ISSN:0975-9646

Decentralized and Vigorous Unidentified Validation Access Control System

G.Anusha[1], R.Sathiyaraj[2]

[1]*PG Student,* [2]*Assistant Professor*

[1 2]*Department of Computer Science and Engineering,*

Madanapalle Institute of Technology and Science, JNTUA, India.

Abstract— **Cloud provides storage as a service where the data can be stored and retrieved remotely. Authentication and secrecy are crucial issues in cloud computing. In existing, it is based on Attribute Based Encryption Technique which is a not a decentralized approach i.e., centralized approach, in which a particular Key sharing Centre generates secret keys and distributes to sever users using asymmetric key approach. We use a decentralized access control technique to store data by providing security in clouds and also we hide the attributes and key access rule of a user. The cloud validates the authentication without knowing the users characteristics previous to the data store. By using this approach only authorized users have right to use the suitable attributes. We propose to generate a secret key for each user independently, the security for the files stored in the cloud storage is increased. This is achieved by using advanced key management technique which provides the security for the files stored in the cloud storage by generating a secret key for each user. In future, time based file revocation scheme can be used to assure the deletion of a file. When time limit of a file expires, we implement the policy based renewal of time to that file.**

Keywords— **Data security, Key policy ABE, Cipher Text policy ABE, Key management, cloud computing, Secret key**

I. INTRODUCTION

Cloud Computing is a most popular and currently using trend for IT. Given the trends, nearly almost all the majority of IT departments will be running applications in cloud. Cloud Computing associated to different applications and services that run spread network using basis resources and accessed by normal internet protocols and networking rules.

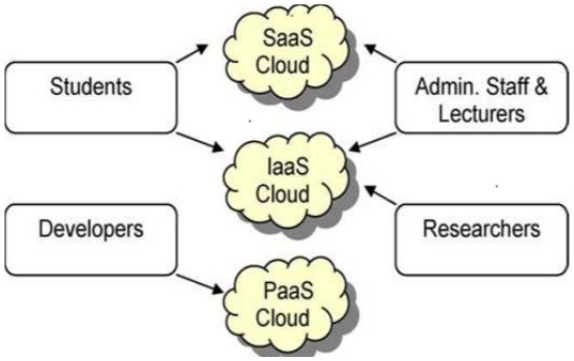

Fig1. Cloud services

The above diagram represents the university with an information technology infrastructure that caters for the needs of students, teaching staff and management, research staff and software developers.

The Service providers offer some techniques that elaborate the services. Among all the service models the best and most known services are software, platform, and infrastructure as a service. There is a deployment model, which is based on division of classes and services. In addition to that it discusses the purpose and location of the cloud.

The cloud verifies user accessing credentials but not about user information. We use a decentralized access control technique for storing data securely in clouds which allows unidentified authentication. KP-ABE and CP-ABE methods are used for developing our new decentralized access control method. In proposed solution, the cloud verifies the authentication of a user while using the path for accessing the file the unique id is generated and validated systematically. This secret key is generated while the user uploads a file into the cloud storage. The secret key for the file uploaded by the user is shared through email. This method also has the added feature of access control in which only valid users are able to decrypt the stored information.

In this paper, we concentrate on providing security and privacy for the cloud storage. There have been numerous plans of putting away information over capability of server nodes. One approach to give information strength is to recreate a data such that every storage server node stores a duplicate of the data. It is extremely cheerful on the grounds that the data can be recovered from the each storage server node survive. Another path is to encode a data of k images into a codeword of n images by deletion coding. To store a data, each of its codeword images is put away in an alternate storage server node. A storage server node disappointment relates to a deletion blunder of the codeword image. The length of the quantity of disappointment server nodes is under the flexibility limit of the deletion code, the data can be restored from the codeword images put away in the accessible storage server nodes by the sorting out procedure. This gives a trade-off between the capacity size and the resilience limit of disappointment server nodes. A decentralized eradication code is a deletion code that autonomously processes each codeword image for a data. Therefore, the encoding procedure for a data can be split into n parallel undertakings of producing codeword images. A decentralized deletion code is suitable for utilization in a disseminated storage framework. After the data images are sent to capacity server nodes, every capacity server node freely figures a codeword image for the data images and stores it. This completes the

encoding and putting away process. The restoring procedure is the same.

