International Journal of Technology and Engineering System (IJTES)
Vol 7. No.4 2015 Pp. 278-282
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

A RAID-BASED SECURE STORAGE OF PERSONAL HEALTH RECORDS IN CLOUD COMPUTING USING MA-ABE

[1]P.SIVAKUMAR [2]K.DEVI [3]V.DEEPALAKSHMI
[1,3]PG scholar, [2]Assistant Professor
Department of Computer Science and Engineering, Valliammai Engineering College,
Chennai, Tamilnadu, India.
[1]vvsiva.p@gmail.com [2]devii.jeya@gmail.com [3]deepavijay.kpm@gmail.com

ABSTRACT

Personal health record (PHR) is an emerging patient-centric model of health data exchange, which is often deployed to be stored at a third party, such as cloud providers. To assure the patients' control over access to their own PHRs, Multi-Authority Attribute Based Encryption is an efficient method to encrypt the PHRs. PHR in cloud is difficult to maintain, because of security issues in performing the read and write operation of patient records. A Multi-Authority Attribute Based Encryption (MA-ABE) technique is proposed to encrypt each PHR file. A high level of patient privacy is guaranteed by exploiting Multiple-Authority Attribute Based Encryption. It also provides security and scalability. To achieve the availability, trustworthiness and confidentiality of the data stored in the cloud, the proposed system encrypts user's data and makes use of the RAID (Redundant Array of Independent Disks) technology principle to manage data distribution across cloud storage providers. The disk failures are avoided using XOR (Exclusive OR)-parity check in RAID level 5. RAID level parity (RAID 5) uses an erasure code to generate parity information at the block level or bit level. If any data is lost due to disk failure, the parity information is used to reconstruct the lost data in the failed disk. The proposed system implements the RAID concept for PHR storage model in cloud computing to provide key management and data restoration.

Keywords:--Cloud Computing, ABE, MA-ABE, Personal Health Records, RAID.

I . INTRODUCTION

Cloud Computing turn into prevalent, more and more susceptible information are being centralized interested in the cloud, such as emails, personal health records, government documents, etc. In recent years, personal health record (PHR) has appeared as a patient-centric model of health information swap over [7]. It lets a patient to make, handle, and organize his/her personal health data in one place during the web, which has finished the storage space, retrieval, and distribution of the health information more efficient. Each patient has assured the full control of his/her medical records and can share their health data with broad range of users, as well as healthcare providers, family members or friends. PHR owners have to decide regarding encryption of files as well as access security to users. Users like family members and friends can access PHR file with equivalent decryption key. The authorized users like medical doctors, pharmacists and researchers could either need to access the PHR for personal use and for some specialized reasons too. In order to keep personal heath data stored on semi trusted servers, this paper obtain on Multi Authority attribute based encryption (MA-ABE) as the major encryption primitive. Using MA-ABE, patient be able to selectively distribute his/her PHR among set of users by encrypting files under a set of attributes without knowing complete list of users.

1.1 Redundant Array of Inexpensive Disks(RAID) RAID allows information to access several disks. RAID uses following techniques *disk striping* (RAID Level 0), *disk mirroring* (RAID Level 1), update the parity (RAID Level 4), and *disk striping with parity* (RAID Level 5), striping with double distributed parity (RAID Level 6) and combine mirroring and stripping (RAID Level 10) [17].We are using RAID level 5 method in cloud computing to achieve redundancy, lower latency, increased bandwidth, and maximized ability to recover from hard disk crashes. The RAID 5 model is used to

