12-2016

An Application for Decentralized Access Control Mechanism on Cloud Data using Anonymous Authentication

Savanth Chintoju
schintoju@stcloudstate.edu

Follow this and additional works at: https://repository.stcloudstate.edu/msia_etds

Recommended Citation
https://repository.stcloudstate.edu/msia_etds/13

This Starred Paper is brought to you for free and open access by the Department of Information Systems at theRepository at St. Cloud State. It has been accepted for inclusion in Culminating Projects in Information Assurance by an authorized administrator of theRepository at St. Cloud State. For more information, please contact rsweelbaum@stcloudstate.edu.
An Application for Decentralized Access Control Mechanism on 
Cloud Data using Anonymous Authentication

by

Savanth Chintoju

A Starred Paper
Submitted to the Graduate Faculty of
St. Cloud State University
in Fulfillment of the Requirements
for the Degree of
Master of Science
in Information Assurance

October 2016

Starred Paper Committee:
Dennis Guster, Chairperson
Susantha Herath
Sneh Kalia
Abstract

In the last few years, Cloud computing has gained a lot of popularity and technology analysts believe it will be the future, but only if the security problems are solved from time-to-time. For those who are unfamiliar with cloud computing, it is a practice wherein users can access the data from the servers that are located in remote places. Users can do so through the Internet to manage, process and store the relevant data, instead of depending on the personal computer or a local server. Many firms and organizations are using cloud computing, which eventually is faster, cheaper and easy to maintain. Even the regular Internet users are also relying on cloud computing services to access their files whenever and wherever they wish. There are also numerous challenges associated with cloud computing like abuse of cloud services, data security and cyber-attacks. When clients outsource sensitive data through cloud servers, access control is one of the fundamental requirements among all security requirements which ensures that no unauthorized access to secured data will be avoided. Hence, cloud computing has to build a feature that provides privacy, access control challenges and security to the user data. A suitable and reliable encryption technique with enhanced key management should be developed and applied to the user data before loading into the cloud with the goal to achieve secured storage. It also has to support file access control and all other files related functions in a policy based manner for any file stored in a cloud environment. This research paper proposes a decentralized access control mechanism for the data storage security in clouds which also provides anonymous authentication. This mechanism allows the decryption of the stored information only by the valid users, which is an additional feature of access control. Access control mechanism are decentralized which makes it robust when compared to centralized access control schemes meant for clouds.
### Table of Contents

LIST OF TABLES .................................................................................................................. 6

LIST OF FIGURES ................................................................................................................ 7

I. INTRODUCTION ................................................................................................................ 9

   Introduction ......................................................................................................................... 9
   Problem Statement ............................................................................................................. 15
   Objective .............................................................................................................................. 16
   Significance of the Study .................................................................................................... 16
   Scope of the Study .............................................................................................................. 16
   Definition of Terms ............................................................................................................ 18

II. BACKGROUND AND LITERATURE REVIEW .................................................................... 19

   Cloud Computing ................................................................................................................ 20
   Advantages of Cloud ......................................................................................................... 20
   Access Control in Clouds ................................................................................................... 21
   Cryptography ...................................................................................................................... 22
   RSA Algorithm ................................................................................................................... 24
   Data Encryption Standard (DES) ........................................................................................ 26
   Advanced Encryption Standard (AES) ................................................................................ 28
   Encryption in Clouds .......................................................................................................... 30

III. SYSTEM ANALYSIS ......................................................................................................... 34

   Existing System .................................................................................................................. 34
   Proposed System ................................................................................................................. 34
   System requirements ......................................................................................................... 36
### IV. SOFTWARE ENVIRONMENT

<table>
<thead>
<tr>
<th>Software</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAVA</td>
<td>37</td>
</tr>
<tr>
<td>MySQL</td>
<td>39</td>
</tr>
<tr>
<td>JDBC</td>
<td>40</td>
</tr>
</tbody>
</table>

### V. IMPLEMENTATION

<table>
<thead>
<tr>
<th>Component</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Owner</td>
<td>Error! Bookmark not defined.</td>
</tr>
<tr>
<td>Data Consumer</td>
<td>Error! Bookmark not defined.</td>
</tr>
<tr>
<td>Cloud Server</td>
<td>Error! Bookmark not defined.</td>
</tr>
<tr>
<td>The Public Cloud</td>
<td>44</td>
</tr>
<tr>
<td>The Private Cloud</td>
<td>44</td>
</tr>
<tr>
<td>The Hybrid or Mixed Cloud</td>
<td>44</td>
</tr>
<tr>
<td>The Community Cloud</td>
<td>44</td>
</tr>
<tr>
<td>Key Distribution Centre</td>
<td>44</td>
</tr>
<tr>
<td>Design Goals</td>
<td>47</td>
</tr>
<tr>
<td>Application Screenshots</td>
<td>48</td>
</tr>
</tbody>
</table>

### VI. SYSTEM DESIGN

<table>
<thead>
<tr>
<th>Component</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Flow</td>
<td>Error! Bookmark not defined.</td>
</tr>
<tr>
<td>Data Flow Diagrams</td>
<td>Error! Bookmark not defined.</td>
</tr>
<tr>
<td>Class Diagram</td>
<td>Error! Bookmark not defined.</td>
</tr>
<tr>
<td>Use Case and Sequence Diagrams</td>
<td>64</td>
</tr>
</tbody>
</table>

### VII. SYSTEM TESTING

<table>
<thead>
<tr>
<th>Component</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of Testing</td>
<td>Error! Bookmark not defined.</td>
</tr>
<tr>
<td>Test Objectives</td>
<td>Error! Bookmark not defined.</td>
</tr>
</tbody>
</table>
Features Tested ................................................................. Error! Bookmark not defined.

VIII. CONCLUSION .................................................................................................................. 73

Introduction ......................................................................................................................... Error! Bookmark not defined.

Results & Conclusion ......................................................................................................... Error! Bookmark not defined.

Future Work ......................................................................................................................... Error! Bookmark not defined.

References ............................................................................................................................. Error! Bookmark not defined.
List of Table

Table 1 Definition of Terms Used in this Document

18
List of Figures

Figure 1: Overall System Design .................................................................................. 17
Figure 2: Cryptography Overview ............................................................................... 23
Figure 3: Public Key Cryptography ............................................................................. 25
Figure 4: DES Algorithm ......................................................................................... 27
Figure 5: AES Algorithm ......................................................................................... 29
Figure 6: Overall System Flow .................................................................................. 35
Figure 7: KDC Architecture ..................................................................................... 48
Figure 8: Application Home Page .............................................................................. 49
Figure 9: Owner Registration Page ............................................................................ 49
Figure 10: Owner Successful Registration Pop Up ................................................... 49
Figure 11: Owner Login Page .................................................................................... 50
Figure 12: Owner Details Page .................................................................................. 50
Figure 13: File Upload Request ................................................................................. 51
Figure 14: Request Success Message ......................................................................... 52
Figure 15: KDC Login Page ...................................................................................... 52
Figure 16: KDC Database Page ............................................................................... 53
Figure 17: File Upload Page of Owner ...................................................................... 53
Figure 18: Uploaded File Successful Encryption Message ..................................... 54
Figure 19: Owner Data Base Page ............................................................................ 54
Figure 20: Re Encryption of Data Page ..................................................................... 55
Figure 21: Re Encryption Success Page .................................................................... 55
Figure 22: Data Consumer Registration Page .......................................................... 56
Figure 23: Data Consumer Login Page ................................................................. 56
Figure 24: Data consumer Details Page .................................................................. 57
Figure 25: File View Page for Consumer ............................................................... 57
Figure 26: Access Request Pop Up ........................................................................ 58
Figure 27: File Database View .............................................................................. 58
Figure 28: Access Key Verification Page ............................................................... 59
Figure 29: Verification Success Pop Up ................................................................. 59
Figure 30: File Download Page ............................................................................ 59
Figure 31: Cloud Admin Access Page .................................................................. 60
Figure 32: Statistics Page ..................................................................................... 63
Figure 33: Class diagram ..................................................................................... 64
Figure 34: Use Case and Sequence Diagrams ...................................................... 68
Chapter I: INTRODUCTION

Introduction

Cloud Computing and storage has achieved the greatest quality as of now in the technical arena. In modern business settings, the rise in demand for data outsourcing can be witnessed, which marks to be the key organization of corporate information. It is also utilized as a key development component behind various online affiliations for individual applications. Nowadays, having a limited quantity of data equivalent to 25 GB (1 TB at an additional cost), it is obviously not hard to apply with the desire of complimentary records for electronic mail, multimedia storage, archive sharing and/or remote access. Going hand-in-hand with the current day remote advancement, data users can get to the bigger part of their records and messages with a cell phone in any side of the universe [Lewko, A., & Waters, B. (2011)].