1.1 Purpose:

We consider the framework that comprises of appropriated storage server nodes and key-server nodes. Since putting away cryptographic keys in a solitary gadget is hazardous, a client conveys his cryptographic key to key-server nodes that should present cryptographic capacities for the benefit of the client.

These key-server nodes are exceedingly ensured by security systems. To well fit the dispersed structure of frameworks, we oblige that server nodes freely perform all operations. With this thought, we propose another edge intermediary re-encryption plan and fit in with a safe decentralized code to frame a protected conveyed storage framework. The tight combination of encoding, encryption, and sending makes the capacity framework effectively meet the necessities of information power, information classifiedness, and information sending.

1.2 scope of the proposed solution:

The main scope of this solution is to assure the users by providing security for the files stored in the cloud storage by providing a secret key for the file. This secret key is generated by using an advanced key management algorithm. This key is shared to the user through email. Our goal is to provide security and assure the users that files stored in the storage are encrypted and stored safely.

II. RELATED WORK

Providing the security to data in clouds[1], they introduce a new privacy preserving authenticated access control. In existing environment, it stores information without knowing the user identity. Our environment supports the features of access control to decrypt the stored data only by a valid user, which will supports to construct, change and read the data storing in cloud. In this environment, a user can create a file and store it securely in the cloud. This method consist the use of two protocols Attribute Based Encryption and Attribute Based Signature. The main advantage is to prevent replay attacks and supports construct, change, and understanding data stored in the cloud and Key distribution is done in a decentralized way. To protect the privacy of user attributes in the future.

In Token based computing[4], they often identify people by their attributes in the cloud. The security of the central authority and user privacy depends on the performance of the attribute authorities. This work gives a more practice-oriented attribute based encryption system.

The existing methodology is mainly focused on centralized approach. It is based on Attribute Based Encryption Technique which is a not a decentralized approach i.e., in which a single Key Distribution Centre generates secret keys and distributes to all users using asymmetric key approach.

The method uses a public key algorithm and do not maintain authentication and it provides a privacy preserving authenticated access control in cloud. A single key distribution centre is not only a single point of failure but not easy to maintain. The main disadvantage of existing system is that a user can create and stockpile a file and other user can only read the file. Write access was not allowed to user other than the creator. In existing methodology, ABS technique is also used, and this method to achieved authenticity and confidentiality.

Drawbacks:

1. The traffic between user and storage server are very high during computation and the communication.
2. Security is broken because the user has to manage his cryptographic keys.
3. It does not support the data resending scheme with secure manner.
4. A single Key Distribution Centre generates secret keys and distributes to all users using asymmetric key approach.

Decentralized access control technique stores the data by providing security in clouds and also we hide the attributes and key access rule of a user using advanced key management technique, which provides the security for the files stored in cloud by generating the secret key for each user.

III. PROPOSED WORK

We propose to use a decentralized access control technique to store data by providing security in clouds and also we hide the attributes and key access rule of a user. The cloud validates the authentication without knowing the users characteristics previous to the data store. By using this approach only authorized users have right to use the suitable attributes. By generating a secret key for each user independently, the security for the files stored in the cloud storage is increased. This is achieved by using advanced key management technique which provides the security for the files stored in the cloud storage by generating a secret key for each user.

In this paper, users are allowed to store their data (text files & images) in Drive-HQ cloud after login. After logging user (1) uploads the file or image, there after publishes that file in the server. So that a path will be generated along with a key generation which is sent to a user id. Using that path, user (1) can send the data to another user(2).

