**DECENTRALIZED ACCESS CONTROL WITH ANONYMOUS AUTHENTICATION OF DATA STORED IN CLOUDS**

**A PROJECT REPORT**

Submitted by

SAI KRISHNA MUDHIGANDLA

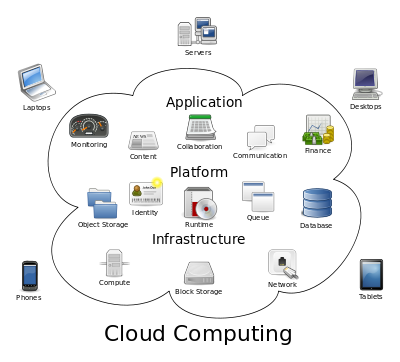
**ABSTRACT**

We propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. In the proposed scheme, the cloud verifies the authenticity of the series without knowing the user’s identity before storing data. Our scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud. We also address user revocation. Moreover, our authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized. The communication, computation, and storage overheads are comparable to centralized approaches.

1. **INTRODUCTION**

**1.1 Cloud Computing**

Cloud computing is the use of [computing](http://en.wikipedia.org/wiki/Computing) resources (hardware and software) that are delivered as a service over a network (typically the [Internet](http://en.wikipedia.org/wiki/Internet)). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.



Structure of cloud computing

**1.2 Working of Cloud Computing**

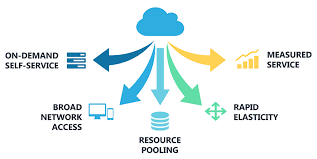
The goal of cloud computing is to apply traditional [supercomputing](http://www.webopedia.com/TERM/S/supercomputer.html), or [high-performance computing](http://www.webopedia.com/TERM/H/High_Performance_Computing.html) power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses [networks](http://www.webopedia.com/TERM/N/network.html) of large groups of [servers](http://www.webopedia.com/TERM/S/server.html) typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared [IT](http://www.webopedia.com/TERM/I/IT.html) infrastructure contains large pools of systems that are linked together. Often, [virtualization](http://www.webopedia.com/TERM/V/virtualization.html) techniques are used to maximize the power of cloud computing.

**1.3 Characteristics Of Cloud Computing**

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

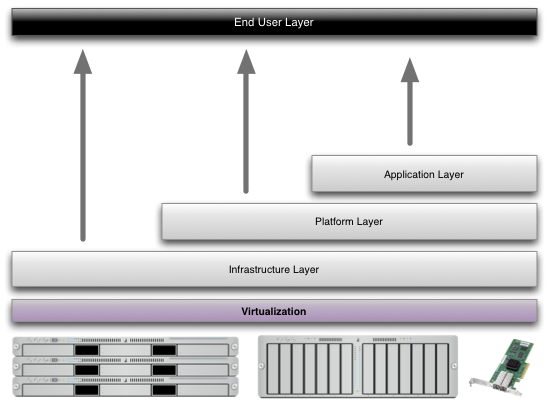
* **On-demand self-service**: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service’s provider.
* **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
* **Resource pooling**: The provider’s computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
* **Rapid elasticity**: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
* **Measured service**: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.



**Characteristics of cloud computing**

**1.4** **Service Models:**

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.



Structure of service models

**1.5 Cloud Computing Types**

**1.5.1 Private Cloud**

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party, and hosted either internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and requires the organization to reevaluate decisions about existing resources. When done right, it can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive.

**1.5.2 Public Cloud**

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free. Technically there may be little or

no difference between public and private cloud architecture, security consideration may be substantially different for services that are made available by a service provider for a public audience and when communication is effected over a non-trusted network.

**1.5.3 Hybrid Cloud**

Hybrid cloud is a composition of two or more clouds that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources. A hybrid cloud service crosses isolation and provider boundaries so that it can't be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration or customization with another cloud service.

**1.6 Security in Cloud**

According to the Cloud Security Alliance, the top three threats in the cloud are "Insecure Interfaces and API's", "Data Loss & Leakage", and "Hardware Failure". In a cloud provider platform being shared by different users there may be a possibility that information belonging to different customers resides on same data server. Information leakage may arise by mistake when information for one customer is given to other. Eugene Schultz, chief technology officer at Imagined Security, said that hackers are spending substantial time and effort looking for ways to as have this data be indexed by search engines (making the information public) penetrate the cloud. The data from hundreds or thousands of companies can be stored on large cloud servers, hackers can theoretically gain control of huge stores of information through a single attack. A process called "hyper jacking". Some examples of this include the Dropbox security breach, and Cloud 2014 leak. Dropbox had been breached in October 2014, having over 7 million of its user’s passwords stolen by hackers in an effort to get monetary value from it by Bit coins (BTC). By having these passwords, they are able to read private data as well.

