Computer Science and Engineering 2016, 6(1): 1-6
DOI: 10.5923/j.computer.20160601.01

Decentralized Access Control Schemes for Data Storage on Cloud

Shraddha V. Mokle*, **Nuzhat F. Shaikh**

Department of Computer Engineering, Modern Education Society's College of Engineering, Pune, India

Abstract In this review paper of decentralized access control schemes are reviewed for secure information storage on cloud. In centralized network storage, data is not secure, data accessing for different user is difficult, updating data is also not that easy, to overcome all these issues decentralized schemes are considered. A decentralized access control scheme with anonymous authentication provides user revocation and prevents replay attacks. The cloud does not know the identity of the user who stores information, but only verifies the user's credentials. Fine-grained access control scheme grants different access control policies to user to do operations on their data. Key distribution is done in a decentralized way. Data stored in clouds is highly sensitive. These schemes gives secure data storage in decentralized network compared to centralized network.

Keywords Fine-grained Approach, Access control, Authentication, Key Distribution Center, Attribute-based signatures, Cloud storage

1. Introduction

Research in cloud computing is reached to its best & receiving a lot of attention from both academic and industrial worlds. Cloud computing is basically for users who can outsource their computation and stores data/information to cloud using Internet. Cloud services like applications (e.g. GoogleApps), infrastructures (e.g Eucalyptus, Amazon's EC2, Nimbus), platforms for developers to write applications (e.g., Amazon's S3, Windows Azure). Much of the data stored in clouds is highly sensitive and which requires security, for example, social networks and medical records. Security and privacy are thus very important and critical issues in cloud computing. In one hand important thing is, the user should authenticate itself before initiating any transaction, and on other hand, it must be ensured that the cloud or other users do not know the identity of the user. The cloud can hold the user who uses cloud for storage with respective to their application to outsources data, and likewise, the cloud itself accountable for the services it provides. The validity of the user who stores the data is also verified. There is need of law enforcement other than technical solutions to ensure other than privacy and security [1].

Cloud access control is gaining attention because it is important that only authorized users have access to valid service. A huge amount of information is being stored on cloud servers, and much of stored data is sensitive information. Sensitive information is online social networking where users (members) store their pictures, videos and personal information and share them with selected groups of users or communities. It is not just enough to store the contents or information securely in the cloud but also it's necessary to ensure invisibility of the user. For example, a user would like to store some sensitive information but does not want to be recognized, user need security for their information that will be stored anywhere. The user might want to post a comment on picture, article but does not want his/her identification to be disclosed. However, the user should be able to demonstrate to the other users that he/she is a valid user who stored the information without revealing the identification or identity.

Existing work on access control in cloud are amalgamated or centralized in nature. Even if some fragmented or decentralized approaches were existing does not support authentication for user. Earlier work provides privacy preserving validated or authenticated access control in cloud. However, the author take amalgamated or centralized approach where single key distribution center (KDC) distributes secret keys and attributes to all users.

2. Literature Survey

Access control of information/data in clouds is centralized in nature. All schemes use ABE or symmetric key approach and does not support user authentication. Earlier work provides privacy preserving authenticated access control in

* Corresponding author:
mokle.shraddha@gmail.com (Shraddha V. Mokle)
Published online at http://journal.sapub.org/computer

cloud.

A decentralized approach is proposed by different others in existing papers, their technique does not authenticate users for data access, who want to remain anonymous while accessing data. Previous work has proposed distributed access control mechanism in cloud. However, the scheme does not provide authentication for users. Other important thing was that only creator can write that stored file other users can not able to write that file which was stored on cloud. In earlier work write access was given to only creator or owner of respective file not to the readers this was the drawback [2].