Advantages:

1. Secure data storage by using key management technique.
2. Secure data resending.
3. Generates unique secrete key for each user.
4. The traffic between user and storage server are very low during computation and the communication.

Algorithm

Key management technique which provides the security for the files stored in the cloud storage by generating a secret key for each user:

Step 1: Choose two distinct prime numbers p an$q$.

Step 2: multiply $n = p*q$

Step 3: Compute $\varphi(n) = \varphi(p)\ \varphi(q)$, where φ Euler's totient function

Step 4: select an integer e

Such that $1 < e < \varphi(n)$ and GCD $(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are co-prime.

(e is released as the public key exponent).

Determine d as $d^{-1} \equiv e \pmod{\varphi(n)}$, i.e., d is the multiplicative inverse of e (modulo $\varphi(n)$).(This is more clearly stated as solve for d given $de \equiv 1 \pmod{\varphi(n)}$).

Encryption: c is a cipher text and m is a message

$$\mathbf{C=m^e \ (mod)\ n.}$$

Decryption:

$$\mathbf{M= c^d \ (mod)\ n}$$

Given m, she can recover the original message M by reversing the padding scheme.

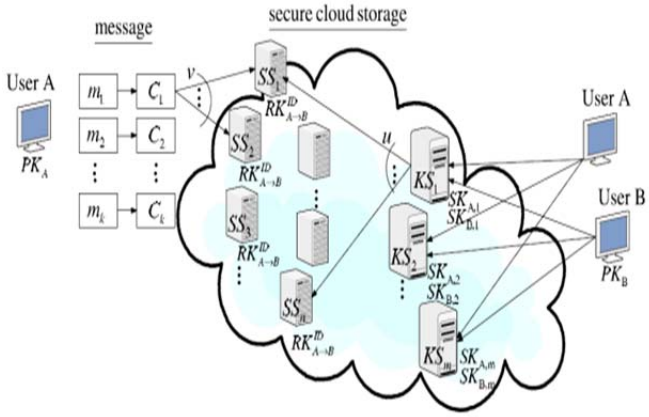

Fig 2.cloud Architecture

The above architecture represents userA sends public key to the cloud. It stores the public key and encrypts the key and generates the secrect key to userB.

IV .IMPLEMENTATION

The design and implementation of our proposed system is developed as follows:

a) Cloud data Storage
b) Data of File Resending
c) Data of File Retrieval

a) Cloud data Storage

In this method, we make set of clients, n stockpiling server nodes SS1, SS2, . . ., SSn, and m key-server nodes KS1, KS2, . . . , KSm. Capacity server nodes give stockpiling administrations and key-server nodes give key administrations. Every client is an holed out an open mystery key pair (PKA, SKA). Client An appropriates his mystery key SKA to key-server nodes such that every key server node KSi holds a key offer SKA,i, $1 < i < m$. The key is imparted to an edge t.

while information stockpiling, client A scrambles his data M and dispatches it to capacity server nodes. A data M is decayed into k squares m1;m2; . . .;mk and has an identifier ID. Client An encodes every square mi into a ciphertext Ci and sends it to v arbitrarily picked capacity server nodes. After accepting ciphertext from a client, every capacity server node straightly joins them with arbitrarily picked coefficients into a codeword image and stores it. Note that a stockpiling server node may get not as much as k data pieces and we expect that all stockpiling server nodes know the worth k ahead of time.

b)Data of file Resending

In information sending process, client An advances his encoded data with an recognizable ID put away server nodes to client B such that B can unscramble the sent data by his mystery key. To do as such, A uses his mystery key SKA and B's open key PKB to register a encryption key RKID A->B and after that sends RKID A->B to all stockpiling server nodes. Every capacity server node utilizes the re-encryption key to re-scramble its codeword image for later recovery asks for by B. The re-encoded codeword image is the mix of figure data under B's open key. Keeping in mind the end goal to recognize re-scrambled codeword images from in place ones, we call them unique codeword images and re-encoded codeword images, individual.

c)Data of file Retrieval

During information recovery, client A solicitations to recover a data from capacity server nodes. The data is either put away by him or sent to him. Client A sends a recovery solicitation to key-server nodes. After getting the recovery demand and executing a fitting verification process with client An, every key server node KSi asks for u arbitrarily picked capacity server nodes to get codeword images and does fractional unscrambling on the got secret code images by utilizing the key offer SKA,i. At last, client A joins the mostly unscrambled codeword images to acquire the first data M.