partition the storage and duplicate the original data in different disks. RAID consistently distributes data across each drive in the array. RAID then breaks down the data into consistently-sized chunks (commonly 32K or 64k, although other values are acceptable). When the data is read, the process is reversed, giving the illusion that the multiple drives in the array are actually one large drive. *Level 5* is the most common type of RAID. By distributing parity across some or all of an array's member disk drives, RAID level 5 eliminates the write bottleneck inherent in level 4. The only performance bottleneck is the parity calculation process. With modern CPUs and Software RAID, that usually is not a very big problem. As with level 4, the result is asymmetrical performance, with reads substantially outperforming writes. Level 5 is often used with write-back caching to reduce the asymmetry. The storage capacity of Hardware RAID level 5 is equal to the capacity of member disks, minus the capacity of one member disk. The block-interleaved distributed-parity disk array eliminates the parity disk bottleneck present in the block-interleaved parity disk array by distributing the parity uniformly over all of the disks [1]. An additional, frequently overlooked advantage to distributing the parity is that it also distributes data over all of the disks rather than over all but one. This allows all disks to participate in servicing read operations in contrast to redundancy schemes with dedicated parity disks in which the parity disk cannot participate in servicing read requests. Block-interleaved distributed-parity disk array have the best small read, large write performance of any redundancy disk array. Small write requests are somewhat inefficient compared with redundancy schemes such as mirroring however, due to the need to perform read-modify-write operations to update parity. This is the major performance weakness of RAID level 5 disk arrays.

1.2Personal Health Record

The Public health operational group explains PHR as: an electronic function through which individuals can access, handle and distribute their health information, and that of others for whom they are certified, in a private, secure, and confidential environment .The Personal Health Record (PHR) is an Internet-based set of tools that let people to contact and synchronize their lifetime health information and build suitable parts of it available to those who want it. *PHRs* present an integrated and complete view of health information, including information people make themselves such as warning signs and medicine use, information from doctors such as analysis and test results, and information from their pharmacies in addition to insurance companies. Individuals access their PHRs by means of the Internet, via state-of-the-art security and privacy controls, at some time as well as from every location. Family members, doctors or school nurses can observe parts of a PHR when essential and emergency room staff can retrieve vital information from it in an emergency [7]. People are able to use their PHR as a communications hub: to send email to doctors, move information to experts, obtain test outcome and access online self-help tools. PHR connects each of us to the unbelievable possible of current health care and provides us control over our own information.

II . RELATED WORKS

Using ABE, access policies are expressed based on the attributes of users, which permits a patient to selectively split her PHR among a set of users by encrypting the file under a set of attributes, without the need to know a whole list of users. Expanding the capacity of a RAID-5 array with adding of disks, data have to be relocated between disks to leverage extra space and performance gain.

Drawbacks

- The saved data's are in encrypted format by means of public key encryption.
- It provides a smaller amount security for data's.
- RAID-5 scaling is restricted by preserving a round-robin data distribution after adding disks.

2.1 Attribute Based Encryption (ABE)

By means of ABE, access policies are expressed based on the user's attributes that enables a patient to selectively distribute their PHR amongst a set of users by encrypting the file under a set of attributes. The difficulties of encryption and decryption methods are greatly reduced using Attribute Based Encryption technique. But, to integrate ABE into a large-scale PHR system is not easy due to the on-demand revocation, key management scalability and dynamic policy updates. [13]Cipher text-Policy Attribute-Based Encryption (CP-ABE) allows encrypting data under an access policy, specified as a logical combination of characteristics. Such cipher texts can be decrypted by anyone with a set of attributes that fits the policy.

2.2 Attribute Based Access Control

In PHR file, access methods are decided by the PHR Owner for particular actor usages. This model follows the PHR Owner have all access rights, Doctors had read and write access controls, but the patient, insurance clients and patient family members have only read access. File should be uploaded and maintained by PHR owner. The modifications of patient's records are done by doctors.

2.3Secure Attribute Based Structures

Attributes describe, categorize, or interpret the datum to which they are assigned, but the traditional attribute architectures and cryptosystems are unprepared to provide security in the face of various access requirements and environments. [11]M.Pirretti introduces a novel secure information management architecture based on emerging attribute-based

encryption (ABE) primitives and demonstrated a policy system that meets the requirements of complex policies. Cryptographic optimizations that greatly improve enforcement effectiveness were offered based on the needs of those policies.