Considering the data security, an ordinary way to deal with objective without question is to rely upon the server to approve the passageway control after affirmation, which suggests any unanticipated advantage increasing speed will reveal all data. In a shared residency disseminated figuring environment, things end up being significantly more serious. Information from different clients can be stored and accessed on different virtual machines (VMs) may or may not be mounted on a singular physical machine. Data in a target VM could be stolen by instantiating another VM on the same machine as the target one.

A secured server, in addition to providing an ensured platform that facilitates the Web applications and web server design, also has to answer the web application's security. A server can prompt unapproved access. Ignored client records can allow an attacker to hack your data without notice. Seeing the dangers to your web server and having the capacity to distinguish proper
countermeasures enables you to suspect numerous security hazards and upset the regularly developing quantities of intruders.

The proposed application gives a bidirectional encryption of correspondences between a client and server, which ensures against listening stealthily and messing with and/or manufacturing the substance of the correspondence. The application that the paper is prepared to show up is to speak with and also assure that the substance of agreements between the client and the server can't be accessed or manufactured by any attacker. Any secure server plus application has to have a primary two-way login security.

Explaining in brief, in the wake of signing into the application client gets a hidden key on his specified email id. This private key must be mentioned in the text box provided in the process of signing into secured server application. This application has two functionalities, Encryption and Decryption. Encoding is the usefulness in which the document to be institutionalized over the mail in firstly converted to byte configuration and alter encoded utilizing distinctive encryption calculations. After Encryption records would be saved into the cloud database. At the consumer end, he will download the documents using the application after being authorized necessarily by the KDC.

Client security is also a mandatory in cloud. In any secured cloud application, the cloud or different clients don't have even the foggiest idea about the individuality or identity of the other clients. The vast storage can hold the client data and their requests in the cloud, and in the similar manner, to provide benefits, the cloud itself is responsible. The genuineness of the client who stores the information is also confirmed. This also specifies the requirement of law authorization for assuring data security and protection. Numerous encryption systems have been practiced to put away information on cloud to peruse the data while doing calculations on the information. By
utilizing Attribute based encryption (ABE), the cloud gets the content of the information, performs calculations and passes the encoded format of the final result to the client. The client can then interpret the outcome, despite the fact that the cloud does not comprehend what information it has worked.

Different methods have been proposed to ensure the information substance protection by means of user control. Identity based encryption (IBE) was initially presented by Shamir, in which the sender of a message can indicate a character such that just a beneficiary with coordinating identity can unscramble it. A twosome of years after the fact, Fuzzy Identity-Based Encryption is proposed, which is otherwise called Attribute-Based Encryption (ABE). In such encryption methodology, an identity is viewed as an arrangement of clear characteristics, and decoding is conceivable if a decrypted character has a few covers with the one defined in the cipher text. Before long, more broad tree-based ABE plans, Key-Policy Attribute-Based Encryption (KP-ABE) and Cipher text-Policy Attribute-Based Encryption (CP-ABE), are introduced to express more detailed condition than clear cut ‘cover’. They are invariants to each other as in the choice of encryption strategy (who can or can't decode the message) is set by various gatherings [Waters, B. (2011)].

In KP-ABE, a cipher text is associated with a course of action of qualities, and a private key is associated with a monotonic access structure like a tree, which describes this present customer's identity (e.g. Bachelors AND (Ph.D. OR Master)). A customer can unscramble the cipher text if and only if the passageway tree in his private key is satisfied with the references in the cipher text. In any case, the encoding scheme is depicted in the keys, so the encrypting user does not possess total control over the encoding approach. It needs to believe that the key generators issue keys with the right structures to the right clients. Additionally, when a re-encryption happens, most of the nodes in the same organization must hold their individual keys,
re-issued keeping in mind the end goal to blend to the re-encoded discs, and this technique causes huge issues in the execution.

Of course, those issues, designing and operating expenses are all rated in the CP-ABE. In the CP-ABE, cipher texts are made with a passage structure, which indicates the encryption methodology, and private keys are created by qualities. A customer can disentangle the cipher text if and only if his attributes in the private key satisfy the passage tree demonstrated in the cipher text. By doing so, the user encrypting holds a complete force about the encoding system. Furthermore, the issued private keys will never be modified unless the whole system reboots [Hajny, J., & Malina, L. (2012)].

Dissimilar to the information secrecy, not much concentration is laid on ensuring clients' identity protection amid those intelligent conventions. Clients' identities, which are described with their properties, are by and large unveiled to key distributors, and in turn private keys are issued as indicated by their traits. In all the scenarios, it is proven to be a common characteristic that clients are willing to keep their identity a mystery while despite everything they get are their private keys. The proposed application permits cloud servers to control clients' login benefits without even a complete knowledge of their character data.

Mentioning about the data files, there are steps towards cryptographic enhancements which allows an outside specialist to decide if the data contents are available in lieu of legitimate concern for the data owner without any data spills or exchange of owner anonymity. It is a well-known fact that even the cloud customers will not have the strong belief that the cloud server is serving their needs to the extreme camouflage. A cryptographic course of action, with showed security re-laid on number-theoretic assumptions are all the more bewildering, at whatever point the customer is not flawlessly content with believing the security of the VM or the dependability of the particular
staff. These owners are somehow stressed to put together their specific keys along with their data files before making them available to the consumers.

Concerned to the circulated data stockpiling, information sharing is meant to be a fundamental handiness. For instance, social media users can give their colleagues a chance to see a portion of their private pictures; an exertion may give her agents access to a touch of fragile data. A reliable way to sufficiently share data is a testing issue. Clearly, customers can download the encoded data from the limit, share them to others after translation, and be that as it may lose the thought of disseminated memory. Data clients or owners need to have the mental ability to assign the passage benefits of sharing the data to others to the objective that they can directly reach the server to get to this data. In any of the case, it is not irrelevant to find a safe and secured way to deal with sharing midway data in circulating stockpiling.

Expect that Alice puts all her private photos on Dropbox, and she does not wish to open her photos to everyone. Due to the various data spillage likelihood Alice can't feel mitigated by essentially relying upon the security insurance segments given by Dropbox, so she encodes each one of the photos using her own specific keys before exchanging. Suppose one day, Alice's buddy Bob asks for what she shares in the photos expected control over each one of these years which Bob appeared in. Alice can then use the offer limit of Dropbox, however the point now is the way by which to assign the unscrambling rights for these photos to Bob. A possible decision Alice can pick is to securely send Bob the puzzle keys included.

Concurrent encryption has been purported to authorize information confidentiality while making duplication possible. A concurrent key is acquired after registering with the cryptographic hash estimation and it is used to unscramble the information duplicate. After the key generation and data encryption, clients obtain the keys and send the cipher text to the swarm. Since the
encryption operation is deterministic and is taken from the information content, identical information duplicates will create the same focalized key and henceforth the same cipher text.

