

International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IJCSMC, Vol. 3, Issue. 5, May 2014, pg.851 – 858

RESEARCH ARTICLE

PROCURE DATA CENTRE SHARING SCHEME IN VIRTUAL CLOUD ENVIRONMENT USING CLOUDSIM

Mr. P. Ranjith Kumar^{*}, Ms. M. Shanthi^{}**

Assistant Professor, Department of Computer Science and Engineering*,
ranjith@subramanya.org *

M.E II year, Department of Computer Science and Engineering**,
shanthimecse@gmail.com **

Sri Subramanya College Of Engineering and Technology,
Palani, Dindigul, Tamilnadu, India-624 615

Abstract-Procure Data Centre (PDC) is a coming forth patient data-centric framework of data interchange, large scale data centric applications. In which the data is been outsourced to be stored to general IT providers, such as cloud providers and how to assure their private data while being stored in the cloud servers. To secure the information govern over entree to their own file, it is a hopeful method to encrypt file and personal information before outwards. Yet, effects such as danger of privacy view, measurability in key management, compromising entree and efficient user revocation, have continued the most significant disputes accomplishing fine-grained, cryptographically imposed information entree assure. In this thesis, propose a new data centric role model and a suit of method for information access control to personal profiles put in half-believed servers. To reached close-grained and measurable information entree assure for PDC's, and gained Distributed Multi Authority-Attribute Based Encryption (DMA-ABE) method to generation cipher text of data through for encrypt each data file. Different from past works in assure information outsourcing, focus on the more than one data proprietor security script, and split the users in the PDC scheme into multiple assured area that heavily shrinks the key management complexity for proprietors and consumers. A peak of data privacy is ensured at the same time by working distributed multi-authority ABE. In this scheme also enables dynamic alteration of access policies or file attribute, confirms efficient availability of data that can be needed by users/attribute revocation. Evaluating the performance of Cloud provisioning policies, application workload models, and resources performance models in a repeatable manner under varying system and user configurations and requirements is difficult to achieve. To overcome this challenge, propose CloudSim: an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as Data Centres, Virtual Machines (VMs).The implementation of proposed algorithm is performed by using CloudSim3.0.1 simulator. General Terms: CloudSim Toolkit, Cloudlet, Virtual machine, Data centre

Keywords: Cloud computing, Distributed multi-authority Attribute Based Encryption, key management, protection

I. INTRODUCTION

Recent advances in IT have greatly facilitated remote data storage and sharing. New applications such as online social networks and online documents provide very convenient ways for people to store and share various data including Personal profile, electronic documents and etc. on remote online data servers. Cloud Computing, regarded as the future IT architecture, and even promises to provide unlimited and elastic storage resource as a service to cloud users in a very cost-effective way. Although still at its early stage, Cloud Computing has already drawn great attention, and its benefits have attracted an increasing number of users to outsource their local data centres to remote cloud servers. Data security is a critical issue for remote data storage. In particular, study a novel Distributed Multi Authority – Attribute Based Encryption (ABE), and enhance it toward providing a full-fledged cryptographic basis for a secure data sharing scheme on untrusted storage. Comparing with the preliminary version of this thesis, there are several additional contribution:

1) Clarify and extend the usage of DMA-ABE in the various domain, and formally show how and which types of user-defined file access policies are realized. 2) Clarify the proposed revocable DMA-ABE scheme, and provide a formal security proof it. 3) Carry out both real-world experiments and simulations to evaluate the performance of the proposed solution in this thesis.

Best way to understand the functionality of Cloud Computing is cloud simulation tool. Cloud simulation tool provide the test bed to understand the association of cloud entity and event. Tool provides the sustainable, fault tolerant environment for experimental evaluation of cloud based application like social sites and scientific work flow. Using simulation tool we can find out the finish time taken by the SaaS modeller to run over the virtual machine using resource provisioning algorithm i.e. time shared and space shared at each level.

