Access Control Scheme for Data in Cloud with Anonymous Authentication

[1]Mr.Gholap Nilesh, [2]Prof. Pritesh Jain
[1]PG Student, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal
[2]Assistant Professor, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

---

***Abstract* - Cloud computing is logically developed technology to stored data from more than one user. Cloud computing is location that allows users to store the data. The important thing is to see that the cloud provider meets the security requirements of the application. In order to have a secured cloud computing, one have to consider the different areas like architecture of cloud computing, interoperability, portability , security, business continuity, data centre operations, Application Security, Key management and encryption, identity and access management. The cloud verifies the authenticity of the series without knowing the user's identity in the proposed scheme. Our features is that only valid users are able to decrypt the stored information. This access scheme support anonymous authentication. This scheme is decentralized access and robust which is different from other access.**

***Keywords* - Access Control, Cloud Computing, Key Policy, Attribute-based signatures (ABS), Attribute Based Encryption (KP-ABE) ,Anonymity Authentication, Key Management.**

I.INTRODUCTION

Cloud computing is a promising computing model which currently has drawn far reaching consideration from both the educational community and industry.In cloud computing users can contract out their calculation and storage to clouds using Internet. This frees users from problem of maintaining resources on-site. The services like applications, infrastructure and platforms are provided by cloud and helps developers to write application. By joining a set of existing and new procedures from research areas, for example, Service-Oriented Architectures (SOA) and virtualization, cloud computing is viewed all things considered a computing model in which assets in the computing infrastructure are given as services over the Internet.Today's computing techniques have attracted more people to store their important data on third-party servers either for sharing easiness or for cost reduction. When people uses features of these new emerging technologies and services invented, their concerns about data security also important. Usually, users would like to make their important and confidential data only accessible to some authorized users.User privacy is also required so that the cloudor other users do not know the identity of the user.Much of the data stored in clouds is highly sensitive, forexample, medical records and social networks. Security andprivacy are, thus, very important issues in cloud computing.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 [1], [2].

Recently experts addressed Anonymous authentication for data archiving to clouds[3]. Anonymous authentication is the procedure of accepting the client without the details of the client. So the cloud server doesn't know the details of the client, which gives security to the clients to conceal their details from other clients of that cloud. Access control in clouds is gaining attention because it is important that only authorized users have access to valid service. 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. However, 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. Security and privacy assurance in clouds are analyzed and tested by numerous researchers. [3] gives storage security utilizing Reed-Solomon eradication correcting codes. Utilizing homomorphic encryption, [4] the cloud gains cipher text and furnishes an encoded value of the result. The client has the capacity to translate the result; however the cloud does not comprehend what data it has worked on.

In this paper key policy Attribute Based Encryption scheme is used to control unauthorized access. In addition revocation scheme is used for time based file assured deletion

Ojectives:

1.Distributed access control of data stored in cloud so only authorized users with valid attributes can access them.
2. Authentication of users who store and modify their data on the cloud.
3. The identity of the user is protected from the cloud during authentication.
4. The architecture is decentralized, meaning that there can be several KDCs for key management.
5. Revoked users cannot access data after they have been revoked.
6. The proposed scheme is resilient to replay attacks.
7. The costs are comparable to the existing centralized approaches, and the expensive operations are mostly done by the cloud.

II.RELATED WORK:

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. Access control in clouds is gaining consideration on the grounds that it is imperative that just authorized clients have access to services.

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. Access control in online social networking has been studied in [5].

The work done by [6] 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. The scheme

In [9] uses a symmetric key approach and does not support authentication.

Multi-authority ABE principle was concentrated on in [7], which obliged no trusted power which requires each client to have characteristics from at all the KDCs.

Matthew Pirretti and Brent Waters introduce a novel secure information management architecture based on emerging attribute based encryption (ABE) primitives also they propose

cryptographic optimizations in Secure Attribute Based Systems in 2007. Decryption decrypts a ciphertext

encrypted by the Encryption. This process begins with the decrypting party verifying that they have the required

attributes. The party performing decryption will then use their attributes to decrypt the decrypt the ciphertext in order to obtain the AES key.