To avoid unapproved or unauthorized access, a safe confirmation of the data ownership convention is mandatorily required to pass the verification step that the customer really owns the same disc when a data transcript is found. After the verification, consequent clients with the same document will be handed a pointer from the server without expecting to change the same disc platter. The encrypted document can then be downloaded by the customer using the pointer, which must also be decrypted by the owners by comparing information using their concurrent keys. Hence, concurrent encryption permits the cloud to perform deduplication on the cipher texts [6].

Nonetheless, past deduplication frameworks can't support differential approval copy check, which is significant in numerous applications. In such an approved deduplication framework, every customer is issued an implementation of the framework. Every record transferred to the cloud is likewise determined by an arrangement of benefits to indicate which sort of clients are permitted to do the copy check and fetch to the documents. Before giving his copy check demand for some record, the customer needs to get this document and his own particular benefits as inputs.

The client can locate a copy of this document if and only if there is a repeat of this record and a coordinated benefit put away in a swarm. For example, in an organization a wide range of benefits will be doled out to actors. Hence, as to spare expense and productive administration, the information will be moved to the capacity server supplier (S-CSP) in the general population cloud with indicated benefits and the deduplication procedure will be connected to store one and only duplicate of the same book [Camenisch, J., Neven, G., & Rückert, M. (2012)].

Summing up all the details discussed, a great amount of importance is being laid on research in cloud computing in both industrial and educational arenas. The facility to outsource
the storage and computation of data to cloud servers using internet attracts the users away from the hurdles of maintaining the same resources on-site. The services of the clouds vary from providing applications and infrastructures, to providing platforms, to developing customized applications. The data stored in the clouds mostly belonging to medical and social networks is highly sensitive and hence privacy and security are of major concern in cloud computing. The user has to be authenticated before he/she initiates any transaction on the cloud data. At the same time, it has to be ensured that the outsourced data is not tampered. The privacy of the user should also be entrusted so that the user’s identity is not revealed to others. In simple words, the user is accountable for the data and the cloud is accountable for the services being provided. For the secured data storage, the data need to be encrypted. One other thing to be noted is that this data is often modified and this needs to be considered in order to build an efficient and secured storage system. User authentication through public key cryptographic techniques and accountability for the operations on the cloud data have to be considered for building a secured system. Though the security to the data can be provided by various encryption mechanisms available in today’s world, the primary concern is about which users will be able to access the data and on what basis. Access control mechanisms are hence being studied in a rigorous manner to bring out a system more reliable than the current day centralized approaches. This paper deals about the various access control systems, proposes a system that uses distributed access control of cloud data, decentralized architecture for key management and will be resilient to replay attacks.

**Problem Statement**

The currently existing systems use a symmetric key approach which does not support authentication. The privacy preserving access control proposed by Zhao for cloud data uses a centralized approach where the keys and attributes to the users are distributed by a single key
distribution center. But this might be a problem if the KDC fails, which results in the unavailability of the data to the cloud users. Additionally, it might increase the load on the KDC if the number of users requesting access to the cloud data, applications and services is high. This might affect the performance of the KDC.

**Objective**

The primary objective of this paper is to propose and build an application that employs a decentralized access control mechanism using multiple KDC. The application also answers the problems of authentication, privacy protection, verification of the data source by initially changing the text to cipher and then generating the keys in a semi-practical format.

**Significance of the Study**

The proposed application implements Attribute Based encryption for high level data security, provides secured network architecture between owner, consumer and cloud. It also allows the cloud servers to monitor the user’s authentication even without any knowledge of their identity.

**Scope of the Study**

This scope of this paper is extended but not limited to developing an application for data storage security implementation in cloud networks. It also extends to have a deeper insight on cloud computing and networking, to present an architecture using Java as programming language and finally to analyze the results of the practical implementation versus theoretical approach.
The figure above explains the overall system design being proposed in this paper, all the included components and the data flow between them. There are multiple Key Distribution Centers (KDC), the cloud where data files will be stored, the different users, their authentication levels and actions performed on the data.
Table 1: Definition of Terms Used in this Document

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDC</td>
<td>Key Distribution Centre</td>
</tr>
<tr>
<td>ABE</td>
<td>Attribute Based Encryption</td>
</tr>
<tr>
<td>RBAC</td>
<td>Role Based Access Control</td>
</tr>
<tr>
<td>UBAC</td>
<td>User Based Access Control</td>
</tr>
<tr>
<td>ABAC</td>
<td>Attribute Based Access Control</td>
</tr>
<tr>
<td>DAC</td>
<td>Discretionary Access Control</td>
</tr>
<tr>
<td>MAC</td>
<td>Mandatory Access Control</td>
</tr>
<tr>
<td>NIST</td>
<td>The National Institute of Standards and Technology</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest-Shamir-Adleman</td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>JVM</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java Database Connectivity</td>
</tr>
<tr>
<td>SQL</td>
<td>Structural Query Language</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management System</td>
</tr>
<tr>
<td>ABS</td>
<td>Attribute Based Signature</td>
</tr>
</tbody>
</table>
Chapter II: BACKGROUND AND LITERATURE REVIEW

Cloud Computing

In a lay man’s language, Cloud Computing can be defined as one application being loaded which allows multiple users to log into a web-based application that hosts many other user programs. Remote machines would do everything from word processing to e-mail to complex programs of data analysis. A Cloud Computing system is generally divided into two parts, front end and back end. These both are connected through a network which is mostly internet. A general computer user or client will be the front end and the cloud will be the back end. Web browsers like Internet Explorer or Chrome are user interfaces and are considered as the front end. Various computers, data storage systems and servers combine to create the back end “Cloud” computing services and generally each application is associated with a different server.

Oberheide, J., Veeraraghavan, K., Cooke, E., Flinn, J., & Jahanian, F. (2008, June) research says all the current day mobile devices keep on approaching the capacities and extensibility of standard desktop PCs. Additionally, these gadgets are likewise started using the features of the Cloud computing and hence to face a large number of the same security dangers as desktops.

According to Schoo, P., Fusenig, V., Souza, V., Melo, M., Murray, P., Debar, H., ... & Zeghlache, D. (2011), Cloud computing is generally considered as an appealing administration model following the client’s requirements for venture and operations are minimized, and expenses are in immediate connection with the utilization and interest. The client demands and traffic monitoring are administered by a central server to make sure everything is running smoothly.
Advantages of Cloud

A software called middleware is used to follow a set of rules called protocols. The more the number of clients, the higher is the demand for data storage space and devices. A few reasons why industries rely on Cloud computing to store data and run the programs are as follows:

- Users or clients can access the applications and data from anywhere and at any time just by using their computers connected to the internet. In this way, the data access will not be confined to a particular hard drive, computer or a network.

- Cost incurred in purchase of physical devices and advanced hardware can be reduced drastically. It requires only a monitor, input devices like a mouse and keyboard and some processing power to run middleware and get connected to the cloud. No hard drive is needed as all the data is stored in the cloud itself.

- Companies will not have to bother about the latest software, its updates and license purchases. Cloud computing systems charge a metered fee and provide company-wide access to all the applications.

- The need for physical space consumed by servers and digital storage devices can be reduced by storing data on someone else’s hardware using cloud computing.

- If the cloud’s back end is a grid computing system, it could take advantage of the complete network’s processing power and thereby, the speed of the calculation can be significantly increased.

Besides having these many advantages, security and privacy are the biggest concerns of Cloud computing. In general, some people will not recommend the idea of handing over organization specific data to another company and hesitate to accept a cloud computing system as they may have to use the lock and key mechanism to keep their specific data under. But the debate
is that the cloud companies have reliable, secured measures and advanced protection techniques in place so as not to lose their clients’ data.

