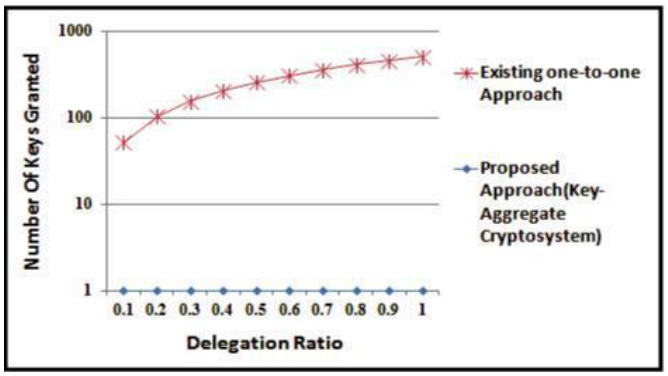

Fig. 3. Number of keys granted in different approaches

The X -axis represents delegation ratio and the Y-axis represents the number of keys granted for different approaches. Here, delegation ratio=No. of categories to which access right is provided/Total no. of categories.

Fig. 4. illustrates the number of trapdoors generated in different approaches in the case of 625 documents. The X-axis represents the total number of documents and the Y-axis represents the total number of trapdoors.

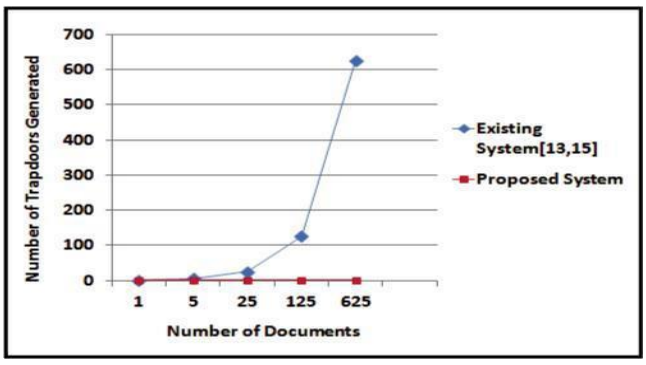

Fig. 4. Number of trapdoors generated in different approaches

Execution time of Test algorithm executed at the server side, increases linearly as the number of documents increases. When the maximum number of documents is 625, total execution time is 2000 milliseconds.

