ISSN (Online): 2349-7084
GLOBAL IMPACT FACTOR 0.238
DIIF 0.876

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 531-536

Improved Self-destructing Scheme by Fine grained approach in Cloud Computing

[1]D.Farooq Basha,[2] R.Vara Prasad,[3]Dr S.Prem Kumar
[1](M.Tech), CSE
[2]Assistant Professor, Department of Computer Science and Engineering
[3]Professor & HOD, Department of computer science and engineering,
G.Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.

Abstract: Cloud computing, a recent computing technology entirely changed the IT industry. Large amount of data can be stored in cloud storage system. Security is the prime concern for this large amount of data. Without, knowledge of authorized client, data can be viewed by other user. This data contain personal information like, account number, password and notes. All the data and their copies become self-destructed after user specified time, without any user intervention. Fine grained approach is used for authorized data accessing, which provides high secured authorization for data accessing thus it achieves data confidentiality. Self-destruction method is consociated with time to live (TTL) property to specify the life time of the keys. After user specified time (TTL) data and its keys becomes destructed or unreadable. Self-destruction mechanism helps reducing overhead during upload and download process in cloud.

Keywords: Time-to-Live (TTL), Controlling, Self Destructing, Fine grained Approach

I.INTRODUCTION:

Now a days cloud computing is a rationally developed technology to store data from more than one client. Cloud computing is an environment that enables users to remotely data management. They can archive their data backups remotely to third party cloud storage providers rather than maintain data centers on their own. An individual or an organization may not require purchasing the needed storage devices. Despite they can store their data backups to the cloud and archive their data to avoid any information loss in case of hardware / software failures. Even cloud storage is more flexible, how the security and privacy are available for the outsourced data becomes a serious concern. There are three objectives to be main issue

Confidentiality –while out sourcing the data from data owner to cloud server, system need to be provide confidentiality without revealing owner details like identity and out sourcing content to third party users.

Integrity – out sourced data need to be protected from adversaries (i.e data modifications) .

Availability – Data need to be ensuring timely and reliable access to and use of information.

Recently, Sushmita ruj [1] addressed Anonymous Authentication [1] for data storing to clouds. Anonymous authentication is the process of validating the user without the details or attributes of the user. So the cloud server doesn't know the details or identity of the user, which provides privacy to the users to hide their details from other users of that cloud. Security and privacy protection in clouds are examined and experimented by many researchers. Wang et al. [16] provides storage security using Reed-Solomon erasure correcting codes. Using Homomorphic encryption, [17] the cloud receives cipher text and returns the encoded

value of the result. The user is able to decode the result, but the cloud does not know what data it has operated on. Time-based file assured deletion, which is first introduced in [5], means that files can be securely deleted and remain permanently inaccessible after a predefined duration.

Recently, Sushmita ruj [1] tended to Unknown Verification [1] for information putting away to cloud storage. Anonymous verification is the procedure of approving the client without the points of interest or properties of the client. So the cloud server doesn't know the points of interest or identity of the client, which gives security to the clients to conceal their subtle elements from different clients of that cloud. Security and privacy protection in clouds are inspected and tested by numerous scientists. Wang et al. [16] gives capacity security utilizing Reed-Solomon erasure correcting codes. Utilizing Homomorphic encryption, [17] the cloud gets figure message and returns the encoded estimation of the outcome. The client has the capacity disentangle the outcome; however the cloud does not recognize what information it has worked on. Time-based record guaranteed erasure, which is initially presented in [5], implies that documents can be safely erased and remain for all time unavailable after a predefined Length of Time.