Privacy is another concern in cloud computing. Having the feasibility to log in and access the data and application in the cloud from any location there is also a possibility of compromising the client’s privacy. The ways that cloud companies protect client privacy are authentication techniques using usernames and passwords and authorization where a user can access only his or her relevant data and applications. Many law firms and universities are debating on a few questions on cloud computing like ‘Does the cloud computing company own the clients’ data?’, ‘Can a cloud computing company deny access to a client on clients’ data?’.[Boneh, D., & Hamburg, M. (2008)][ Liu, Z., Li, J., Chen, X., Yang, J., & Jia, C. (2014, July)]

**Access Control in Clouds**

**User Based Access Control (UBAC):** This is a method of security wherein the system, applications and services are secured at individual or user level. UBAC, also known sometimes as User based permissions, is basically implemented in the form of a simple login and password combination which either grants or rejects access. UBAC enhances granular control, but with a high management overhead as the changes to any permission settings need to be done for every user. Companies with high security concerns mostly prefer this type of access control.

**Role Based Access Control (RBAC):** It is a method of regulating access to system, network or cloud resources based on the individual user roles. Here, access is defined as the ability to create, view or modify some data or perform some specific operation. Roles may be defined based on the responsibility or authority of a particular user. RBAC, when properly implemented, lets the users do a wide range of authorized tasks by regulating their actions according to the constraints and
relationships dynamically. In the RBAC, it is easy to create, change or delete the roles as per the needs specified by the clients.

**Attribute Based Access Control (ABAC):** This is a distinguishable access control mechanism where access to the system or services in the cloud is granted after evaluating certain rules against the attributes of the entities. Attributes may be defined as a considerable characteristic to which a value can be assigned. Basically, ABAC relies on the evaluation of attributes of the subjects and objects, environment conditions and the access control rules defining the acceptable operations for subject-object attributes. ABAC systems are capable to enforce both Discretionary Access Control (DAC) and Mandatory Access Control (MAC). [Li, J., Ren, K., Zhu, B., & Wan, Z. (2009)][ Cui, B., Liu, Z., & Wang, L. (2015)]

**Cryptography**

According to The NIST Computer Security Handbook [NIST95], the term computer security is defined as “The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).” The process or methodology that makes sure the information and services are being safeguarded from unauthorized access, control or modification and destruction can be coined as security mechanism. In general, this security in networking where data will be transferred over protocols will be relied upon Cryptography. Cryptography is a word derived from Greek language meaning “Secret Writing”, in other ways, an art of making messages secured against external attacks by transforming them.
Encryption can be mentioned as one of the most reliable ways to assure that the information sensitivity is preserved. All the well-known encryption algorithms available in the current day for any information security transform the plain text into cipher text by applying different transformations, substitutions or both in parallel. The plain text is the original data and the cipher text is the result of applying the encryption methodologies to the plain text. The reverse engineering of encryption is called decryption where in some methodologies are applied on the cipher text to retrieve the plain text. These encryption algorithms are categorized into Symmetric and Asymmetric based on the keys being used for encryption. Symmetric encryption algorithms, also called as conventional encryption algorithms are those which use the same key for both the encryption and decryption processes. Similarly, asymmetric encryption algorithms are those in which two different keys, public and private keys are used for encryption and decryption.

Generally, a key will either be a numeric, alphabetic, alphanumeric or even a special symbol character. These keys will be used for encrypting the plain text into cipher text during the process of encryption and also while decrypting the cipher text to plain text during the process of decryption. These keys have to be selected with intensive care as the whole security of the
algorithm used will be directly dependent on them. It is proven that the strength of any encryption
algorithm relies on the length of the keys, secrecy of the keys, the initialization vectors and finally
the way they associate with each other.

**RSA Algorithm**

RSA, designed in 1978 was named after its designers Ron Rivest, Adi Shamir and Leonard
Adleman. It is known to be one of the best public key encryption algorithms till date for key
generation, encryption, decryption and digital signatures. The encryption blocks and the key in
RSA are of variable sizes. RSA is an asymmetric algorithm depending on the number theory. The
public and private keys are generated using prime numbers which are later used for encryption and
decryption. The data sender uses the receiver’s public key to encrypt the data and send it, while
the data receiver uses his own private key to decrypt the data and access it. The whole RSA process
can be distributed into three steps like key generation, data encryption and decryption.

RSA is not mostly preferred for commercial use as it has few drawbacks. The encryption
of data becomes very weak if the prime numbers used for key generation are very small which is
because one can easily decrypt the data with the help of random probability and also side channel
attacks. On the other hand, the encryption process consumes a high amount of time if large prime
numbers are used for key generation. Another negative point to be noted about RSA is that both
the prime numbers chosen should be of similar length and in practice this is very tough to be
satisfied. Padding techniques can be utilized in such scenarios which again is a process overhead
and consumes more time than it has to be. Typically, RSA requires keys of at least 1024 bits to be
considered as good keys. Any keys of size 2048 bits will be considered the best keys. RSA is
mostly used for securing a communication channel and authenticating the identity of the service
provider for key distribution.
The RSA works like this:

- Alice chooses two large primes \( p_A \) and \( q_A \).
- Alice computes \( n_A = p_A q_A \) and \( \phi(n_A) = (p_A-1)(q_A-1) \).
- Alice chooses an integer \( e_A \) with \( \gcd(e_A, \phi(n_A)) = 1 \), possibly at random.
- Alice computes \( d_A \equiv e_A^{-1} \pmod{\phi(n_A)} \).
- Alice’s public key is \((n_A, e_A)\). She distributes this. Her private key is \( d_A \). She keeps this secret. Alice can discard \( p_A \), \( q_A \), and \( \phi(n_A) \).
- If \( 2^k \leq n_A < 2^{k+1} \), Alice’s function of encryption for short texts (\( k \) bits or even less, so \( M < n_A \)) is: \( E_A(M) = M^{e_A} \pmod{n_A} \). Anyone can compute \( E_A(M) \). A long message is encrypted by splitting it into \( k \)-bit blocks, and encrypting each block separately. Note that each encrypted block has \( k+1 \) bits.
• Alice’s decryption function for short messages is: \( D_A(M) = M^{d_A} \mod n_A \), given \( 0 \leq M < n_A \). No one else other than Alice (or others who has Alice’s private key) can compute this. Note: \( D_A(E_A(M)) = (M^{e_A})^{d_A} = M^{e_A \cdot d_A} = M \mod n_A \) since \( e_A \cdot d_A \equiv 1 \mod \phi(n_A) \)

Once Alice has done this, she can

• Receive messages that are encrypted from Bob (or others), and

• Send digitally-signed messages to Bob (or anyone else). If Alice has to send encrypted texts to Bob, or to receive digitally-signed texts from Bob, Bob will need to choose his own public and private keys, \((n_B, e_B)\) and \(d_B\). Bob sends a short message \( M \) (at most \( k \) bits) to Alice like this:
  
i) Bob encrypts \( M \) as \( M^{e_A} \mod n_A \), and sends \( M^{e_A} \) to Alice. (Note Bob knows \( e_A \) and \( n_A \).) ii) Alice decrypts \( M^{e_A} \) as \( (M^{e_A})^{d_A} = M \mod n_A \). Thus Alice recovers \( M \). (Note Alice actually recovers the value of \( M \mod n_A \), but this is equivalent to \( M < n_A \).)

For big messages, Bob could divide the message to \( k \)-bit blocks, and encrypt separately every block. Alice would divide the message that is encrypted in \( k+1 \) bit blocks, and decrypt separately every block.

[Behrouz A Forouzan, “Networking and Data Communications”]

**Data Encryption Standard (DES)**

DES, an encryption algorithm that has been developed in the mid-1970’s, has gained the best standard by the National Institute of Standards and Technology (NIST). It was also adopted worldwide by various sectors of governments. It was always given a high priority when it comes to the financial industry. It is a block cipher and uses a block size of 64 bits. The key size is 56 bits and 8
Figure 4: DES Algorithm

bits will be used for parity. This configuration of the algorithm makes it susceptible to exhaustive search for keys for the current day computers and special hardware. DES is said to be easily breakable with the help of special hardware used by criminal organizations, governments or major corporations, but at the same time strong enough against the random hacking techniques. It is not being used these days for any new applications as it is getting weaker.