III.SYSTEM ARCHITECTURE:

The architecture is decentralized, meaning that there are several KDCs for key management. There are three different users, a creator, a reader and writer. Creator receives a token from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc. For example, these can be servers in different parts of the world. A creator after validating the token to one or more KDCs, receives keys for encryption/decryption and signing. The message is encrypted under the access policy. The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy, to prove his/her authenticity and signs the message using this claim. The ciphertext with signature is sent to the cloud. The cloud verifies the signature and stores the ciphertext. When a reader wants to read, the cloud sends ciphertext. If the user has attributes matching with access policy, he/she can decrypt and get original message. Writing process is similar as file creation.

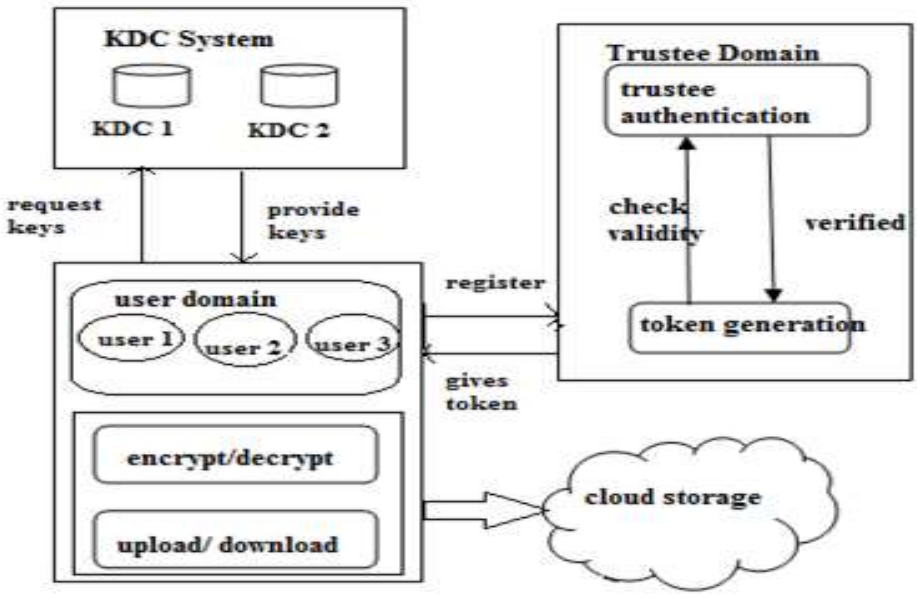

Figure 1.1: System Architecture

By assigning the verification process to the cloud, individual users are relived from time consuming verifications. When a reader wants to read data in the cloud, it tries to decrypt data in ciphertext form using the secret keys it receives from the KDCs. If user has sufficient attributes matching with the access policy, then he/she can decrypt the information stored in the cloud.

The main modules are

(a) Trustee

A trustee can be someone like the federal government who is responsible for managing social insurance numbers etc. On presenting her/his id like health/social insurance number, the trustee gives her/him a token.

(b) KDC

The function of KDC is to distributes secrete key and writer key to all authentic users. Cloud has many KDCs in different locations in the world. If there is single KDC then it is centralizing approach and if multiple KDCs then decentralize approach. KDC is a key distribution center which generates keys and assign the keys to the article, each article have unique keys. Each article has separate reading and writing key. KDC generates keys using dynamic key generation algorithm using random function. In this system use of decentralized access that means more than one KDC are uses at different locations in the world. If one KDC is get failed then it automatically switches to another available KDC.

(c) Creator

Authorized Creator can write the file and upload to the cloud. If any other user wants to read or modify the file of creator, he has to then send the request to the KDC to get access keys to the particular file. If KDC provide the key then only user is able to read, update or modify that file.

(d) Client

i. Reader

Reader can read the file online with help of secret key (SK).Reader per- form the request to the cloud server for the key. When the Reader enters the valid key only then the file is visible to the reader. When user request data from cloud then Cloud send cipher text C. Decryption proceeds using algorithm ABE. Client is any user who wants to read or write or modify the files which are stored on the cloud server. If client want to read the file at that time he/she requires secret key to decrypt the file. If attributes of reader matches with the access policy then he/she can download the file.

ii. Writer

When the writer want to upload or modify file then if the writer key valid then he/she can update the article. To write the already existing file, User send its request to Cloud, then cloud will send the ciphertext C and ask for key(WK) If key matches, then user is authenticated and allow to write. If client want to write or modify the file he/she requires secret and writing key.

(e) Cloud Server

Cloud server is used to storage of data where user can upload the data. When user wants to upload files first he/she has to send request to KDCS. Then the KDC generate the secret key and writer key. 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. When any user request for file then cloud ends the data in encrypted format. And user can view that file by providing the valid keys.