Integration of PDC with cloud service provides the following benefits:

- 1) *Reduced cost*: Since cloud providers provide the basic infrastructure, platform, software, and storage space, Hospitals no longer need to create their own medical data centre, cutting back on hardware setup costs, as well as Software and hardware upgrade costs.
- 2) *Medical resource sharing and exchange*: Cloud technology allows quick and spontaneous medical resource sharing and exchange from different sources upon users' connection to cloud servers via the Internet.
- 3) *Dynamic scalability of resources*: Cloud services are very flexible in scaling and adjusting to demands, and can support storage expansion demands for medical information systems when required.
- 4) *On-demand self-service*: In cloud computing, computation resource is a shared pool that can provide quick dynamic deployment to hospitals' demands upon purchase.
- 5) *Enhanced flexibility*: Medical documents stored in cloud servers can be accessed by authorized users anytime.
- 6) *Elimination of device limitation*: Irrespective of what computer or mobile services, users can enjoy services as long as they can connect to the Internet.
- 7) *High scalability and service integration*: Through cloud computation, services from different providers can all be integrated to create a single data centre.

II. RELATED WORK

This thesis is mostly related to works in cryptographically enforced access control for outsourced data and DMA-based encryption. To realize fine-grained access control, 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 solutions, one-to-many encryption methods such as ABE can be used. In Goyal *et al.*'s seminal paper on ABE [2], data are encrypted under a set of attributes so that multiple users who possess proper keys can decrypt. This potentially makes encryption and key management more efficient [4]. A fundamental property of ABE is preventing against user collusion. In addition, the encryption or is not required to know the ACL.

A. Achieving secure, scalable, and fine-grained data access control in cloud computing

Cloud computing is a fairly new concept that offers a lot of opportunities for business and companies. As any new system it faces a lot of challenges. One of the most important issues is how to make companies trust cloud providers and how to secure their private data while being stored in the cloud without direct monitoring over it. A trivial and effective solution is to encrypt data while being in the cloud. On the other hand, this solution introduces performance and key management issues. This paper targets the second issue, by introducing a combined system between key policy attribute-based encryption (KP-ABE), Proxy encryption (PRE) and lazy re-encryption.

This system offers secure, scalable, and self-key managed system. The scalability of the systems comes from KP-ABE's properties. The complexity of the system and operations depends on the number of attributes in the system not on the number of users using the system. The system is secured in two ways, first that the data is encrypted in the cloud, but also any communication between any entities requires the use of data signature in order for the receiver to be able to validate the data and its source.

B. Self-Protecting Electronic Medical Records using Attribute-Based Encryption

In additional C.U. Lehmann, M.D. Green, M.W [3] has proposed Self-Protecting Electronic Medical Records Using Attribute-Based Encryption. In general, EMRs offer the potential for greater privacy and better access to records when they are needed. The shift towards EMRs has highlighted the need to develop meaningful techniques for securing records, both inside and outside of the hospital environment. There are emerging XML-based standards for representing EMRs, such as the Continuity of Care Record (CCR) and Continuity of Care Document (CCD). These standards call for protecting EMRs, but they do not provide enough guidance as to how such protection can be achieved. The Standard Specification for Continuity of Care Record states: The CCR document instance must be self-protecting when possible, and the nil point has particular salience in the context of EMR protection. In this thesis, describes aborts to provide ovine, available self-protecting EMRs utilizing recent developments in attribute-based encryption (ABE).

The work is collaboration between security researchers at Johns Hopkins University and medical practitioners at the Johns Hopkins Medical Institution (JHMI). Our approach to access control using ABE facilitates granular role-based and content-based access control for EMRs, without the need for a single, vulnerable centralized server. Providers place a greater emphasis on the availability of medical records in their work than on issues such as security and privacy.