2.4 Key Policy Access Control

Compared with the threshold policy a fine-grained access control can be achieved in key policy methods. The ciphertexts labeled with a set of attributes and private keys are associated with a more generalized access control structure in key policy methods. One of the disadvantages of key policy is that since access policies are built into users' private keys, the data owner has limited control over who can decrypt the data. In addition, the data owner must trust the key issuer and employ a trusted server to store all of the expressive access structure in plaintext. Goyal et al, proposed the first key-policy scheme (KP-ABE)[16] in 2006.

2.5Ciphertext Policy Access Control

Ciphertext-policy is another finegrained access control. However, attributes in ciphertext-policy are associated with keys while access structures are embedded into the ciphertext. With this manner, the data owner can determine who can decrypt. Moreover, if the policy needs to be updated frequently, the ciphertext-policy can be more flexible since the data owner only needs to update the access structure in the ciphertext. This makes ciphertext-policy closer to Role Based Access Control (RBAC). In threshold-policy and key-policy, collusion resistances are ensured by using a secret sharing scheme (SSS) with a random polynomial for each private key. However in ciphertext-policy, SSS no longer holds since the access structure is moved away from the key and the ciphertext is only left with attributes. The Ciphertext-policy must utilize a two-level random masking methodology, which makes use of groups, with efficiently computable bilinear maps, to randomize the private key[15].

2.6 RAID Based Parity Method

In a bit-interleaved, parity disk array, data is conceptually interleaved bit-wise over the data disks, and a single parity disk is added to tolerate any single disk failure. Each read request accesses all data disks and each write request accesses all data disks and the parity disk [1]. Thus, only one request can be serviced at a time. Because the parity disk contains only parity and no data, the parity disk cannot participate on reads, resulting in slightly lower read performance than for redundancy schemes that distribute the parity and data over all disks. Bit-interleaved, parity disk arrays are frequently used in applications that require high bandwidth but not high I/O rates.Architecture diagram shows the relationship between different components of system. This diagram is very important to understand the overall concept of system. The proposed system architecture diagram shows the cloud registration using service directory, and gets access from cloud provider pool.

III . PROPSED SYSTEM

3.1 Problem Definitions

This paper describes a PHR system where there are several PHR owner, PHR consumers, and RAID storage models in cloud computing. This paper also recommends a narrative ABE-based skeleton for patient-centric secure distribution of PHRs in cloud computing surroundings, under the multi-owner backgrounds. To deal with the key management challenges, we abstractly separate the users in the system into three types of domains namely Public, PHR Owner and Emergency domains. In the public domain, we make use of multi-authority ABE (MA-ABE) to get better the security and keep away from key escrow trouble. The disk striping and rotated parity, RAID-5 achieves high performance, large capacity, and data reliability. To gain a uniform data distribution, the minimal fractions of data blocks and parity blocks to be moved for RAID-5 scaling are identical to the percentage of new disks.

Advantages:

- The complexities in encryption, key generation and decryption are reduced by using MA-ABE.
- XOR-parity check used to avoid of data disk failures and make the restore of the data using parity disk.

The PHR owners know how to identify personalized role-based access guidelines at some stage in file encryption. The Architecture Diagram for Proposed system is given in Fig 1

The PHR Owner provides public and private keys for accessing the PHR records by users, doctors and patients. The emergency department needs to view the PHR records means the permission get from any one authorized user with the identification of emergency clients. The emergency department can read the PHR records; they cannot modify any document of the PHR details. The service directory used to register the cloud for new users and update the data's in cloud memory using RAID-5 techniques.