4.1.1 Data Storage in Clouds

- The KDCs are given keys for encryption/decryption and ask for signing/verifying.
- The users obtain attributes and secret keys from one or more KDCs.
- The message is encrypted using the equation C = ABE: Encrypt(MSG; key)

4.1.2 Reading from Clouds

- When a user request a data from cloud, the cloud sends Ciphertext c using SSH Protocol.
- Decryption proceed using the equation ABE: Decrypt(C; SKi;u)

4.1.3 Writing to the cloud

- To write to an already existing file the user must send his/her message during file Creation.
- The cloud verifies WK (writing key) and only if the user is authenticated he/she can write on the file.
-

4.2 Algorithm

(a) ABE (Attribute Based Encryption) it works under the following stages.

- Setup: This is a random algorithm that takes no input other than security parameter. It outputs the public parameters P and a master key K.
- Encryption: This is a random algorithm that takes as input a message m, a set of attributes n, and the public parameters P. It outputs the cipher text C.
- Key Generation: This is a random algorithm that takes as input an access structure A, the master key K and the public parameters P. It outputs a decryption key D.
- Decryption: This algorithm takes as input the ciphertext C that was encrypted under the set n of attributes, the decryption key D for access control structure A and the public parameters P. It outputs the message M if n 2 A.

(b) ABS (Attribute Based Signature) An Attribute-Based Signature (ABS) scheme is depend on a possible attributes A and message space M, and consists of the following four algorithms.

- ABS. Setup (to be run by a signature trustee): Generates public reference information TPK.
- ABS. Setup (to be run by an attribute-issuing authority): generates a two keys PK and SK.
- ABS. AttrGen: On input (SK, A_A), outputs a signing key SK.
- ABS.Sign:On input (PK = (TPK, PK), SK, m ϵ M, ϒ), where (A) = 1, outputs a signature σ
- ABS. Ver: On input (PK = (TPK, PK), m, Υ, σ) outputs a Boolean value 0 or 1.

(c) Paillier Encryption The Paillier Cryptosystem is well known Homomorphism encryption. It is an asymmetric algorithm for public key cryptography

IV. PROPOSED SYSTEM:

Although we proposed a decentralized approach, their technique does not authenticate users, who want to remain anonymous while accessing the cloud. In an earlier work, proposed a distributed access control mechanism in clouds. However, the scheme did not provide user authentication. The other drawback was that a user can create and store a file and other users can only read the file. Write access was not permitted to users other than the creator. In the preliminary version of this paper, we extend our previous work with added features that enables to authenticate the validity of the message without revealing the identity of the user who has stored information in the cloud. In this version we also address user revocation, that was not addressed. We use ABS scheme to achieve authenticity and privacy. Unlike our scheme is resistant to replay attacks, in which a user can replace fresh data with stale data from a previous write, even if it no longer has valid claim policy. This is an important property because a user, revoked of its attributes, might no longer be able to write to the cloud. We, therefore, add this extra feature in our scheme and modify appropriately. Our scheme also allows writing multiple times which was not permitted in our earlier work.

ADVANTAGES OF PROPOSED SYSTEM:

- It provides authentication of users who store and modify their data on the cloud.
- It revoked users cannot access data after they have been revoked.
- Costs are comparable t**o the existing centralized approaches.**

V.SYSTEM CONFIGURATION:-

HARDWARE REQUIREMENTS:-

- Duel core Processor
- 512 MB RAM
- 80 GB HDD

SOFTWARE REQUIREMENTS:

- JAVA (AWT, SWING, SERVLETS)
- Netbeans IDE
- SQLServer 2008
- Windows XP/ Windows 8(32 bit/64 bit)

VI. DETAIL DESIGN

1. System Initialization

The System Initialisation is the initial process for the system. The system get initialised for the user. The single user or the group of user can register within the system.

2. User Registration

The User have to register themselves under the registration module. According to the user credentials, which will be provided by the users, the user will get the private key. And by using that private key the user can then upload or download the required data in the future.

3. KDC setup

We emphasize that clouds should take a decentralized approach while distributing secret keys and attributes to users. It is also quite natural for clouds to have many KDCs in different locations in the world. The
architecture is decentralized, meaning that there can be several KDCs for key management**.**

4. Attribute generation

The token verification algorithm verifies the signature contained using the signature verification. This key can be checked for the consistency. There are two types of access permissions are given to the user. Read and read
write access. The user will allow the another user to only read the content of the sent file or they can permit to read and make some required modification and write it back again.