C. Securing the E-Health Cloud

In contrast to PDCs, which are managed by the patients, Electronic Health Records (EHR) are managed by health professionals only. In most countries this involves different legal requirements and a clear distinction between PDCs and EHRs. As a result, infrastructures that involve EHRs are usually more complex than our simple e-health cloud model. The advanced model, which not only involves more parties (e.g., health insurances), but also includes some technical means to enforce data security and privacy of EHRs. The general requirement in this model is still the functional and semantic interoperability of the data stored in EHRs. The EHRs are created, maintained, and managed by health care providers, and can be shared (via the central EHR server in the cloud) with other health professionals. But storing and processing EHRs is not the only service that can be outsourced to the cloud. The health care providers can use billing services that manage their billing and accounting with the health insurances of the patients.

This is a typical scenario that can be found in practice: Many doctors outsource the billing to third party providers. Those billing services accumulate the billing of several patients for different health insurances, but also for various health care providers at the same time. To protect the EHR data, smartcards are typically used to (1) authenticate health professionals and patients, (2) sign EHR documents to provide authenticity, (3) encrypt the EHR data before they are stored in the cloud, and (4) Authorize the access to EHR data. Data and services of the e-health cloud can only be accessed with special interface connections to the telematics infrastructure boundary.

D. Secured Data Consistency and Storage Way in Untrusted Cloud using Server Management Algorithm

It is very challenging part to keep safely all required data that are needed in many applications for user in cloud. Storing our data in cloud may not be fully trustworthy. Since client doesn't have copy of all stored data, he has to depend on Cloud Service Provider. But dynamic data operations, Read-Solomon and verification token construction methods don't tell us about total storage capacity of server allocated space before and after the data addition in cloud. It have efficient storage measurement and space comparison algorithm with time management for measuring the total allocated storage area before and after the data insertion in cloud. So by using our proposed scheme, the value or weight of stored data before and after is measured by client with specified time in cloud storage area with accuracy. If there occurs any server failure, by using this scheme the data can be recovered automatically in cloud server. Here the TPA necessarily doesn't have the delegation to audit user's data.

III. MODELS AND ASSUMPTIONS

A. System Models

Assume that the system is composed of the following parties: the Data Owner, many Data Consumers, many Cloud Servers, and a third Party Auditor if necessary. To access data files shared by the data owner, Data consumers, or users for brevity, download data files of their interest from Cloud Servers and then decrypt. Neither the data owner nor users will be always online. They come online just on the necessity basis. For simplicity, we assume that the only access privilege for users is data file reading. Extending our proposed scheme to support data file writing is trivial by asking the data writer to sign the new data file on each update as does. From now on, it will also call data files by files for brevity. Cloud Servers are always online and operated by the Cloud Service Provider (CSP).

They are assumed to have abundant storage capacity and computation power. The Third Party Auditor is also an online party which is used for auditing every file access event. In addition, also assume that the data owner can not only store data files but also run his own code on cloud Servers to manage his data files.

B. Assurity Models

In this work, just consider Honest but Curious Cloud Servers as does. That is to say, Cloud Servers will follow our proposed protocol in general, but try to find out as much secret information as possible based on their inputs. More specifically, assume Cloud Servers are more interested in file contents and user access privilege information than other secret information. Cloud Servers might collude with a small number of malicious users for the purpose of harvesting file contents when it is highly beneficial. Communication channel between the data owner/users and Cloud Servers are assumed to be secured under existing security protocols such as SSL.

Users would try to access files either within or outside the scope of their access privileges. To achieve this goal, unauthorized users may work independently or cooperatively. In addition, each party is preloaded with a public/private key pair and the public key can be easily obtained by other parties when necessary.

C. Data Confidentiality Models

The owners upload ABE-encrypted PDR files to the server. Each owner's PDR file is encrypted both under a certain fine grained and role-based access policy for users from the PUD to access, and under a selected set of data attributes that allows access from users in the PSD. Only authorized users can decrypt the PDC files, excluding the server.