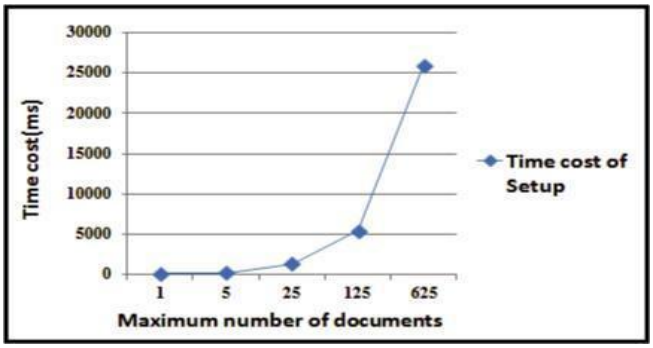

Fig. 5 (a) Execution time of Setup algorithm

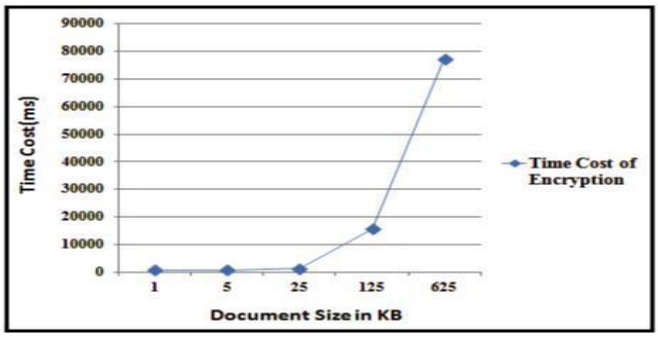

Fig. 5 (b) Execution time of Encryption algorithm

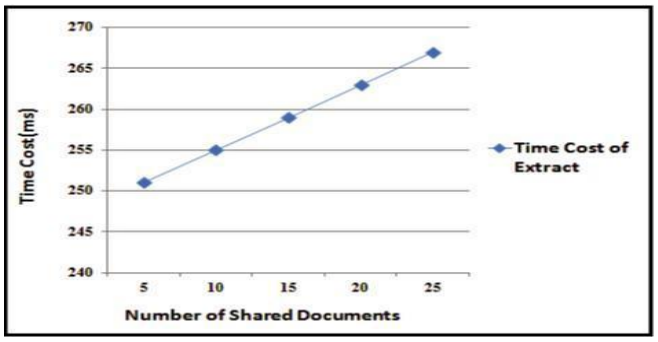

Fig. 5 (c) Execution time of Extract algorithm

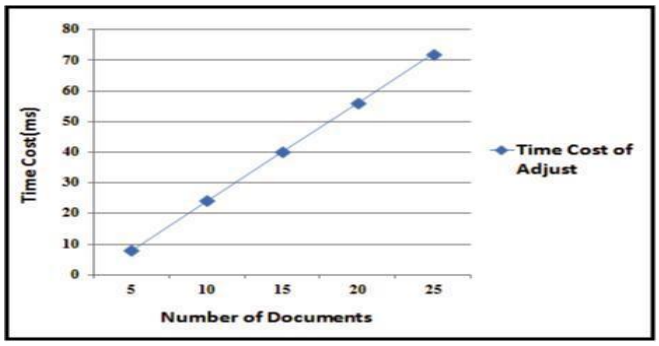

Fig. 5 (d) Execution time of Adjust algorithm

VI. CONCLUSION AND FUTURE SCOPE

Considering the practical problem of privacy-preserving data sharing in modern health care environments, we have proposed an asymmetric-key cryptosystem based on flexible hierarchy to enable patients to exercise complete control over their personal health records (PHR) in the cloud. Using our scheme the patient only needs to distribute a single aggregate key when sharing a large number of documents with the different users like doctors, nurses, etc. and they only need to submit a single aggregate trapdoor while querying over all the PHRs shared by the same patient. Performance evaluation results confirm that our work can provide an effective solution for secure and scalable data sharing system based on the cloud storage.

Our proposed scheme generates a single aggregate trapdoor under multi-key setting for single keyword search, so to reduce the number of trapdoors for multi - keyword search using key aggregate cryptosystem is a future work. Federated clouds are getting popular nowadays, but our scheme cannot be directly applied. So it is also a future scope to apply this scheme in the federated cloud environment.

REFERENCES

[1] C. Chu, S. Chow, and W. Tzeng, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage," *IEEE Transactions on Parallel and Distributed Systems*, 2014, 25 (2): 468-477.

[2] Kuo-Hsuan Huang, En-Chi Chang, and Shao-Jui Wang, "A Patient Centric Access Control Scheme for Personal Health Records in the Cloud, " *Fourth International Conference on Networking and Distributed Computing*, IEEE 2014.

[3] Wu R (2012), "Secure sharing of electronic medical records in cloud computing, " *Arizona State University*, ProQuest Dissertations and Theses.

[4] Leng, C., Yu, H., Wang, and J., Huang, "Securing Personal Health Records in the Cloud by Enforcing Sticky Policies," TELKOMNIKA Indonesian Journal of Electrical Engineering, 11 (4), 2200-2208, 2013.

[5] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records, " Proc. ACM Workshop Cloud Computing Security (CCSW 09), pp. 103-114, 2009.

[6] Chen Danwei, Chen Linling, Fan Xiaowei, He Liwen, Pan Su, and Hu Ruoxiang "Securing Patient-Centric Personal Health Records Sharing System in Cloud Computing," China Communications, Supplement No.1, 2014.

[7] Ming Li, Shucheng Yu, and Yao Zheng, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption, " IEEE Transactions on Parallel and Distributed Systems, 24 (1), pp. 131-143, 2013.

[8] 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.

[9] M. Chase, and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption, " Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009.

[10] R. Canetti, and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption, " Proc. 14th ACM Conf. Computer and Comm. Security (CCS 07), pp. 185-194, 2007.

[11] C. K. Chu, and W. -G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles, " Proc. Information Security Conf. (ISC 07), vol. 4779, pp. 189-202, 2007.

[12] C. K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng,"Conditional Proxy Broadcast Re-Encryption, " Proc. 14th Australasian Conf. Information Security and Privacy (ACISP 09), vol. 5594, pp. 327-342, 2009.

[13] S.S.M. Chow, J. Weng, Y. Yang, and R.H. Deng, "Efficient Unidirectional Proxy Re-Encryption, " Proc. Progress in Cryptology (AFRICACRYPT 10), vol. 6055, pp. 316-332, 2010.

[14] J. W. Li, J. Li, and X. F. Chen, "Efficient Keyword Search overEncrypted Data with Fine-Grained Access Control in Hybrid Cloud, "In: Network and System Security 2012, LNCS, pp. 490-502, 2012.

[15] Z. Liu, Z. Wang, and X. Cheng, "Multi-user Searchable Encryption withCoarser-Grained Access Control in Hybrid Cloud, " Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, pp. 249-255, 2013.

[16] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing, " Proc. IEEE INFOCOM, pp. 534-542, 2010.

[17] F. Zhao, T. Nishide, and K. Sakurai, "Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control, "Information Security and Cryptology, LNCS, pp. 406-418, 2012.

[18] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys, " Proc. Advances in Cryptology Conf. (CRYPTO 05), vol. 3621, pp. 258-275,2005.

[19] R. A. Popa ,and N. Zeldovich, "Multi-key searchable encryption, " Cryptology ePrint Archive, Report 2013/508, 2013.

[20] J. Benaloh, "Key Compression and Its Application to DigitalFingerprinting," technical report, Microsoft Research, 2009.

[21] T. Okamoto and K. Takashima, "Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption, "Proc. 10th Intl Conf. Cryptology and Network Security (CANS 11), pp. 138-159, 2011.

[22] Department of Veterans Affairs, "VA Personal Health Record Non