II.RELATED WORK

Existing work on access control in cloud are centralized in nature [6], [7], [8], [9], [10], [12], [18]. Except [18] and [12], all other schemes use ABE. The scheme in [18] uses a symmetric key approach and does not support authentication. The schemes [6], [7], [10] do not support authentication as well. Security and privacy protection in clouds are being explored by many researchers. In paper [2], Wang addressed storage security using Reed-Solomon erasure-correcting codes. Authentication of users using public key cryptographic techniques has been studied in [3]. Many Homomorphic encryption techniques have been suggested [4], [5] to ensure that the cloud is not able to read the data while performing computations on them. Using Homomorphic encryption, the cloud receives ciphertext of the data and performs computations on the ciphertext and returns the encoded value of the result. The user is able to decode the result, but the cloud does not know what data it has operated on. In such circumstances, it must be possible for the user to verify that the cloud returns correct results. Author Wang, in paper [2] addressed secure and dependable cloud storage. Cloud servers prone to Byzantine failure, where a storage server can fail in arbitrary ways [2].

The cloud is also prone to data modification and server colluding attacks. In server colluding attack, the adversary can compromise storage servers, so that it can modify data files as long as they are internally consistent. To provide secure data storage, the data needs to be encrypted. However, the data is often modified and this dynamic property needs to be taken into account while designing efficient secure storage techniques. In paper [9], Zhao provides privacy preserving authenticated access control in cloud. However, the authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users. Unfortunately, a single KDC is not only a single point of failure but difficult to maintain because of the large number of users that are supported in a cloud environment. Thus, emphasis should be given on that clouds should take a decentralized approach while distributing secret keys and attributes to users. In paper [17], Yang proposed a decentralized approach; their technique does not authenticate users, who want to remain anonymous while accessing the cloud.

In a paper [10], Ruj 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 proposed system, a decentralized architecture is proposed meaning that there can be several KDCs for

key management. The main aim of paper is to design a scheme for distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.

III.PRESENTED SYSTEM:

In our presented system Cloud and its services are highly influenced by people and organization. Many of are migrating to cloud for their data or related applications. Storage service is widely deploying service of cloud. So that, cloud user feels free to store and share huge no of files. Unfortunately, most of them never think about those files after sharing. Since the shared files remains in cloud for long period of time, it raises security and privacy issues in cloud groups. The shared files may include sensitive information which may misused by any miscreant or even service providers. Another issue is that dumping of huge no of files in cloud consumes more storage space and reduces search efficiency of the system. To resolve these issues we proposed a self-destruction system that automatically removes shared files after certain time period specified by its owner. The following sessions describes design and implementation details of our scheme in detail.

Key management: with our Presented System Key management is a challenging issue where sharing and storing keys in order to provide the data security.

Authentication: here with our presented system there is no proper access controlling scheme performed while out sourcing the data from data owner to Cloud Server or from Cloud Server to end users while data accessing due to performing by Coarse grained approach.

Key Distribution Center:

Here KDC emphasize that clouds should take a centralized method while allocating secret keys among the users. It is somewhat difficult for clouds to have a single KDC to issues keys for different locations in the world. The architecture is centralized; meaning that there can be single KDCs for key management in this regards system performance is very poor due to limited handling capacity.

Data integrity: When the system is failure in providing data confidentiality and security due to weak cryptosystem then became to loosing data integrity, so here lack of data integrity due above reasons.

IV.WORKING MODEL OF PROPOSED SYSTEM

In order to address the above issues, our proposed system performs a secure data transaction in the cloud; the suitable cryptographic method is used .i.e. RSA algorithm. The owner must encrypt the file with some specified attributes, with owner's private key which was generated by the KDC operated by the Trustee.

Setup Phase: in this phase data owner can obtain Private Key from KDC ,get his public key and get Time interval tag from Time server for data availability and collect all this things as attribute set and apply RSA algorithm to encrypt the data be out sourcing to Cloud server.

Encrypt: in this phase data will encrypted along with attribute set, which consist of E(M,Pk,T,Puk)-→RSA→CT, where M: Message,Pk: Private Key which is generated by KDC,T: Time Interval ,Puk: Public Key

Decrypt: in this phase data will be decrypted along with attribute set, which consist of D(CT)→RSA→ M,Pk,T,Puk.