D. Cloud Server Models

In this models, consider the server to be semi-trusted. That means the server will try to find out as much secret information in the stored PDR files as possible. 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, 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.

Data insert/deletion: This operation can be performed only by the file owner. To insert/delete a file, the owner signs and sends a delete request to the cloud. Upon receiving this request, the cloud checks if the sender is the real owner of the file based on the signature and proceeds to the insert/deletion.

Revocation: There are two types of revocation. The first one consists of limiting access to data through modifying the access policy. To change a data access policy, we should create a new data access structure and re-encrypt the desired data.

The second one consists of revocation of attributes that are associated to user to limit his access scope. This operation induces more computing overhead. To allow an efficient revocation and solve this challenge, we add an expiration time attribute to each users key. This expiration time represents until when the key is considered valid. Another method to deal with the revocation problem is using the proxy re-encryption technique to delegate most of laborious revocation tasks to cloud servers in secure manner.

End-to-end Encryption. In contrast to traditional access control solutions, our approach is designed to secure records from the point of origin (at the hospital or provider), all the way to the recipient. This eliminates the need to rely on an online, trusted server to handle access control decisions and maintain record confidentiality. Individual components within the record may be encrypted depending on security policies developed by the provider. Since records are encrypted, they may be stored in untrusted locations, such as cloud-based systems.

Cloud Record Storage Service (Cloud RS2)

Cloud RS2 is storage for the Internet. It is designed to make web-scale computing easier for developers. Cloud RS2 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. The container for objects stored in Cloud RS2 is called an Cloud RS2 bucket. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Cloud uses to run its own global network of websites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

Image Encryption. In the history of image encryption, various image encryption algorithms can be divided into two categories. One is to convert the image into an one-dimensional data matrix and then encrypt it utilizing an existing data encryption method, such as data encryption standard (DES), advanced encryption standard (AES) or public-key cryptography. The other is to treat the original image as two-dimensional data format and encrypt it applying the discrete Fourier transform (DFT), wave transmission, chaos systems/maps or other image encryption algorithms. The DFT-based image encryption algorithms utilize the phase keys and system parameters to encrypt images. The algorithms have advantages in the multi-parameter selection, high speed, and parallel implementation.

IV. OUR PROPOSED SCHEME

A. Main Idea

The main goal of this framework is to provide protected patient-centric PDC access and efficient key management at the same time. The key idea is to divide the system into multiple security domains (namely, public domains and personal domains) 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 PDCs based on access rights assigned by the owner.

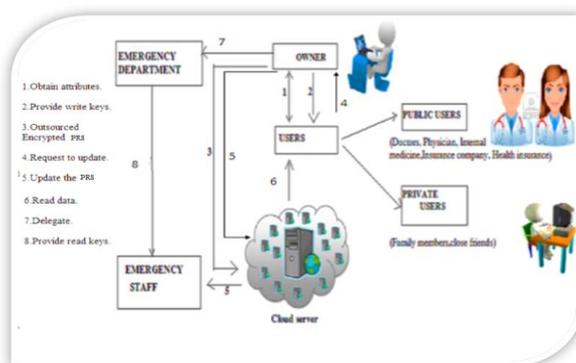


Fig. 1 The proposed framework for protected Patient Centric PDC sharing on semi-trusted storage under multi owner scheme.

In both types of security domains, it utilizes ABE to realize cryptographically enforced, patient-centric PDC access. Especially, Distributed Multi Authority - ABE is used, in which there are multiple "attribute authorities" (AAs), each governing a disjoint subset of attributes. Each data owner (e.g., patient) is a trusted authority of own PSD, to manage the secret keys and access rights of users. Since the users are personally known by the PDC owner, to realize patient centric access, the owner is at the best position to grant user access privileges on a case-by-case basis. When encrypting the data for PSD, all that the owner needs to know is the intrinsic data properties.