Cloud servers are liable to suffer from Byzantine failure, where a storage server can fail in arbitrary ways [3]. Author Proposes Authority to revoke user attributes with minimal effort [3]. The cloud is also liable to suffer from data modification and server colluding attacks. In server colluding attack, the adversary can compromise storage servers, so that it can change or modify data files as long as they are internally consistent. Data encryption is needed to provide secure data storage on cloud. However, the data is often modified and this dynamic property needs to be taken into consideration while designing efficient secure storage techniques. Efficient search on encrypted data is also an important deal in clouds. The clouds should not know the processing of data i.e. query but should be able to return the records that satisfy the query. This is achieved by means of searchable encryption.

Key Distribution Center (KDC) is centralized approach where a single KDC distributes secret keys and its attributes to all users that are present. Single KDC is single point of failure, with single secret key failure whole system can collapse. KDC is difficult to maintain because of large number of users that are present in the cloud for information sharing or storage. Therefore, this emphasize that clouds should take decentralized approach for distributing secret keys. Now days it's quite natural that clouds have many KDCs in different remote places in the network [4].

ABE scheme which is proposed in [5] has set of attributes defined with Unique Id. Two classes has been defined in the ABEs ABE or KP-ABE, sender has an access policy to encrypt data. In key policy ABE or KP-ABE [6] the sender has an access policy to encrypt data. A writer whose attributes and keys have been revoked cannot write back stale information. The receiver receives attributes and secret keys from the attribute authority and is able to decrypt information if it has matching attributes. In Cipher text-policy, CP-ABE, the receiver has the access policy presented in form of a tree, with attributes as leaves and monotonic access structure with AND, OR and other threshold gates [6] [7].

All the approaches take a centralized approach and allow only one KDC, which is a single point of failure [9]. Chase proposed a multi authority ABE, in which there are several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys to users. Multi authority ABE protocol was studied in, which required no trusted authority which requires every user to have attributes from at all the KDCs. Recently, proposed work fully, decentralized ABE where users could have zero or more attributes from each authority and did not require a trusted server [9]. In all these cases, decryption at user's end is computation intensive. So, this technique might be inefficient when users access using their mobile devices. To get over this problem, work proposed to outsource the decryption task to a proxy server, so that the user can compete with minimum resources (for example, hand held devices). However, the presence of one proxy and one KDC makes it less robust than decentralized approaches. Both these approaches had no way to authenticate users, anonymously. A modification of authenticate users, who want to remain anonymous while accessing the cloud [5]. A recently different scheme works on decentralized approach and provides authentication without disclosing the identity of the users. However, as mentioned in the previous section it is prone to replay attacks.

3. Analysis of Access Control Schemes

3.1. Overview

In cloud computing data access authentication is very important concern. Different papers proposed authentication for data stored on cloud server, data access polices and how the key distributed amongst creator, writer and reader. This paper, analyze existing work for cloud data storage in decentralized network.

In existing work related to access key policies and data storage, entire operation i.e. read or write was performed on centralized in nature scheme, it uses AES Scheme which is known for single key manner it seems less secure and high performance comparatively it's a somewhat faster than asymmetric scheme. For uploading data on cloud servers, owner takes a centralized method where single KDC (Key Distribution Center) allocates keys with access controlling attributes which are distributed in different users when there is huge number of request from more than one user at same time, there may be chances to crash because of server overload and the server may go down and key management for multiuser through single KDC is one of the challenging issue.

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

Authentication: In earlier existing 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: KDC emphasize that clouds should take a centralized method while providing or 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 because single KDC might not sufficient for users. Centralized architecture meaning that there can be single KDCs for key management, inefficient due to limited handling capacity and system performance is very poor.

Data integrity: When the system falling in providing data confidentiality for users and security due to weak cryptosystem then became to loosing data integrity, because this data integrity affects.

3.2. Existing Schemes

3.2.1. Fine-Grained Access Control Scheme

Fine-grained Access Control Networked storage systems provide cloud storage services for users over networks [6]. To ensure data confidentiality in secure networked cloud storage systems data stored into encrypted form. Here existing system Fine-grained access control facilitate granting differential access rights to a set of users and allow exibility in specifying the access rights of individual users. There are several techniques which are known for implementing fine grained access control.