5. Sign

The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy Y, to prove her authenticity and signs the message under this claim. The ciphertext C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the ciphertext C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message.

6. Verify

The verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it
receives from the KDCs.

VII. SCREENSHOT

In this page user has to fill the information and attribute he/she posses.

i) Main Menu:

Figure 2: Client Side Main Menu

This page contains all the modules that have integrated with this paper.

ii) Request Token

Figure 3: Token Request to Trustee

User will request token to trustee, trustee will assign a unique identifier to each user

iii) Key Generation

Figure 4: Key Generation

When user gives token to one or more KDCs, KDC assigns a key for encryption and decryption to each user

iv) Cloud Login

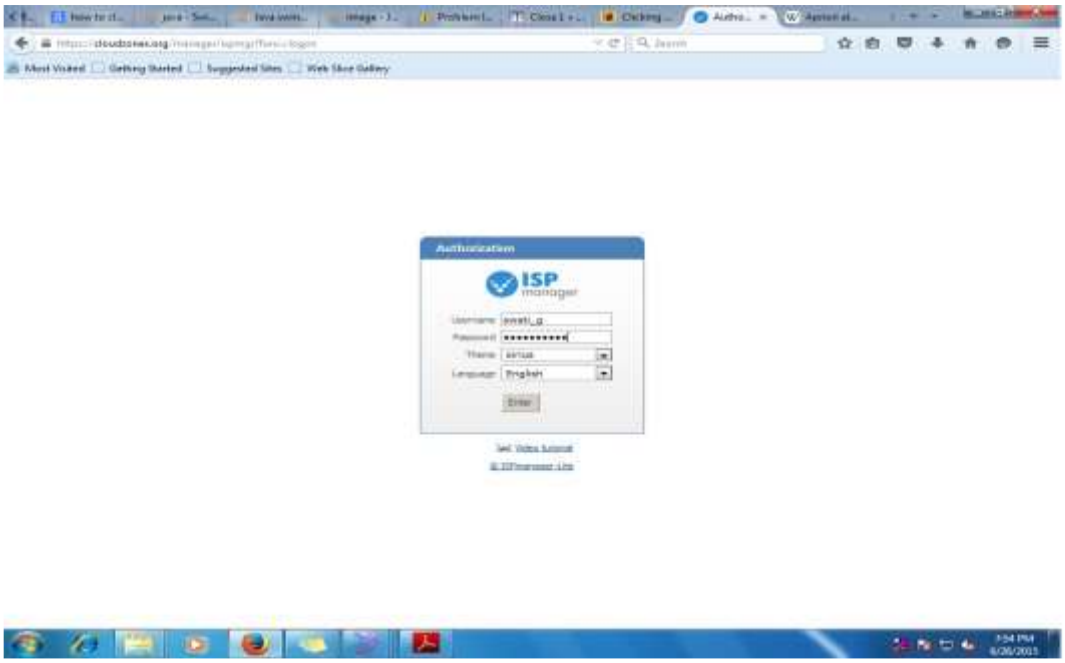

Figure 5: Cloud Login

This snap shows the cloud login. In this project as for cloud storage online cloud is used, user must have his/her login details

v) File uploading with Encryption:

Figure 6: File uploading with Encryption

User can upload data by using the key he/she taken from KDC the data is encrypted first n then uploaded to cloud