For PSD, data attributes are defined which refer to the intrinsic properties of the PDC data, such as the category of a PDC file. For the purpose of PSD access, each PDC file is labelled 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. The multi domain approach best models different user types and access requirements in a PDC system. The data contributors will be granted write access to someone's PDC, if they present proper write keys. The use of DMA-ABE makes the encrypted PDC self-protective, i.e., they can be accessed by only authorized users even when storing on a semi-trusted server, and when the owner is not online.

In addition, efficient and on-demand user revocation is made possible via our DMA-ABE enhancements. Frequently used notation are given in the below table.1

NOTATION	DESCRIPTION
PK,MK	system public key and master key
SK,ASK	Symmetric and asymmetric key
T,L(T)	A user access tree and its leaf node set
T _i	public key component for attribute I
P	Access Policy for a PDC document
M _i	master key component for attribute i
S _{ki}	user secret key for attribute i
I	attribute set assigned to a data file
S _{ki}	user secret key component
DEK	symmetric data encryption key of a data
AttD	the dummy attribute
UL	the system user list
AHL _i	attribute history list for attribute i.

Table.1 frequently used Notation in our scheme description

V. DETAILS OF THE PROPOSED FRAMEWORK

In our framework, there are multiple owners, multiple AAs, and multiple users in addition, DMA-ABE is used. The framework is illustrated in Fig. 1. in this users having read and write access as data readers and contributors, respectively.

A. System Setup in this operation, the data owner chooses a security parameter κ and calls the algorithm level interface $Setup(\kappa)$, which outputs the system public parameter PK and the system master key MK . The data owner then signs each component of PK and sends PK along with these signatures to Cloud Servers.

Cloudsim .Cloud user can deploy the large scale application over the real cloud without taking any responsibility for resource management and resource provisioning. Cloudsim toolkit provides the modelling and simulation of cloud computing system and application provisioning policy implementation. We can model the cloud component using this simulation tool kit. Cloud main resource data centre can be model and configured across the different time zone. Internet applications are accessed by users around the world. This Simulation tool provides the repeatable and controlled environment to setup our own virtual cloud computing environment with different cloud component properties. Using cloudsim toolkit we evaluate the performance of SaaS modeler on the basis of estimated finish time like social networking application. We get the simulation results for Cloudlet running over the cloud environment implemented over the cloudsim at user code level. These results are helpful in quality of service improvement. Finish time of Cloudlet run act as a performance evaluation parameter for cloud task or Cloudlet. Cloudsim Toolkit provides the flexibility to the user to implement his own resource provisioning policy. To construct the virtual cloud computing environment we use the layered architecture of cloudsim and implement the virtual cloud environment the tool.

B.PDC Encryption and Access. The owners upload ABE encrypted PDC files to the server. Each owner’s PDC file is encrypted both under a certain fine-grained and role based access policy for users from the PUD to access, and under a selected set of data attributes that allows access from users in the PSD. Only authorized users can decrypt the PDC files, excluding the server. For improving efficiency, the data attributes will include all the intermediate file types from a leaf node to the root. For example, in Fig. 2, an “allergy” file’s attributes are PDC; medical history; allergy.

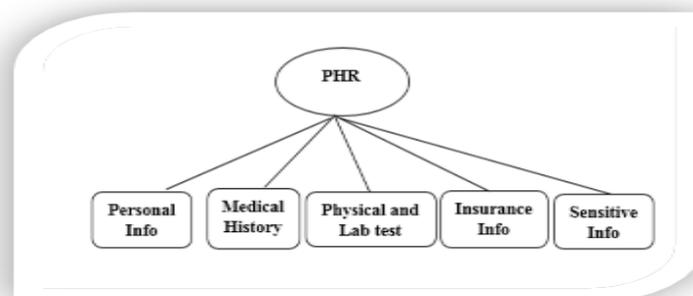


Fig. 2 The attribute Hierarchy of file.