Existing system gives high security for that which performs a secure data transaction in the cloud; the suitable cryptographic method is used .i.e. RSA algorithm. The creator must encrypt the file with some specified attributes, with creator's private key which was generated by the KDC operated by the Trustee. Fig 1. Shows fine- grained access control of data on cloud server [12].

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

Encrypt: In this phase data encrypted along with attribute set, which consist of

$$E(M,PK,T,PuK)\rightarrow RSA\rightarrow CT,$$

where terms defined in existing system

M: Message,

Pk: Private Key which is obtained by KDC,

T: Time Interval,

Puk: Public Key

Decrypt: In this phase data decrypted along with Attribute set, which consist of

$$D(CT)\rightarrow RSA\rightarrow M,PK,Puk.$$

Before to outsourcing the data to cloud server data owner gives time interval tag which was issued by the time server and which will be used as a time stamp. Finally Owner/creator can able to upload encrypted data/information into cloud server with Time intervals. In case of other user who wants to access that file from cloud server, user need be authorized by the cloud server i.e. fine grained approach will be performed at cloud after authorized by cloud server, cloud server respond to request and send encrypted content to user, now user need get Decrypted keys that is Private key and Public Key by the Trustee of system 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 [1]. Sometimes this may be a failure due to the technology development and the hackers. Mainly key distribution center is a server that is responsible for cryptographic key management. In fine grained approach expiration time for public key is specified when the file is first declared or uploaded on cloud server, hence public key is time based key which will be deleted or removed by key manager when an expiration time reached.

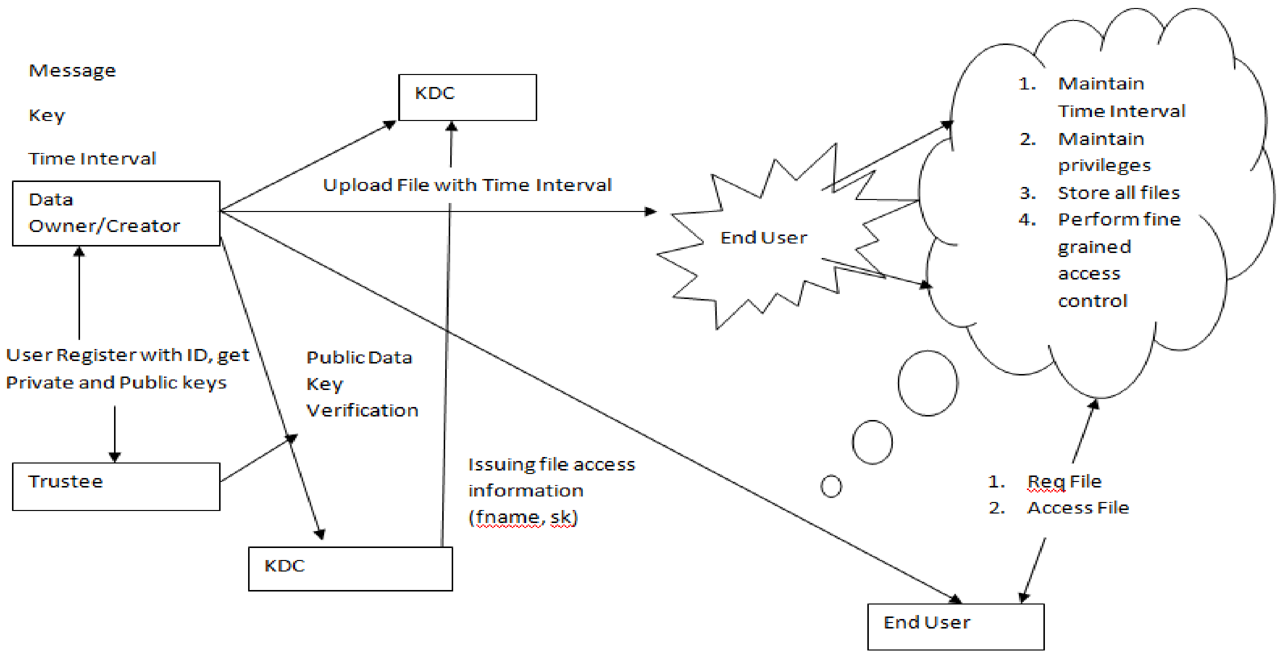
Message
Key
Time Interval
Data Owner/Creator
KDC
Upload File with Time Interval
End User
1. Maintain Time Interval
2. Maintain privileges
3. Store all files
4. Perform fine grained access control
User Register with ID, get Private and Public keys
Public Data Key Verification
Trustee
Issuing file access information (fname, sk)
KDC
1. Req File
2. Access File
End User