V. RESULTS

We have implemented our proposed solution using Java and we have used DriveHQ as the cloud service provider. We have represented the results of our methodology as shown in the below screenshots.

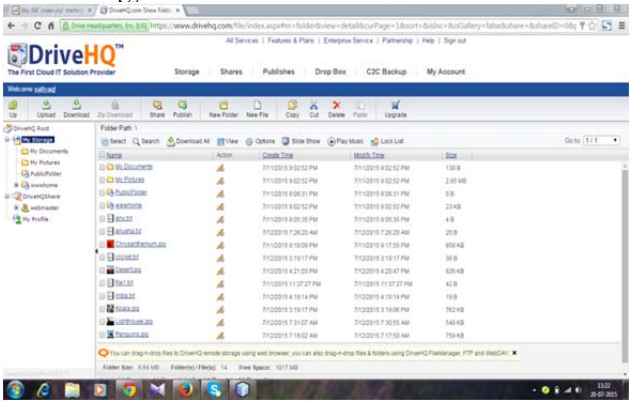

Fig.3 Upload a file and image in DriveHQ

Fig 3.To store the data into a private cloud (i.e, Drive HQ).to perform transitions like update, upload, download. we can use group communication.

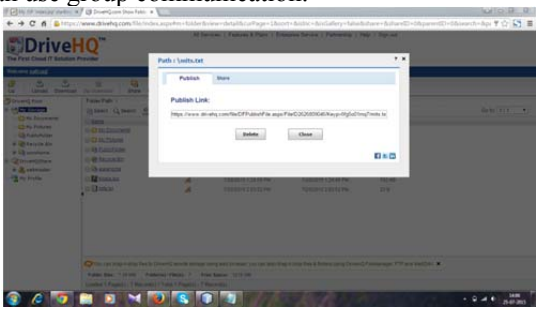

Fig.4 path can be generated

Fig 4: choosing to upload a file ,while browsing the file in local disk to select the particular file to publish in drivehq. We can generate the path.

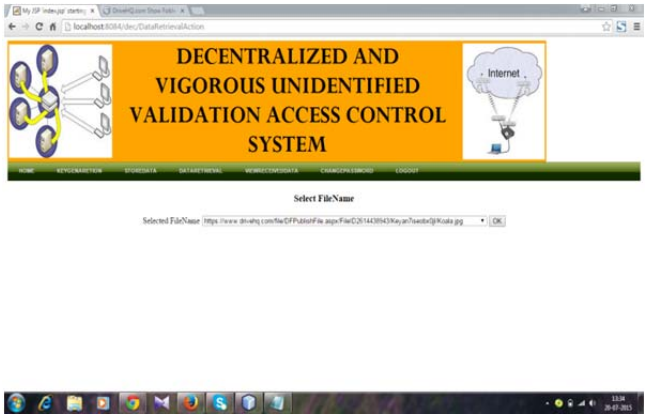

Fig.5 user will select the path

Fig5: user login into the cloud database and shares particular path which we allready published in driveHQ.