Personal Information: Name,DateOfBirth,Age,Sex,Height,SSN

Medical History: Conditions,Allergies,Medications/Perscriptions.

Examination: Physical Test include Pulse, Heart rate, etc. Lab Test include X-ray images,Blood test

Sensitive Info: HIV/Profile and sensible information of patient.

C. Definitions of Distributed Multi-Authority ABE

We begin by defining a Distributed multi-authority ABE scheme with a trusted setup (but without an online trusted CA), and without any privacy guarantees. For now, we consider a key policy threshold scheme, where the user's decryption key corresponds to a set of attributes and a threshold value.

In a Distributed multi-authority ABE system, we have many attribute authorities, and many users. There are also a set of system wide public parameters available to everyone. A user can choose to go to an attribute authority, prove that it is entitled to some of the attributes handled by that authority, and request the corresponding decryption keys. The authority will run the attribute key generation algorithm, and return the result to the user. Any party can also choose to encrypt a message, in which case uses the public parameters together with an attribute set of his choice to form the ciphertext. Any user who has decryption keys corresponding to an appropriate attribute set can use them for decryption.

Our multi-authority attribute-based access control scheme consists of the following algorithms:

GlobalSetup (λ) \rightarrow (GPK): This algorithm takes in the security parameter λ , it then outputs the global parameters GPK for the system.

CASetup (GPK) \rightarrow (CPK, CMK): The CA runs this algorithm with GPK as input to produce its public parameter CPK and the corresponding master secret key CMK . CPK will be used by AAs only.

AASetup (GPK, f, U_f) \rightarrow (APK_f, AMK_f): Each AA_f runs this algorithm with GPK and its attribute domain U_f as input to produce the public parameter APK_f and the corresponding master secret key AMK_f . For $i \neq j$, we have $U_i \cap U_j = \emptyset$.

Encrypt ($M, A, GPK, UAPK_f$) \rightarrow (CT): This algorithm takes in GPK , a message M , an access structure A and the set of public parameters for relevant AAs. It produces a ciphertext CT .

We assume the access structure A is implicitly included in CT .

CAKeyGen (GPK, gid) \rightarrow ($DSK_{gid}, CASK_{gid}, CAPK_{gid}$): This algorithm takes in GPK and the user's gid . It then outputs a decryption key DSK_{gid} , a gid -related private key $CASK_{gid}$ and a gid -related public key $CAPK_{gid}$, where DSK_{gid} will be used by the user, $CASK_{gid}$ will be used in pre-decrypt the ciphertext and $CAPK_{gid}$ will be used to generate the attribute-related keys by the AAs.

AAKeyGen ($S_{gid}, f, GPK, CPK, CAPK_{gid}, AMK_f$) \rightarrow ($ASKS_{gid}, f$): When a user submits a set of attributes S_{gid}, f belongs to AA_f to request the attribute-related key ASK_{gid}, f , AA_f runs this algorithm with $S_{gid}, f, GPK, CPK, CAPK_{gid}$ and AMK_f as input. If $CAPK_{gid}$ is invalid, it outputs L . Otherwise, it outputs $ASKS_{gid}, f = \{ASK_{ATT}, gid | ATT \in S_{gid}, f\}$. We let $ASKS_{gid} = \cup UASKS_{gid}, f$ denotes the attribute-related key of S_{gid} , where $S_{gid} = \cup S_{gid}, f$. We assume the set S_{gid} is implicitly included in $ASKS_{gid}$.

Pre-Decrypt ($CT, GPK, CASK_{gid}, ASKS_{gid}$) \rightarrow ($PDKEY$): This algorithm takes in $CT, GPK, CASK_{gid}$ and $ASKS_{gid}$. It outputs the pre-decryption key $PDKEY$ of CT if and only if S_{gid} satisfies A .