There is the problem of legal ownership of the data. Physical control of the computer equipment (private cloud) is more secure than having the equipment off site and under someone else's control (public cloud). This delivers great incentive to public cloud computing service providers to prioritize building and maintaining strong management of

secure services. Some small businesses that don't have expertise in IT security could find that it's more secure for them to use a public cloud. There is the risk that end users don't understand the issues involved when signing on to a cloud service (persons sometimes don't read the many pages of the terms of service agreement, and just click "Accept" without reading). This is important now that cloud computing is becoming popular and required for some services to work, for example for an intelligent personal assistant.

Fundamentally private cloud is seen as more secure with higher levels of control for the owner, however public cloud is seen to be more flexible and requires less time and money investment from the user. Efficient search on encrypted data is also an important concern in clouds. The clouds should not know the query but should be able to return the records that satisfy the query. This is achieved by means of searchable encryption. The keywords are sent to the cloud encrypted, and the cloud returns the result without knowing the actual keyword for the search. The problem here is, that the data records should have keywords associated with them to enable the search. The correct records are returned only when searched with the exact keywords.

Security and privacy protection in clouds are being explored by many researchers. Many holomorphic encryption techniques have been suggested to ensure that the cloud is not able to read the data while performing computations on them. Using holomorphic encryption, the cloud receives cipher text of the data and performs computations on the cipher text and returns the encoded value of the result. The user is able to decode the result, but the cloud does not know what data it has operated on. In such circumstances, it must be possible for the user to verify that the cloud returns correct results.

**1.7 Access Control in Cloud**

Access control in clouds is gaining attention because it is important that only authorized users have access to valid service. A huge amount of information is being stored in the cloud, and much of this is sensitive information. Care should be taken to ensure access control of this sensitive information which can often be related to health, important documents (as in Google Docs or Dropbox) or even personal information (as in social networking).

There are broadly three types of access control:

* User Based Access Control (UBAC),
* Role Based Access Control (RBAC),
* Attribute Based Access Control (ABAC).

In UBAC, the access control list (ACL) contains the list of users who are authorized to access data. This is not feasible in clouds where there are many users. In RBAC, users are classified based on their individual roles. Data can be accessed by users who have matching roles. The roles are defined by the system. For example, only faculty members and senior secretaries might have access to data but not the junior secretaries. ABAC is more extended in scope, in which users are given attributes, and the data has attached access policy. Only users with valid set of attributes, satisfying the access policy, can access the data. For instance, in the above example certain records might be accessible by faculty members with more than 10 years of research experience or by senior secretaries with more than 8 years’ experience. There has been some work on ABAC in clouds all these work use a cryptographic primitive known as Attribute Based Encryption (ABE). The extensible Access Control Markup Language (XACML) has been proposed for ABAC in clouds. An area where access control is widely being used is healthcare. Clouds are being used to store sensitive information about patients to enable access to medical professionals, hospital staff, researchers, and policy makers. It is important to control the access of data so that only authorized users can access the data.

Using ABE, the records are encrypted under some access policy and stored in the cloud. Users are given sets of attributes and corresponding keys. Only when the users have matching set of attributes, can they decrypt the information stored in the cloud. Access control is also gaining importance in online social networking where users (members) store their personal information, pictures, videos and share them with selected groups of users or communities they belong to. A colossal measure of data is constantly archived in the cloud, and much of this is sensitive data. Utilizing Attribute Based Encryption (ABE), the records are encrypted under a few access strategy furthermore saved in the cloud. Clients are given sets of traits and corresponding keys. Just when the clients have matching set of attributes, would they be able to decrypt the data saved in the cloud. Studied the access control in health care. Access control is likewise gaining imperativeness in online social networking where users store their personal data, pictures, films and shares them with selected group of users they belong. The work done by gives privacy preserving authenticated access control in cloud. Nonetheless, the researchers take a centralized methodology where a single key

distribution center (KDC) disperses secret keys and attributes to all clients. Unfortunately, a single KDC is not just a single point of failure however troublesome to uphold due to the vast number of clients that are upheld in a nature's domain. Multi-authority ABE principle was concentrated on in, which obliged no trusted power which requires each client to have characteristics from at all the KDCs. Access control in online social networking has been studied in such data are being stored in clouds. It is very important that only the authorized users are given access to those information. A similar situation arises when data is stored in clouds, for example in Dropbox, and shared with certain groups of people. It is just not enough to store the contents securely in the cloud but it might also be necessary to ensure anonymity of the user. For example, a user would like to store some sensitive information but does not want to be recognized. The user might want to post a comment on an article, but does not want his/her identity to be disclosed. The user should be able to prove to the other users that he/she is a valid user who stored the information without revealing the identity. There are cryptographic protocols like ring signatures, mesh signatures, group signatures, which can be used in these situations. Ring signature is not a feasible option for clouds where there are a large number of users. Group signatures assume the pre-existence of a group which might not be possible in clouds. Mesh signatures do not ensure if the message is from a single user or many users colluding together. For these reasons, a new protocol known as Attribute Based Signature (ABS) has been applied.