vi) File downloading with Decryption

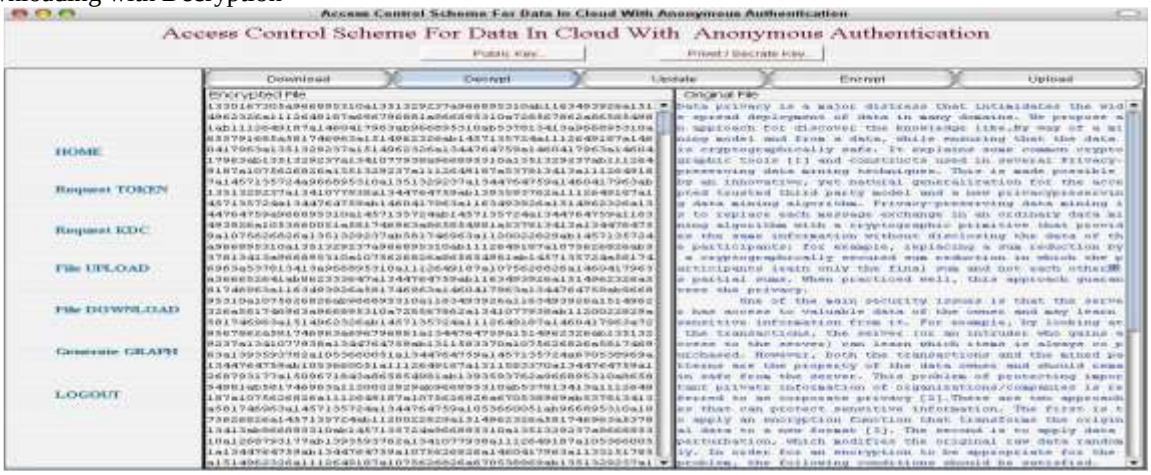

Figure 7: File downloading with Decryption

When user want to download data from cloud if he/she has attribute matching with access policy can only download a file.

VIII. RESULT ANALYSIS

1) Graphical representation of comparisons with DES (Time):

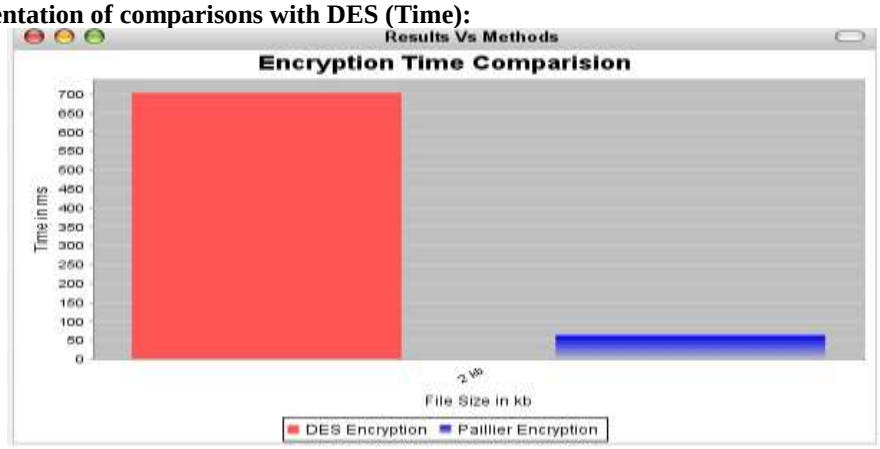

Figure 8: Graphical representation of comparisons with DES

This graph shows the comparision of Pailier with DES from figure it is clear that Pailier is time efficient than DES.

2) Graphical representation of comparison with DES (Time):

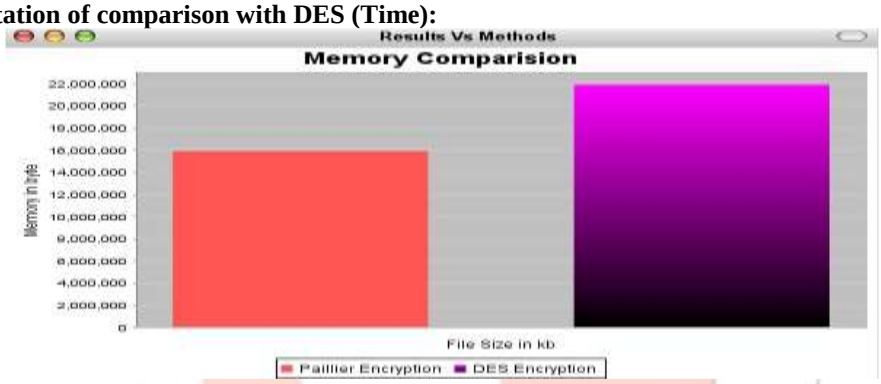

Figure 9: Graphical representation of comparisons with DES (Storage)

This graph shows the comparisons of Paillier with DES; from figure it is clear that Paillier is memory efficient than DES.

8.2 Time Performance

This system compares the time performance between Symmetric Key algorithm and Asymmetric key Algorithm. Graphs show the execution time of implementations over the various instances. The Paillier Algorithm (Homomorphism asymmetric key encryption) provides an alternative to the Symmetric key encryption algorithm (DES). Paillier showed better performance over Symmetric encryption algorithm in term of encryption time. Figure 8.11 shows the graph comparison of encryption time in milliseconds among DES and Paillier with respect to Table 2.