Decrypt ($CT, PDKEY, DSK_{gid}$) \rightarrow (M): This algorithm takes in $CT, PDKEY$ and DSK_{gid} . It outputs the plaintext message M .

Key-updating ($gid, RLATT, ASK_{ATT}, gid$) \rightarrow (ASK'_{ATT}, gid): This algorithm takes in a gid , a revocation list of an attribute $RLATT$ and the original key ASK_{ATT}, gid . If the $gid \notin RLATT$, it outputs a new ASK'_{ATT}, gid .

Re-Encrypt ($CT, ATTRC$) \rightarrow CT' : If a revocation operation on attribute $ATTRC$ occurs, this algorithm takes in the original ciphertext CT and $ATTRC$, it outputs a new ciphertext CT' which can only be decrypt by those who have appropriate attributes and are not in RL .

D. Policy updates. A PDC owner can update her sharing policy for an existing PDC document by updating the attributes (or access policy) in the cipher text. The supported operations include add/delete/modify, which can be done by the server on behalf of the user.

VI. PERFORMANCE MEASURES

The results are given in Table 2. The cipher text size only accounts for the encryption of FEK. In our scheme, for simplicity assume there is only one PUD, thus the cipher text includes m additional wildcard attributes and up to $N - 1$ dummy attributes. In this scheme requires a secret key size that is linear, the number of attributes of each user, while in the E-Health and HIPA schemes this is linear with since a user needs to obtain at least one key from each owner whose PDC file the user wants to access.

SCHEME	SECURITY	USERDOMAIN	ACCESS POLICY
E-Health [2]	Not against user-server collusion	All	ACL level
HIPA [3]	No collusion risk	PUD	ACL level
EMD [1]	Single TA	PUD	Attribute and ID-based policy
Our scheme	Against N-2 AA collusion	All (PSD &PUD)	Conjunctive form with wildcard

Table.2 Comparison of Security

A. Efficient Way For Data Storage Measurement And Space Comparison Algorithm

Suppose that $D \in S$ and, before data insertion, initial values are, (a) $D \notin S$, where $D =$ data inserted, S is the cloud server space allocated by CSP. Now values are as the following equations, (b) $\sum_{i=0}^n [(S_i - D_i)]S - 1$, where $(i \geq 0)$, $S =$ total number of multi-server and $n =$ number of count in cloud data insertion. Now after insertion, it becomes $D \in S$, Here S „server“ has the data „ D “ then, (c) $D = \{b_i, b_{i+1}, b_{i+2}, \dots, b_{i+n}\}$, where $b =$ bits of data or bytes of data or any amount of data that we add it in cloud server.

B. Data Value Measurement in Cloud Database

It is followed in a few steps when we attempt to calculate the data from its storage area excluding from CSP’s allocated space. All these data insertions are checked for allocated space to test the remaining space that we want to know and to maintain integrity of data apart from some other methods such as file distribution and error correctness where existing systems have from its system design. These data calculations are clearly explained in our next part. Figure 3 explains the upload of a file to one specified web site using some software tools such as file uploader. At present there is more such type of tools available.

C. Comparison of New Data or Data File Values with Allocated Space

Old value of the data is compared with a newly inserted data value. Hereafter, the following conditions must be satisfied with each other for the data integrity, and then in our entire data comparison, we have the following equations as, (a) $(D \notin S) \neq (D \in S)$ then, it is confirmed that multi-server has some data in its cloud server. Also, (b) $\sum_{i=0}^n [(S_i - D_i)]S - 1 + \sum_{i=0}^n [(D_i)]S - 1$ and the above equation is only a single time equation for overall data with its summation when client is trying for data insertion in the cloud server.