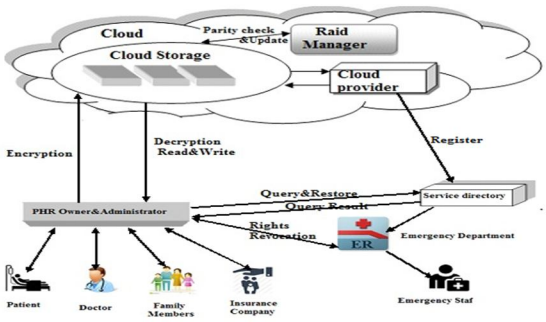

Fig 1: Architecture Diagram

3.2 MA-ABE Security implementation for PHR:

This system makes use of MA-ABE algorithm to provide security to the PHR in cloud. The authority attributes in this system includes hospital admin, patients, doctors, family members, hospital staffs and emergency clients. The encryption and decryption is carried out based on the access rights (read and write) of the authorities. Fig 2 explains the steps involved in the MA-ABE algorithm.

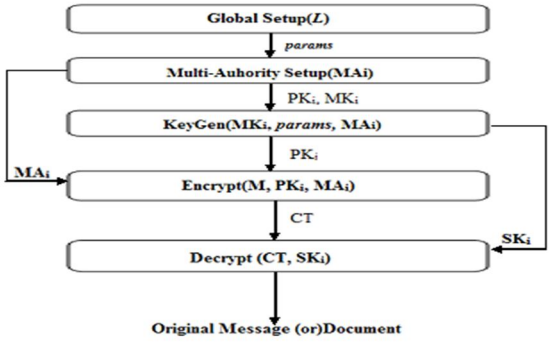

Fig 2 MA-ABE implementation flow

The MA-ABE algorithm used in this system involves four main steps viz. initializing the setup, generating secret key, Encryption and decryption. The detailed explanations of these steps are given below.

Setup:
The setup algorithm takes data and security parameters as inputs and gives Public key (PK_i) and Master key(MK_i) as outputs.

1. Global setup algorithm takes as input, Public parameter ***L*** and outputs the system parameters *params*.

Global Setup(1^L) ->params.

2. In authority setup algorithm, Multi Authority (MA_i) generates their own Public-Master key pair (PK_i,MK_i)

Multi-Authority Setup(MA_i)->(PK_i, MK_i).
Where **i** ranges from 1, 2, . . . , N.

Generating Secret Key:
The key generation algorithm takes the input public parameter value *params,*master key of attribute and multi authority structure and outputs the Secret Key(SK_i), Public Key(PK_i) for each attributes.

KeyGen(MK_i, *params*, MA_i) -> (PK_i,SK_i)

*Encryption:*This method take the input message(M), Multi Authority attribute (MA_i) and each authority's Public key (PK_i) to be encrypted with and outputs the cipher text(CT).

Encrypt(M, PK_i, MA_i) -> CT

Decryption:
This decryption algorithm takes cipher text (CT) and the secret key (SKi) as input and gives out the original message (M) as output.

Decrypt (CT, SK_i) -> M

3.3 Ex-OR parity algorithm for RAID-5:
Parity is a form of error detection that uses a single bit to represent the odd or even quantities of `1's and `0's in the data. Parity usually consists of one parity bit for each eight bits of data, which can be verified by the receiving end to detect transmission errors.

If ((a=0)&&(b=0))&&((a=1)&&(b=1))
Parity doesn't check the data
Otherwise
Parity checks and reloads the data.

After checking the parity if it results in disk failure, the parity updating model retrieves the losing data. The disk striping and rotated parity, RAID-5 achieves high performance, large capacity, and data reliability. Parity across the array is computed using the XOR (Exclusive OR) logical operation (fig 2). XOR parity is a special kind of erasure code. N blocks of data are transformed into N+M blocks such that upon loss of any M blocks of data, they can be recovered from the remaining N blocks, irrespective of which blocks are lost. For example, if the parity information of block A is stored in block C, parity information of B block is stored in D block, similarly D block's parity in A and parity of C in B. So if any data is lost in any of these blocks it can be easily regained from the other blocks using the parity information stored in it.