Input File Size(KB)	Symmetric Key Algorithm	Asymmetric key algorithm (Homomorphism)
2	700	80
31	300	180
50	150	120

Table 1: Encryption Time Comparison between Symmetric Key encryption algorithm (DES) and Asymmetric key encryption Algorithm (Paillier)

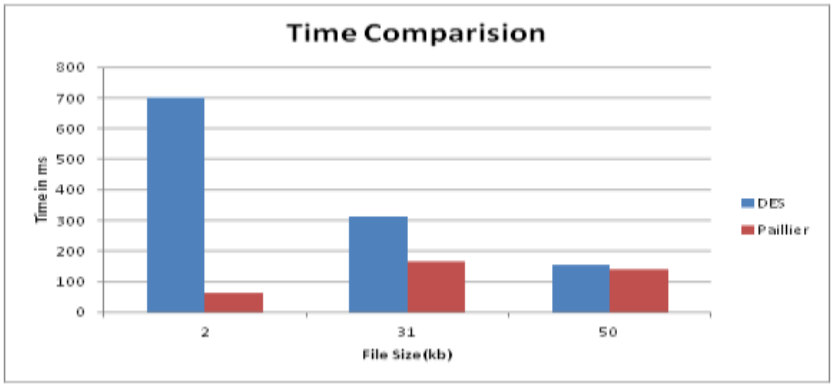

Figure 10: Graph comparison of Symmetric key algorithm (3DES), and Asymmetric key algorithm (Paillier) (Time)

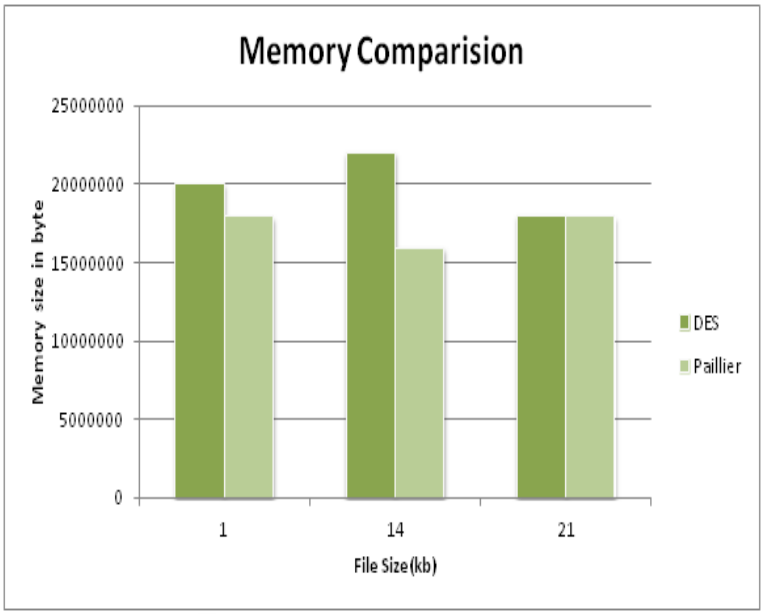

Figure 11: Graph comparison of Symmetric key algorithm (3DES), and Asymmetric key algorithm (Paillier) (Memory)

IX. CONCLUSION

In this system, a secure cloud storage model is addressed. This system is decentralized in nature with anonymous authentication. Using this system uploading and downloading of a file to a cloud with Encryption/Decryption is more secure as this system uses Pailier algorithm which is computationally more complex. Revocation module will remove the files of revoked users from the system. The cloud does not know the identity of the user who stores the information. Key distribution is done in decentralized manner using multiple KDC structure

XI.REFERANCES

[1] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM, pp. 441-445, 2010.

[2] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptography and Data Security, pp. 136- 149, 2010.

[3] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.- June 2012

[4]. C.Gentry, "A fully homomorphic encryption scheme", Ph.D. dissertation, Stanford University, 2009, http://www.crypto.stanford.edu/craig.

[5]. S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-based access control in social networks with efficient revocation," in ACM ASIACCS, 2011.

[6]. F. Zhao, T. Nishide, and K. Sakurai, "Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems," in ISPEC, ser. Lecture Notes in Computer Science, vol. 6672. Springer, pp. 83–97, 2011.

[7]. M. Chase and S. S. M. Chow, "Improving privacy and security in multi authority attribute-based encryption," in ACM Conference on Computer and Communications Security, pp. 121–130, 2009.