VII.SIMULATION

Evaluation of alternative designs or solutions for Cloud computing on real test-beds is not easy due to several reasons. Firstly, public Clouds exhibit varying demands, supply patterns, system sizes, and resources (hardware, software, network). Due to such unstable nature of Cloud resources, it is difficult to repeat the experiments and compare different solutions. Secondly, there are several factors which are involved in determining performance of Cloud systems or applications such as user’s Quality of Service (QoS) requirements, varying workload, and complex interaction of several network and computing elements. Thirdly, the real experiments on such large-scale distributed platforms are considerably time consuming and sometimes impossible due to multiple test runs in different conditions.

The experiments in this research were performed on the CloudSim cloud simulator which is a framework for modeling and simulating the cloud computing infrastructures and services. The CloudSim simulator has many advantages: it can simulate many cloud entities, such as datacentre, host and broker. It can also offer a repeatable and controllable environment. And we do not need to take too much attention about the hardware details and can concentrate on the algorithm design. The simulated datacentre and its components can be built by coding and the simulator is very convenient in algorithm design. The functions of those components are explained in table 2.

CLOUDSIM COMPONENT	FUNCTION
Cloud Information Service	It is an entity that registers, indexes and discovers the resource.
Datacenter	It models the core hardware infrastructure, which is offered by Cloud providers.
Datacenter Broker	It models a broker, which is responsible for mediating negotiations between SaaS and Cloud providers.
Host	It models a physical server.
Vm	It models a virtual machine which is run on Cloud host to deal with the cloudlet.
Cloudlet	It models the Cloud-based application services.
VmAllocation	A provisioning policy which is run in datacenter level helps to allocate VMs to hosts.
VmScheduler	The policies required for allocating process cores to VMs. It is run on every Host in Datacenter.
CloudletScheduler	It determines how to share the processing power among Cloudlets on a virtual machine. It is run on VMs.

Table.3 Cloudsim components and their functions

VIII. CONCLUSION

In this thesis, have utilize DMA-ABE to encrypt the PDC data, so that patients can allow access not only by personal users, but also various users from public domains with different professional roles, qualifications, and affiliations. By this method records are share in secure way, also manage the key escrow problem, on-demand efficient user/attribute revocation, multiple authority can be used for PDC owners and users, fully protect the data from the unauthorized users. To study and analyze the cloud base application performance under the cloud computing environment, we setup our own virtual cloud computing environment and perform test to identify the performance of Cloudlet. Entire Application like facebook cannot be deployed without using real cloud. Task model is used to model application by Cloudlet. We get the value of execution time taken by the cloudlet to run over the virtual machine in a timeshared mode. Quality of service evaluation parameter value is strongly affected by the available storage, compute and network resources at infrastructure level.

The simulation results for Cloudlet which holds the user request, user program running over the virtual cloud with optimal cloud configuration. Using our own resource provisioning policy at virtual machine level, we can improve the quality of service. Simulation results help us to fine tune the performance while deploying the application over the real cloud.

REFERENCES

- [1]Ming Li, Member,ShuchengYu,WenjingLou“Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption “, 2013.
- [2] J.A. Akinyele, C.U. Lehmann, M.D. Green, M.W. Pagano, Z.N.J. Peterson, and A.D. Rubin, (2010) “Self-Protecting Electronic Medical Records Using Attribute-Based Encryption,” Cryptology ePrint Archive, Report 2010/565, <http://eprint.iacr.org/>.
- [3] Lohr H, Sadeghi AR, Winandy M. (2010) “Securing the e-health cloud. Proceedings of the 1st ACM International Health Informatics Symposium, ser. IHI '10, 2010; 220–229.
- [4] Yu, C. Wang, K. Ren, and W. Lou, (2010)“Achieving Secure, Scalable, and Fine-GrainedData Access Control in Cloud Computing,” Proc. IEEE INFOCOM.
- [5] Yuxiang Shi; Xiaohong Jiang and Kejiang Ye, "An Energy-Efficient Scheme for Cloud Resource Provisioning Based on CloudSim," Cluster Computing (CLUSTER), 2011 IEEE International Conference on, 26-30 Sept. 2011, pp.: 595 - 599.