One other variant of DES called TRIPLE-DES (3DES) has been introduced later. This algorithm works by applying DES in three stages with different keys for each stage. The length of
the key in 3DES would eventually become 168 bits. In spite of being much stronger than DES, 3DES is slow when compared to some other block ciphers newly emerging. DES and its variants are still considered important, even if it seems to look like having less interested in today’s applications. DES, as the first ever block cipher played a very crucial role in providing the public with strong cryptographic algorithm and was hence deployed widely in the public sector.

The DES outline was especially useful for a cipher that was intended to be utilized just a couple of years. DES turned out to be an exceptionally strong cipher and it assumed control over a decade for any analytical attacks to be developed against it. Improvement of differential and linear cryptanalysis paved paths in order to better acknowledge the block ciphers and its design. Indeed, DES is most often used to explain different emerging crypt analytical techniques even if it is no more preferred for future encryption solutions. Despite the weakness of its small block and short key sizes, it is a bitter fact that even today there is no method as such that would totally break DES.

**Advanced Encryption Standard (AES)**

In 2001, NIST recommended AES as the latest encryption standard that replaces the DES algorithm. AES algorithm supports many combinations of data like 128, 192, 256 bits, etc. Depending on the length of the key used in encryption or decryption techniques, the DES algorithm is termed as AES128, AES 192, and AES256. The number of rounds in encryption and decryption process differ in DES algorithm based on the length of the key. AES has 10, 12, 14 rounds for 128, 192, 256 bits respectively. These rounds take the plain text as input and generates the final cipher text as output. In AES, the 128 bit data block is partitioned into four separate operational blocks where each block represents an array of bytes. These arrays of bytes are arranged in a matrix form of 4 rows and 4 columns that is termed as a state. In either encryption or decryption techniques,
the first operation that takes place is an Add Round Key. In the next step, the output undergoes nine more steps where each round involves four different transformations of data. These transformations include Sub-bytes, Shift-rows, Mix-columns, Add Round Key. No Mix-Column transformation takes place in the final tenth round. The below figure illustrates the complete transformation and encryption process. The decryption process is same as encryption process in AES except that the sequence of steps is reversed and inverse functions like Inverse Sub-bytes, Inverse Shift Rows, Inverse Mix-Columns, and Inverse Add Round Key are used. All other steps remain the same. [Gurpreet Singh., Supriya.,(2013)]

Each round of AES undergoes the below mentioned transformations:

![AES Algorithm Diagram](image-url)
**Substitute Byte transformation.** AES consists of 128-bit blocks of data, where each of the block includes 16 bytes. In this sub-byte transformation step, each 8-bit byte of a data block is converted to another block with the help of an 8-bit substitution box that is termed as Rijndael S-box. This box used as a lookup table to get the replacement of bytes for transformation.

**Shift Rows transformation.** Shift Row Transformation is a simple step where the bytes present in the last 3 rows of the state are cyclically shifted based on the location or index of the row. For instance, in the second row, one byte is shifted left in a cyclic manner. Similarly, in third and fourth rows, two and three bytes are shifted left in a cyclic manner respectively.

**Mix columns transformation.** Mix Column Transformation is similar to the logic of a matrix multiplication. In this step, each Column of the state is applied with a matrix multiplication. A constant matrix is taken as reference and is multiplied to every column in the state. The unique thing in this transformation is that the bytes of the state are taken as polynomials instead of integers or numbers.

**Add round key transformation.** Add Round Key is equivalent to applying a bitwise XOR operation on the 128-bits of a state and 128 bits of the round key. Hence, this transformation applies is its own inverse.

**Encryption in Clouds**

According to Cui, B., Liu, Z., and Wang, L. (2015), the capacity of a specific offering encoded information to various clients through open distributed storage might incredibly ease security worries over coincidental information spills in the cloud. A key test to outlining such encryption plans lies in the proficient administration of encryption keys. All the above mentioned access control mechanism uses Attribute Based Encryption (ABE) as a cryptographic primitive. Using this, the data is encrypted using some access policy and then stored in the cloud. Users will
be given a set of attributes and corresponding keys. The users can decrypt the data stored in the cloud only when they have the matching attribute set. This is being widely used in the field of health care where sensitive information about patients are stored in the cloud for access to various doctors, staff, researchers and policy makers.

Allison Lewko, An., and Waters, B. (2011), Brent Waters proposed a Multi-Authority Attribute-Based Encryption (ABE) framework. As per their theoretical account, any gathering can turn into a power and there is no requirement for any worldwide coordination other than the output of an underlying system of normal reference parameters. A gathering can essentially go around as an ABE power by hitting an open key and issuing private keys to several clients that mirror their traits. A client can encode information regarding any Boolean equation over characteristics issued from any picked set of abilities. In a long lasting, the proposed framework does not call for any focal power. In the process of developing the proposed framework, the biggest hurdle to be mentioned is to make it arrangement safe.

Not only storing the data in the cloud securely, but also ensuring the anonymity of the user is also equally important. Considering an example where the user might not want to be recognized when storing a sensitive information or where a user wants to respond to an article without disclosing his identity. In such cases, the user has to prove to other users that he is a valid user storing information without revealing his identity. Few cryptographic protocols like ring, mesh and group signatures are available, but can’t be used for specific reasons. Ring signatures will not suit for clouds due to the large number of users, group signatures are good when there is a possibility to assume the pre-existence of a group and mesh signatures won’t ensure whether the message is from a single or multiple users. Hence, a new protocol called Attribute Based Signature (ABS) is used. In ABS, users will have a claim predicate associated with each message which
helps to detect the user as authorized, without his identity being revealed. ABS and ABE can be combined to provide authenticated access control without the use of the user identity being disclosed.

According to Subashini, S., & Kavitha, V. (2011) Distributed cloud computing is an approach to expand the limit or include abilities powerfully without putting resources into the new framework, preparing new work force, or permitting new programming. It broadens Information Technology's (IT) existing capacities. In the most recent few years, distributed computing has developed from being a guaranteeing business idea to one of the quickest developing sections of the IT business. Anyway, as more data of users and organizations is stored in the cloud, concerns are increasing to become about exactly how safe a domain it is. Regardless of all the buildup encompassing the cloud, endeavor clients are still hesitant to send their business in the cloud. Security is one of the significant issues which decreases the development of distributed computing and inconveniences with information protection and information insurance keep on plaguing the business sector. The coming of a propelled model ought not arrange with the obliged usefulness and abilities show in the current model. Another model focusing on enhancing highlights of a current model must not chance or debilitate other vital highlights of the current model.

Ruj, S., Nayak, A., & Stojmenovic, I. (2011), proposed a propelled deduplication framework supporting approved copy check. In this new deduplication framework, a half breed cloud design is acquainted with take care of the matter. The individual keys for benefits won't be released to clients straightforwardly, which will be kept and overseen by the private cloud server. On these lines, the clients can't share these secret keys of benefits in this proposed development, which means that it can sustain the benefit key sharing among customers in the above direct development. To find a document token, the customer needs to transmit a solicitation to the private
cloud server. The instinct of this growth can be portrayed as takes after. To execute the copy, check for any document, the customer demands to make the record token from the private cloud server. The private cloud server will likewise match the client's character before issuing the comparing document token to the node. The approved copy check for this record can be performed by the client with people in general cloud before transferring this document. Taking into account the aftereffects of copy check, the client either transfers this document or runs program of work.