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 9, SEPTEMBER 2015, PP 531-536

System Architecture:

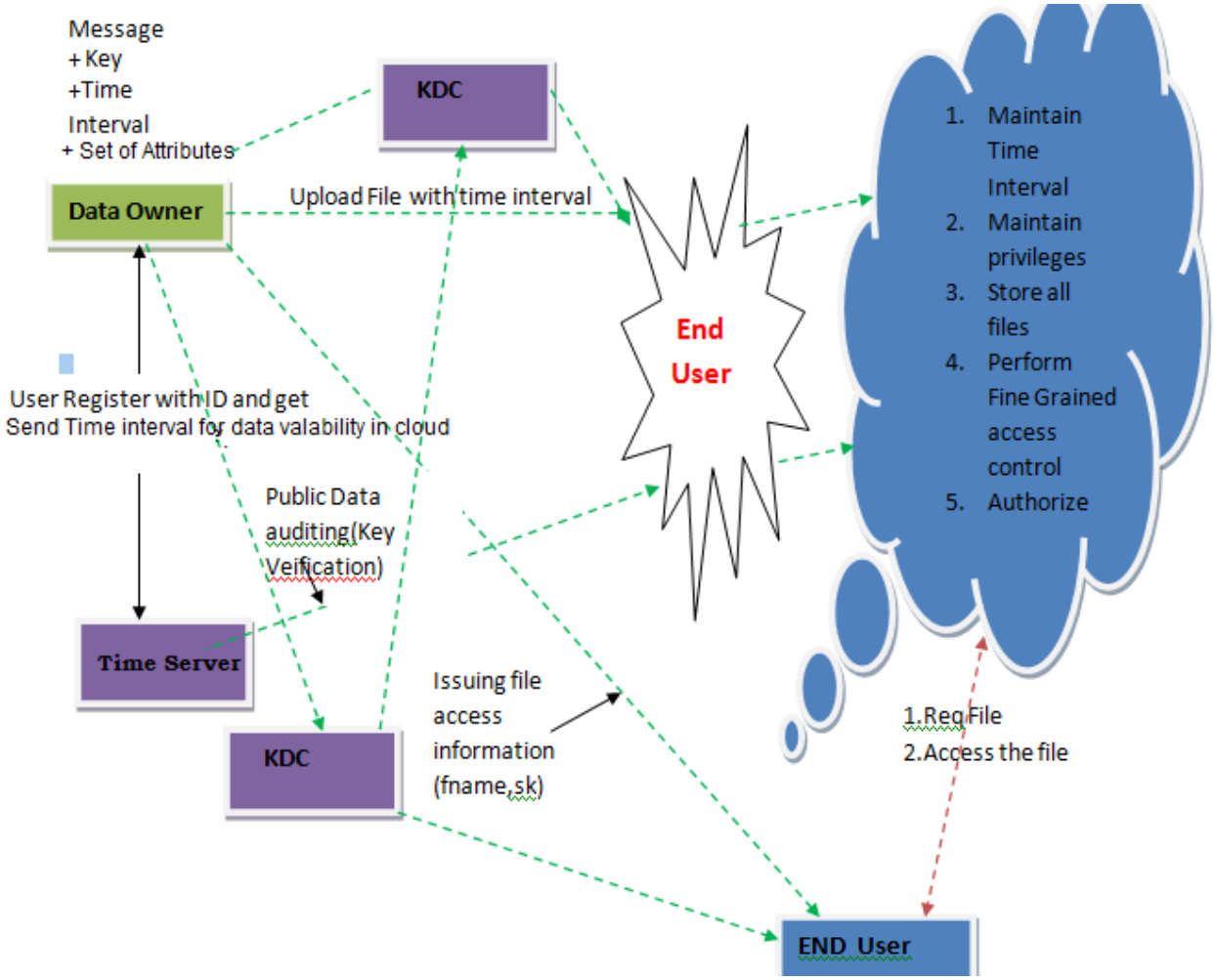

Fig 1.System Architecture

Before to outsource the data into cloud server data owner append a time interval tag which was issued by the time server which will be used as a time stamp. Finally Owner can upload encrypted data into cloud server with Time intervals. If a third person want to access that file remotely from cloud server, user need be authorized by the cloud server i.e. here fine grained approach will be performed at cloud level soon after authorized by cloud server, cloud server send encrypted content to user, now user need get Decrypted keys that is Private key and Public Key by the Trustee it will done based on user identity. Users may view the record if the user had the key which is used to decrypt the encrypted file . Sometimes this may be a failure due to the technology development and the hackers. The key distribution center is a server that is responsible for cryptographic key management. The public key is

time-based, it means if key will be deleted or removed by the key manager when an expiration time is reached, where the expiration time is specified when the file is first declared or uploaded . Without the public key, the private key and hence the data file remain encrypted and are deemed to be inaccessible. Thus, the main security property of file assured deletion is that even if a cloud provider does not remove expired file copies from its storage, those files remain encrypted and unrecoverable. We propose a policy based file access [6] and policy based file assured deletion [6], [7], [8] for better access to the files and delete the files which are decided no more.

Our system also has the added feature of fine grained access control in which only valid users are able to decrypt the loading information. The system prevents replay attacks and supports creation, modification, and reading data collected in the cloud.

ADVANTAGES:

Distributed access control of data collected in cloud so that only certified users with fully valid attributes can read them. The confirmation of users who collection and modify their data on the cloud. The identity of the user is secure from the cloud during confirmation.

V.CONCLUSSION:

In this paper we presented our proposed system has the added feature of fine grained access control in which only valid users are able to decrypt the loading information among cloud system. The system prevents replay attacks and supports creation, modification, and reading data collected in the cloud. Finally our system proves that high security, secure authentication, better in performance with data integrity.

REFERENCE:

[1] S Sushmita Ruj, Milos Stojmenovic and Amiya Nayak, "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS

[2] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, Apr.- June 2012.

[3] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Computing," Proc. First Int'l Conf. Cloud Computing, 2009.

[4] C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., 2009.

[5] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-Based Cloud Computing," Proc. Third Int'l Conf. Trust and Trustworthy Computing (TRUST), 2010.

[6] M. Li, S. Yu, K. Ren, and W. Lou, "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings," Proc. Sixth Int'l ICST Conf. Security and Privacy in Comm. Networks (SecureComm), 2010.

[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), 2010.

[8] G. Wang, Q. Liu, and J. Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services," Proc. 17th ACM Conf. Computer and Comm. Security (CCS), 2010.

[9] F. Zhao, T. Nishide, and K. Sakurai, "Realizing Fine-Grained and Flexible Access Control to Outsourced Data with Attribute-Based Cryptosystems," Proc. Seventh Int'l Conf.

Information Security Practice and Experience (ISPEC), 2011.

[10] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," Proc. IEEE 10th Int'l Conf. Trust, Security and Privacy in Computing and Communications (TrustCom), 2011.

[11] S.SeenuIropia, R.Vijayalakshmi, "Decentralized Access Control Of Data Stored In Clouds Using Key Policy Attribute Based Encryption", International Journal Of Invention In Computer Science And Engineering, 2014.

[12] http://securesoftwaredev.com/2012/08/20/xacml-in-the-cloud, 2013.

[13] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), 2001.

[14] X. Boyen, "Mesh Signatures," Proc. 26th Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), 2007. [15] D. Chaum and E.V. Heyst, "Group Signatures," Proc. Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), 1991.

[16] H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance," IACR Cryptology ePrint Archive, 2008.

[17] K. Yang, X. Jia, and K. Ren, "DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems," IACR Cryptology ePrint Archive, 2012.

[18] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and Efficient Access to Outsourced Data," Proc. ACM Cloud Computing Security Workshop (CCSW), 2009.