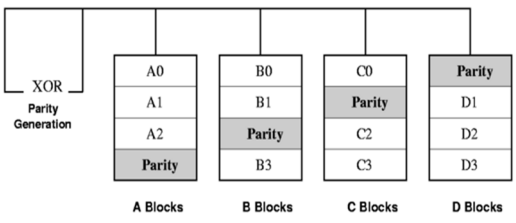

Fig 2: RAID-5: Independent data disks with distributed parity

The parity information in RAID can either be stored on a separate, dedicated drive, or be mixed with the data across all the drives in the array. Most RAID schemes are designed to operate on fail-stop disks. Any disk failure in RAID (including the parity disk) can be recovered from the remaining disks by just performing an XOR on their data.

3.4 Parity-based Updating algorithm for RAID

Updating Ratio = (total num of blocks should be updated) / (total num of Original blocks).
= m/(n+m).
Where, **m** is the number of new blocks.
n is the number of existing blocks.

The failed disk is identified and it is replaced by the new disk. The data of the failed disk is retrieved using the parity algorithm and placed into the new disk.

V . CONCLUSION AND FUTURE WORK

In this paper, a system of secure sharing of personal health records in cloud computing using MA-ABE is discussed. This system addresses the unique challenges brought by multiple PHR owners, users and greatly reduces the complexity of key management. Data reliability is ensured by RAID-5 by maintaining the parity information as the XOR sum of all the data blocks in a stripe and some blocks are copied in a stripe, without need of erasing old blocks. In the future, we will focus on the RAID-6 model for improving efficiency, reliability of storage and double parity in cloud computing.

REFERENCES

[1] *Guangyan Zhang, Weimin Zheng, and Keqin Li, "Rethinking RAID-5 Data Layout for Better Scalability", IEEE Transactions on computers, vol. 63, Aug 2014.*

[2] *Jiguang Wan, Jibin Wang, Changsheng Xie, and Qing Yang, Fellow, "S2-RAID: Parallel RAID Architecture for Fast Data Recovery" IEEE Transactions,Vol. 26, No.6, June 2014.*

[3] *Vijay Varadharajan, Udaya Tupakula, "Security as a Service Model for Cloud Environment", IEEE Transactions on network and Service Management, vol. 11, no. 1, March 2014.*

[4] *Marian K. Iskander, Tucker Trainor, Dave W. Wilkinson,Adam J. Lee, "Balancing Performance, Accuracy, and Precision for Secure Cloud Transactions", IEEE Transactions, vol. 25, No. 2, Feb 2014.*

[5] *Aijun Ge, Jiang Zhang, Rui Zhang, Chuangui Ma,and Zhenfeng Zhang, "Security Analysis of a Privacy Preserving Decentralized Key-Policy Attribute-Based Encryption Scheme" IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No.11, November 2013.*

[6] *Daniel Fitch, Haiping Xu ,"A raid-based secure and fault-tolerant model for cloud information storage", University of Massachusetts Dartmouth North Dartmouth, MA 02747,USA, April 2013.*

[7] *Ming Li, Shucheng Yu, Yao Zheng, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption", IEEE Transactions on Parallel and Distributed System, pp. 131-143, 2013.*

[8] *Alexandru Iosup, Simon Ostermann, M. Nezih Yigitbasi, Thomas Fahringer, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing", IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 6, June 2011.*

[9] *Qian Wang, Cong Wang, Kui Ren, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, May 2011.*

[10] *Guangyan Zhang, Weimin Zheng, and Jiwu Shu, "ALV: A New Data Redistribution Approach to RAID-5 Scaling" IEEE Transactions on Computers, vol. 59, No. 3, March 2010.*

[11] *M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems,"Journal of Computer Security, vol. 18, no. 5, pp. 799-837, 2010.*

[12] *J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems,"IEEE Transactions on Parallel and Distributed Systems, vol. 99, 2010.*

[13] *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.*

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

[15] *J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption," in Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEE, 2007, pp.321–334.*

[16] *V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006.*

[17] *http://www.webopedia.com/TERM/R/RAID.html.*