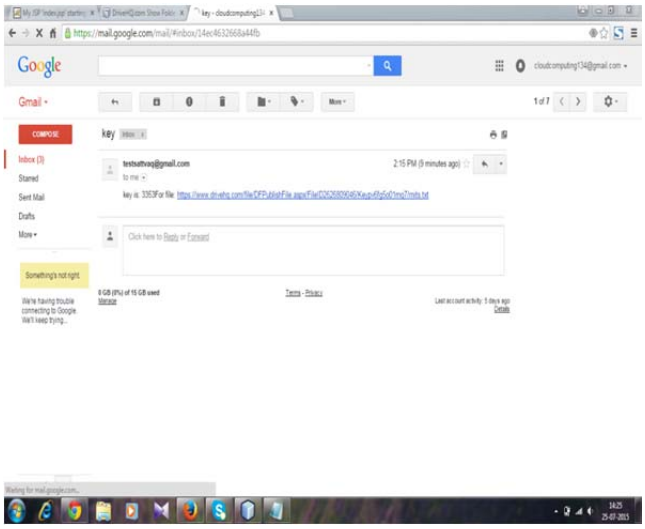

Fig.6 key is generated and sends the user mail

Fig 6: User selects the particular path of a file for which the key is generated and mailed to the user

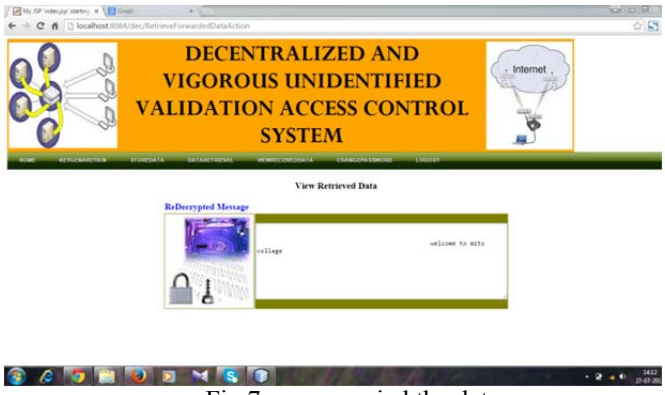

Fig.7 user recevied the data

Fig 7. Automatically the path recieves the unique key generated and user recieves the file.

User can uplaod their files in driveHQ and those files are stored and a user can publish a particular file, so that a path will be generated automatically then a user1 can login into the cloud database then select the path and send to the user2. Then user2 will have the secret key at his mail. Then they will check and a unique key is generated then user2 will receive the data.in this way we are providing the security to the data.

VI .CONCLUSION

In this paper, we provide data security in a cloud storage using decentralized approach. We use a new decentralized access control technique to store data by providing security in clouds and also we hide the attributes and key permission rule of a user. The cloud validates the authentication without knowing the users characteristics previous to the data store. By using this approach only authorized users have right to use the suitable attributes. By generating a secret key for each user independently, the security for the files stored in the cloud storage is increased. This is achieved by using advanced key management technique which provides the security for the files stored in the cloud storage by generating a secret key for each user.

REFERENCES

[1] . S. Ruj, M. Stojmenovic and A. Nayak, “Privacy Preserving Access Controlwith Authentication for Securing Data in Clouds”, *IEEE/ACM InternationalSymposium on Cluster, Cloud and Grid Computing*, pp. 556–563, 2012.

[2]. C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou, “Toward Secure and Dependable Storage Services in Cloud Computing”, *IEEE T. Services Computing*, vol. 5, no. 2, pp. 220–232, 2012.

[3]. Li, Y. Dai, L. Tian, and H. Yang, “Identity-based authentication for cloudcomputing,” in *CloudCom*, ser. Lecture Notes in Computer Science, vol.5931. Springer, pp. 157–166, 2009.

[4]. A.-R. Sadeghi, T. Schneider, and M. Winandy, “Token-based cloud computing,”in *TRUST*, ser. Lecture Notes in Computer Science, vol. 6101.Springer, pp. 417–429, 2010

[5]. V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryptionfor fine-grained access control of encrypted data,” in *ACM Conference onComputer and Communications Security*, pp. 89–98, 2006.

[6]. A. Beimel, “Secure Schemes for Secret Sharing and Key Distribution,” PhD Thesis. Technion, Haifa, 1996.