All the access control schemes proposed earlier by M. Li, S. Yu, G. Wang, F. Zhao, S. Ruj, and W. Wang are centralized in nature. W. Wang, used a symmetric approach that doesn’t support authentication and all others have used ABE. Though the scheme proposed by Zhao provides privacy preserving authenticated access control, it takes a centralized approach with one Key Distribution Centre being used to distribute keys and attributes to all the users. This single KDC might become a single point of failure and also might be difficult to maintain as the number of users supported by the clouds will be drastically increasing. Hence, it is highly preferred to take up a decentralized approach to distribute keys and attributes to users. It also helps clouds to have multiple KDCs at different physical locations. A decentralized approach was proposed by Yang, but was unable to answer the authentication issues of users who want to remain anonymous. Ruj also proposed a decentralized access control scheme that was unable to provide user authentication. Other users were able to only read the file when the creator alone has the write permissions and this is considered as another drawback. Later, Ruj proposed an extension to their previous work by adding the features of authenticating the user without revealing the identity. This paper implements the proposal of Ruj. ABS scheme has been used for authenticity, and privacy. The proposed application enables users to write multiple times by which the user can replace any old or outdated data with the new data. [Liu, Z., Li, J., Chen, X., Yang, J., & Jia, C. (2014, July)].
Chapter III: SYSTEM ANALYSIS

Existing System

Information access control turns out to be a high priority issue in distributed storage systems when the information is outsourced to the cloud servers that are un-trustable. The existing access control plans seem to be inappropriate as they require a complete, trusted cloud server or they produce scrambled duplicates. Even more, they rely on a single KDC for all the access key requests from different users.

In any scenario, the data owner will not be concerned or associated with the process that the data administrator provides data access to the data consumer in today’s distributed storage administration systems. And when this system relies on un-trusted cloud servers, the data access control will be an issue to be tested and taken additional care. And hence, it is no more appropriate to use the conventional server-based access control techniques for distributed storage systems.

Proposed System

In the proposed application system, Attribute Based Encryption is used to have the information in memory to be scrambled to the very last bit. The application allows cloud servers to control both the owners and users access permissions, even without knowing their identity information. It ensures that the client’s security in safeguarded in all possible ways.

The proposed application has both encryption and decryption functionalities. As part of encryption, the file that has to be secured is first divided into four equivalent sized byte arrangements and then encrypted using the AES encryption algorithm. After encryption, the files will be stored in the cloud. At the user end, the file would be downloaded once the user gets authenticated by the KDC with the owner key and n-authority key, using which the file will be
decrypted. All the password authentication and access key communication are done through Gmail using the mail addresses given by the users during the registration.

Figure 6: Overall System Flow
System Requirements

Hardware requirements:

System : Pentium IV 2.4 GHz.
Hard Disk : 500 GB.
Monitor : Any (1).
Mouse : Any (1).
Keyboard : Any (1).
Ram : 2 GB.

Software requirements:

Operating system : Windows 7/Linux.
Coding Language : Java 1.7
IDE : Eclipse/ Netbeans
Data Base : MySQL DB
Server : Apache Tomcat 7
Browser : Chrome/ IE/ Firefox
Chapter IV: SOFTWARE ENVIRONMENT

JAVA

In order to implement any application or a cryptographic technique, we have several ways and one such would be a programming language. Of all the programming languages available in the current day, Java would be a strong and appropriate one for developing and implementing applications. Due to its ease of use, robustness, security features and cross-platform capabilities, Java is accepted universally as frequently chosen language to provide better internet solutions. This research paper uses the Java 1.7 version.

Advantages of JAVA:

1. Java is easy to learn: Java was developed to be easy to use and is therefore easy to understand, code, compile, run, debug, and finally learn when compared to other programming languages.

2. Java is object-oriented: Object oriented concept enables you to develop modular and reusable code.

3. Java is robust: Robust stands for reliability. Java has much emphasis on checking for possible and unexpected errors, as Java compilers have the ability to detect many issues which would first appear during the time of execution in other languages.

4. Java is platform-independent: Platform independent is one of the most important uses of Java, as is the ability to move the code easily from one computer system to another and execute successfully. The ability to run the same Java program on different systems is important to software on World Wide Web, and Java achieves it 100% by being platform-independent on both the source and also binary levels.
5. Java is distributed: Java is developed to make computing, distributed across different places easy with the capability of networking which is integrated inherently into it. Writing programs of networking in Java is similar to sending and receiving information to and from a file.

6. Java is multithreaded: Multithreaded is the ability of a program to do several operations concurrently in a program. In Java, multithreaded programming is integrated into it, whereas in other languages, OS-specific functions have to be explicitly called to enable multithreading.

7. Java is secure: Java views security at its design phase itself. The Java language, JDK – Java Runtime, Developer Kit, compiler, interpreter, and finally runtime environment are incorporated with security.

Java Virtual Machine (JVM) can be defined as the basic functional concept in Java. It acts as the core of the Java platform and can run on various equipment dependent platforms. In order to facilitate instant programming which provides many useful abilities like Graphical User Interface (GUI) gadgets, Java API has been provided. It is a collection of multiple packages called libraries of interrelated classes and interfaces. Continuing further, the paper describes what are the special benefits provided by a particular part of the package in the Java API. The Java API and JVM can together protect the application systems from the environmental attacks.

A Native code will be the code that after you assemble it, the ordered code keeps running on a particular equipment platform. As a platform-independent environment, the Java platform can be a bit slower than local code. Nonetheless, savvy compilers, very much tuned mediators, and in the nick of time byte code compilers can convey execution near that of local code without undermining convey ability.
The most widely recognized sorts of projects written in the Java programming language are applets and applications. On the off chance that one has surfed the Web, you're likely officially acquainted with applets. Java applets are well known for their fast speeds and are suitable for the best for visualizations that are intensive as they access to 3D hardware acceleration.

An applet is a project that sticks to specific traditions that permit it to keep running inside a Java-empowered program. Notwithstanding, the Java programming language is not only to write charming, engaging applets for the Web. The universally useful, abnormal state Java programming language is additionally a capable programming platform. Utilizing the liberal API, you can compose numerous sorts of projects.

An application is a standalone program that runs specifically on the Java platform. An exceptional sort of use known as a server serves and backings customers on a system. Case of servers are Web servers, intermediary servers, mail servers, and print servers. Another specific project is a servlet. A servlet can practically be considered as an applet that keeps running on the server side. Java Servlets are predominantly chosen for building intuitive web applications. Servlets are like applets in that they are runtime expansions of utilizations. Rather than working in programs, however, servlets keep running inside Java Web servers, arranging or fitting the server.

**MySQL**

MySQL is an open source Relational Database Management System (RDBMS), owned by the Oracle Corporation. It has been known for its reliability, performance, ease of use and stands as a primary option to support most of the web based applications these days. It is also being provided with Oracle, MySQL cloud service to help industries build low cost database systems for their applications. The current application uses MySQL workbench to connect the application to
the database where all the keys and data will be stored. MySQL workbench is an additional tool for MySQL which enable to have a user friendly GUI to access the database tables, query them and retrieve data rather than using the command prompt. It also helps database administrators to visually create, design, model and manage databases.

**JDBC**

With an end goal to set a free database standard API for Java; Sun Microsystems created Java Database Connectivity, or JDBC. JDBC offers a nonexclusive SQL database access system that gives a steady interface to an assortment of RDBMSs. On the off chance that a database provider wishes to have a JDBC bolster, he or she should give the driver to every platform that the database and Java keep running on.

There are some programming packages and APIs that are outlined without a primary concern of objectives. On the other hand, JDBC is an API, in which, numerous objectives have been laid and driven the advancement of the API. These objectives, in conjunction with early commentator input, have settled the JDBC class library into a strong system for building database applications in Java. The objectives that were set for the JDBC are critical and will give you some knowledge with reference to why certain classes and functionalities carry on the way they do.