In ABS, users have a claim predicate associated with a message. The claim predicate helps to identify the user as an authorized one, without revealing its identity. Other users or the cloud can verify the user and the validity of the message stored. ABS can be combined with ABE to achieve authenticated access control without disclosing the identity of the user to the cloud. A key-policy (KP) ABE scheme that allows for threshold policies. Key-policy means that the encryption only gets to label a cipher text with a set of attributes. The authority chooses a policy for each user that determines which cipher texts he can decrypt. In a multi-authority ABE system, we have many attribute authorities, and many users. A user can choose to go to an attribute authority, prove that it is entitled to some of the attribute handled by that authority, and request the corresponding decryption keys. Any party can also choose to encrypt a message, in which case he uses the public parameters together with an attribute set of his choice to form the cipher text. Any user who has decryption keys corresponding to an appropriate attribute set can use them for decryption.

**2. SYSTEM ANALYSIS**

**2.1 Existing Systems:**

Existing work on access control in cloud are centralized in nature. Except and, all other schemes use ABE. The scheme in uses a symmetric key approach and does not support authentication. The schemes do not support authentication as well.

It provides privacy preserving authenticated access control in cloud. However, the authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users.

**2.1.1 Disadvantages Of Existing Systems:**

The scheme in uses asymmetric key approach and does not support authentication.

Difficult to maintain because of the large number of users that are supported in a cloud environment.

**2.2 Proposed Systems:**

We propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication.

In the proposed scheme, the cloud verifies the authenticity of the series without knowing the user’s identity before storing data.

Our scheme also has the added feature of access control in which only valid users are able to decrypt the stored information.

The scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud.

**2.2.1 Advantages Of Proposed Systems:**

Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.

Authentication of users who store and modify their data on the cloud.

The identity of the user is protected from the cloud during authentication.

**3. IMPLEMENTATION**

**3.1 Modules:**

* Encryption / Decryption
* File Upload / Download
* Policy Revocation for File Assured Deletion
* File Access Control
* Policy Renewal

**3.2 Modules Decryption:**

**3.2.1 Encryption / Decryption**

We used RSA algorithm for encryption/Decryption. This algorithm is the proven mechanism for secure transaction. Here we are using the RSA algorithm with key size of 2048 bits. The keys are split up and stored in four different places. If a user wants to access the file he/she may need to provide the four set of data to produce the single private key to manage encryption/decryption.

**3.2.2 File Upload**

The client made request to the key manager for the public key, which will be generated according to the policy associated with the file. Different policies for files, public key also differs. But for same public key for same policy will be generated. Then the client generates a private key by combining the username, password and security credentials. Then the file is encrypted with the public key and private key and forwarded to the cloud.

**3.2.3 File Download**

The client can download the file after completion of the authentication process. As the public key maintained by the key manager, the client request the key manager for public key. The authenticated client can get the public key. Then the client can decrypt the file with the public key and the private key. The users credentials were stored in the client itself. During download the file the cloud will authenticate the user whether the user is valid to download the file. But the cloud doesn’t have any attributes or the details of the user.

**3.2.4 Policy Revocation for File Assured Deletion**

The policy of a file may be revoked under the request by the client, when expiring the time period of the contract or completely move the files from one cloud to another cloud environment. When any of the above criteria exists the policy will be revoked and the key manager will completely removes the public key of the associated file. So no one recover the control key of a revoked file in future. For this reason we can say the file is assuredly deleted. Automatic file revocation scheme is also introduced to revoke the file from the cloud when the file reaches the expiry and the client didn’t renew the files duration.

**3.2.5 File Access Control**

Ability to limit and control the access to host systems and applications via communication links. To achieve, access must be identified or authenticated. After achieved the authentication process the users must associate with correct policies with the files. To recover the file, the client must request the key manager to generate the public key. For that the client must be authenticated. The attribute based encryption standard is used for file access which is authenticated via an attribute associated with the file. With file access control the file downloaded from the cloud will be in the format of read only or write supported. Each user has associated with policies for each file. So the right user will access the right file. For making file access the attribute based encryption scheme is utilized.

**3.2.6 Policy Renewal**

Policy renewal is a tedious process to handle the renewal of the policy of a file stored on the cloud. Here we implement one additional key called as renew key, which is used to renew the policy of the file stored on the cloud. The renew key is stored in the client itself.

**4. DESIGN**

**4.1 Input Design:**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**4.1.1 Objectives:**

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user

will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

**4.2 Ouput Design:**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

Select methods for presenting information.

Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**DATA FLOW DIAGRAMS**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



**5.1 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**Goals:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**5.1.1 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



**5.1.2 CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