**JAVA MAIL API**

The Java mail API provides users a framework to send mails and messages independent of the platform and transfer protocols. It is available with both Java SE and Java EE editions as an optional packaged library. The API library is built with functions wherein one can specify all the attributes like, from, to, message subject, body and other network details. Using all these details and other functions of the library, one can send mails to one or many recipients with a single function call. It also provides with functions that can notify if the mail has been delivered or not.
**JFree Chart API.**

JFree Chart is one other library provided with Java that makes the users to show up their applications with high quality graphs and charts. The current version available supports line charts, bar charts, pie charts and x-y plots. Several objects and functions of the API coordinate to be able to produce the required graphs or charts.

**Apache Tomcat**

Apache Tomcat Server is an open source web server developed and supported by the Apache group. It consists of the servlet container which will be used by the Java servlets and Java server pages (JSP) technologies as a reference for implementation. These servlets, JSPs and their specifications are developed by Java Community Process under the collaboration of Sun Microsystems. Unlike the traditional application servers like Web Logic, Tomcat is a web server and supports only the applications that are built using any programming language and any IDE. Tomcat is used to run the web applications on the host and acts as local server that is built on the port 8080. It is composed of a web container named Catalina and bin directory. It can initiate the response methods or objects like GET and POST after loading all the http related requests.

**Key Management**

The proposed application implementation uses various keys at various stages of the application. They are as follows

- Public or private access key, a key generated and maintained by the key manager is a randomly generated binary key. It is used for authentication of the file owner.
- Owner key, in general, is a key associated with the file and its owner and is generated when the owner requests the KDC for file uploading.
- The N-key is a key that is generated by the KDC when the consumer requests the KDC to download or access a file in the cloud. This is generally associated with the file access and is built in Attribute based encryption.

- Other than the above three, each file that is uploaded into the cloud will be associated with a file key that remains unique for each file and in an encrypted form.
Chapter V: IMPLEMENTATION

The main roles in the application implemented are the data owner, data consumer, cloud admin and the n-Key authority (KDC). All these have their functionalities which are described as follows as per the workflow implementation.

Data Owner

The clients or the users who would like to save their data files in the cloud in a safe and secure way compose the data owners section. These data owners register in the application by providing all the necessary details and thereafter login with the received password. Once the owner logs in, he can request the KDC for an attribute key which is mandatory for him/her to upload the file he/she wishes to. The file that has been uploaded can be encrypted and finally uploaded in the cloud database. The owner is provided with an option to encrypt the file for the second time based on his requirement.

Data Consumer

The legitimate users who wish to consume the data that is stored in the clouds comprise the data consumer section of the application. Similar to the data owners, the data consumers also have to register themselves by providing all the required information and registering. Then after, they need to choose the file that they want to download and request the KDC for the access key to do so. Once the consumer submits the same in the application, he/she will be allowed to download the file after it is decrypted using the keys submitted.

Cloud Server

The cloud server stores the information of the data owners as well as the data consumers and gives access to them on the files according to their permissions. It allots specific ids for the files being stored in it and thereby makes it easier for the users to choose which file to download.
It produces the unscrambling tokens of a cipher text for the users by utilizing the keys of the users generated by the KDC.

**The Public Cloud**

A public cloud is a shared one which can be accessed by everybody who uses an internet connection or a credit card by paying as per usage without any subscription. Hence public clouds have a virtualized infrastructure that is usually shared by several different users. They are accessible with ease and can be managed from a dedicated self-service portal.

**The Private Cloud**

A private cloud is accessible only to specific users. In a private cloud, the services and infrastructures are managed on a private network. Unlike a public cloud, a private cloud is managed and owned privately. Its access is restricted to a single or part of a business. In terms of stability, security, data persistency and privacy, a private cloud seems much safer to organizations.

**The Hybrid or Mixed Cloud**

A Hybrid or Mixed cloud is a combination of private and public cloud. It includes the advantages of both private and public cloud for a company.

**The Community Cloud**

A Community Cloud is specifically used by a professional community that might include subcontractors or partners who work collaboratively on the same project or a cloud dedicated to government or state institutions.

**Key Distribution Centre**

KDC is an autonomous attribute authority center that stays in charge of issuing, renouncing and upgrading the users’ policies by part or character in its area. In DACMACS, each quality is connected with a solitary KDC, however every KDC can deal with a discretionary number of
properties or attributes. Each KDC has full control over the structure and semantics of its properties. Every KDC is in charge of creating an open property key for every characteristic it oversees and an access key for every client partner with their properties.

For ensuring the accuracy of data that is stored in the cloud server, we have designed a system that validates the correctness of the data stored along with avoiding the system behavioral discrepancies.

**Correctness of data stored.** In any circumstances, the system is designed to ensure that appropriate and relevant data is placed in the cloud. It also keeps the data stored in cloud undamaged all the times.

**Fast localization of error data.** Even when the server failures occur, the designed system dynamically repairs or modifies the server irrespective of the type of failure in order to prevent the data from being lost or corrupted or damaged.

**Dynamic operations support of data.** The proposed system supports all the dynamic operations whenever the user alters the data by performing any operations like insert, append, delete, etc.

**Dependability.** The system developed is away from dependencies. It decreases the effect of server failures or data errors and thereby safeguards the data from the unexpected failures, malicious threats, unauthorized data alterations, server colluding attacks, etc.

**Lightweight communication.** This framework also allows the users to perform security checks to verify if the data stored is correct and not corrupted without least expense.

The architecture designed authenticates users and enables them to remain anonymous while using the cloud. Many previous implementations provide only binary results of the data storage state among multiple distributed servers, but our system provides localization of error data along with binary results.
The below figure depicts the architecture of typical Key Distribution Center used in our proposed system. Any user who owns a set of files, stores them in the cloud server in encrypted form and by using indexing.

![KDC Architecture Diagram](image)

**Figure 7: KDC Architecture**

Whenever the user logs into the system, two steps of authentications take place. One is with TPA (Trusted Party Authenticator) and the other with the KDC (Key Distribution Center). It involves the following operations:

**Service Request to TPA.** Initially, any new user needs to get registered into the system by entering his first name, last name, phone number, email, username, etc. The user registers with his/her original identity and validates his identity against TPA. For registration, the user sends a request to TPA. After successful registration, he/she gets an email with a temporary password which he can change at any point of time by using “Change Password” option. All existing users can directly login into the system without registering.
TPA Policy Creation. The TPA (Trusted Party Authenticator) provides the rules and regulations that need to be followed along with the token or secret code Reader or Writer or Creator.

User File Upload. The user has an option to upload a file after he successfully logs into the system. He can give any name to the file he intends to encrypt. The file creator of the system encrypts the file after successful upload of the file and places the encrypted file in the cloud.

KDC Key Generation. The Key Distribution Centers generate various keys and assign them to various users after receiving the tokens or secret codes from the users.

Cloud Admin. The cloud admin manages the set of KDCs (Key Distribution Centers) and the TPA (Trusted Third Party). This admin is responsible to set any rules and norms that need to be followed by the KDC and TPA. It also monitors the policies of key generation and notifies abnormal behaviors.

Design Goals

In this research paper, we targeted to develop an efficient method for dynamic data storage, verification and operation in order to reach the following goals:

- Access control of the data is distributed across the cloud in such way that only authorized individuals having valid attributes and identity will be able to access the cloud.
- The system is designed in such a way that replay attacks are avoided.
- With the efficient and two level authentication of both TPA and KDC, the identity of the user can be safe in the cloud environment.
- Multiple KDCs exist in the proposed system used for key management making the architecture decentralized.
At any point of time, two users cannot collude and access the data by themselves or authenticate by themselves in case they are not individually authorized because both the access control and authentication are collusion resistant.

Once a user is revoked the data access, he/she cannot access the data anymore.

**Application Screenshots**

The following screenshots illustrate the functionality of the system proposed in this research paper.

![Application Home Page](image)

*Figure 8: Application Home Page*
Figure 9: Owner Registration Page

Figure 10: Owner Successful Registration Pop Up
Figure 11: Owner Login Page

Figure 12: Owner Details Page
Figure 8: File Upload Request

Figure 9: Request Success Message
Figure 10: KDC Login Page

Figure 11: KDC Database Page
Figure 12: File Upload Page of Owner

Figure 13: Uploaded File Successful Encryption Message
Figure 19: Owner Data Base Page

Figure 20: Re Encryption of Data Page
Figure 21: Re Encryption Success Page

Figure 22: Data Consumer Registration Page
Figure 23: Data Consumer Login Page

Figure 24: Data consumer Details Page
Figure 25: File View Page for Consumer

Figure 26: Access Request Pop Up
Figure 27: File Database View

Figure 28: Access Key Verification Page
Figure 29: Verification Success Pop Up

Figure 30: File Download Page
Figure 31: Cloud Admin Access Page

Figure 32: Statistics Page
Chapter VI: SYSTEM DESIGN

Data Flow

The data flow diagram, also called as bubble chart is a formal graphical representation of the application or system which describes the input data to the application, the different processes that are done on the input data and the final output data being generated by the application.
Data Flow Diagrams

Data Owner

Start

Data Owner

File Upload

Encrypt File

Re-encrypt file

Visit files data

Unauthorized User

Stop
Data Consumer:

Start → Data Owner → Check file list → Request access → Provide keys → Download file → Stop

Unauthorized User → Data Owner → Check file list → Request access → Provide keys → Download file → Stop

Cloud Admin and KDC:

Start → User Check → Show File Database & Graph → Stop

User Check → View access request and respond → Stop

Figure 14: Data Flow Diagrams
Class Diagram

A class diagram in Software Engineering is a class graph in the Unified Modelling Language (UML) which represents the modules in terms of classes that compose of the attributes and the functions within them. The below diagram shows all the four classes used in the application, its attributes and functions.

Use Case and Sequence Diagrams

A Use Case diagram is a graphical representation of interaction of user with the elements of the system. It illustrates the relationship of the user, the use cases that the user is involved,
actors, system, etc., but it does not clearly state the exact order of performing the steps to attain the goals of all use cases. It is used to give an overview of who is using the system or application and what possibly can they do with it. A use case diagram is used at the system design phase of SDLC (Software Development Life Cycle). It helps in analyzing the system to clarify, identify, and organize system requirements.

A typical Use Case diagram describes the following items:

- All possible scenarios of application or system interacting with organizations, systems, people, external systems.
- Lists the set of the goals that might help the users or actors to achieve.
- The entire scope of the system.

A Sequence Diagram focuses primarily on illustrating the interaction between different objects of the system. It also specifies the order in which the objects are interacted. These diagrams depict the flow of logic in the system that enables one to easily document and also verify the logic and flow. It explains how the business works to programmers making them understand the business flow better. These diagrams show the time span of the events by showing if they are active or not. These are also termed as event scenarios or event diagrams.
Figure 34: Use Case and Sequence Diagrams
Chapter VII: SYSTEM TESTING

Any system that is developed needs to go through a phase of testing. It is a mandatory thing to find out the bugs and shortcoming in any application. It gives the user enough idea about the working of various components and functions both individually and also when integrated as a whole. Testing is of several types like Unit testing, integration testing, functional testing, system testing, white and black box testing. Each of these tests addresses some particular necessity as required by the application.

Types of Testing

Black box testing. Black box testing is a type of software testing that ignores the internal design of the system and targets only on the verification of expected output for a given input. Both positive and negative test case scenarios are validated in this testing. This testing is also termed as a functional testing.

White box testing. White Box testing validates the internal design or application logic of the software component under test on the basis of knowledge of the functionality. This testing is also termed as Glass Box testing or structural testing. White box testing is often used for verification, whereas Black box testing is used for validation purposes.

Unit testing. Unit testing is done typically at the beginning of the software development phase. Any small unit or software component developed by the programmer has to be unit tested. This type of testing is usually done by the programmer itself instead of the testing team because of the need for in-depth knowledge of the software unit developed for reliable testing purposes. The programmer needs to test the software component developed with different input values and validate the expected output. It is a form of white-box testing.
**Integration testing.** Integration testing plays a major role in the application testing. This testing ensures that the application or system is working as expected, even after integrating with other modules or external interfaces. The software and hardware interaction used across multiple interfaces are also tested in Integration testing. It is a form of both white box and black box testing. This type of testing is generally performed against client server frameworks, distributed environment that interacts over a network.

**Functional testing.** Functional testing is a testing strategy where the functionality of a software component is tested to work as expected as per the defined client or business requirements. This testing usually ignores the internal logic and targets on only the expected output. This is a form of black box testing.

**System testing.** System testing is performed after the complete system is implemented in a typical software project environment. This testing ensures that the entire system is working properly in different environments such as different operating systems or web browsers. This is a form of black box testing where the combination of system parts is tested based on the overall requirements specified.

**Stress testing.** Stress testing ensures that the system is functioning as expected, even in unfavorable situations like heavy system or database load, complex database queries, overloading system capacity, system crash or hang, power off, etc. The system is tested beyond the testing requirements specifications and focuses mainly on how the system behaves in failing scenarios. This is a form of black box testing.

**Performance testing.** Performance testing plays a vital role in any real time application where a large number of users are involved. This testing verifies the effectiveness and speed of the system
under test. It ensures that required results are generated in acceptable time. It is a form of black box testing.

**Regression testing.** Regression testing is performed to test the application or system after applying some modifications. This testing ensures that the system is working as expected, even after the changes made to it. Usually different automated tools are used to perform this testing. It is a form of black box testing.

**Acceptance testing.** Acceptance testing comes into picture after the Software System is completely built and delivered to the customers. This testing is generally performed by the users/customers instead of software testers to ensure that the system delivered is functioning as per the requirements. This testing is also a form of black box testing.

**Test Objectives**

- Check if all the UI fields are working correctly for valid entries.
- Different pages of the application are linked properly and are in line with the work flow.
- The different messages that notify the user about the status are not misleading.

**Features Tested**

- Tested all UI fields and elements with different types of valid and invalid data.
- Tested if any duplicates are being allowed or not.
- Tested if the control is being properly moved over the application.
- Tested at the boundaries of the functions wherever applicable.
- Tested if the warning, error and success messages or prompts are displayed properly.
- Tested if all the DB connections are correctly pointed and working.
- Tested the application in both positive and negative scenarios.
All the test cases have been passed in all the different scenarios and no defects have been observed.
Chapter VIII: CONCLUSION

Introduction

In the proposed paper, the ideas of various encryption methodologies like IBE, ABE, KP-ABE, and CP-ABE are discussed with details of their favors and also inconveniences. Various side effects with the centralized approach of access control mechanisms have been outlined. Also, the benefits of the decentralized access control mechanism have been explained which uses multiple KDC architecture with anonymous authentication. A plan has been laid, investigated upon, discussed and explained on how to build an application that answers the distributed access control mechanism in the cloud architecture with anonymous authentication.

Results & Conclusion

As proposed and planned, an application has been developed fulfilling all the requirements. Intense care and concentration has been put in the whole process starting from picking the technologies, analyzing the correlations and implications, developing and testing the application rigorously. The end result was shown in the form of screenshots. All the necessary testing has been to done to ensure the functionality. The application is implemented in the local databases and found to be better comparative to the centralized access control mechanism.

Future Work

The current scope of study is confined only to one cloud and the application has been developed to use only one cloud at a time. The future scope for the study could be trying to implement a similar architecture across different clouds and different services. Also, deploying the developed application on to a real cloud architecture would also define the future scope.
References
Cui, B., Liu, Z., & Wang, L. (2015). Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage.