Figure 1. Fine-grained Access Control

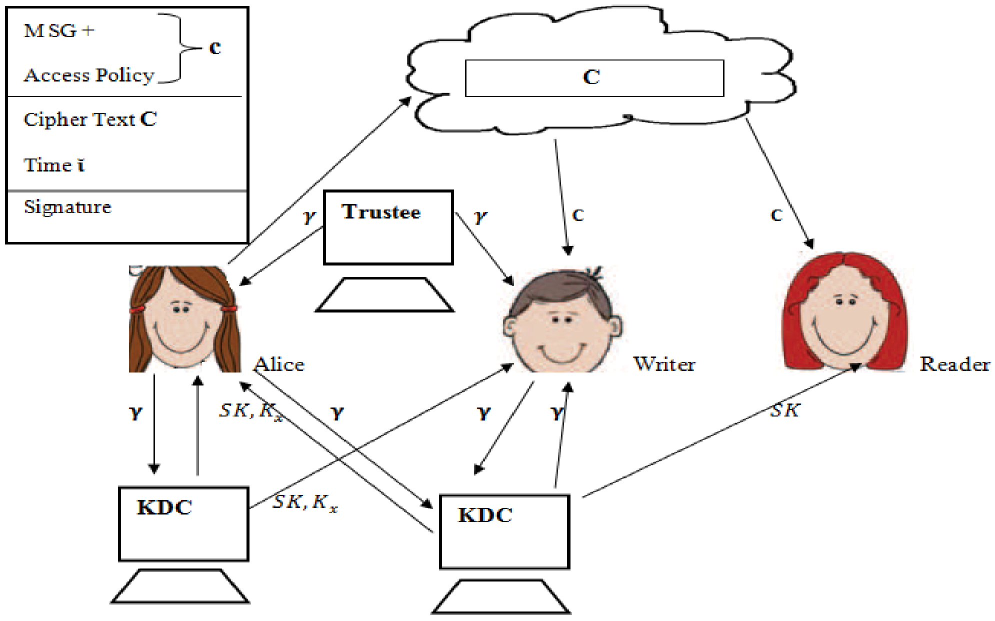

Figure 2. Secure Cloud

3.2.2. Authenticated Access Control Scheme

The system consists of three users' owner or creator, writer and reader. Creator will create a file and upload it to cloud server. Here creator will receive a token from trustee and trustee is the federal government, which manages social insurance numbers. The owner will send the id to the trustee then receives token γ from trustee. Here τ is time stamp is used to intercept write old information to cloud when the user is revoked. The creator will then send the token γ to Key Distribution Centre and there are various KDC available in different regions of world. After sending token to KDC creator will receive Encryption, Decryption keys and signing keys. In Fig 2 SK Is a secret key and Kx are signing keys. Message which is going to store on cloud is encrypted using access policy X and it decides who have the right to access or modify the data stored in the cloud. The Claim Policy y is used to confirm authenticity and message is signed under this claim policy. Along with the signature, Ciphertext i.e. c, C respectively is sent to cloud. The sent signature is verified by cloud and accordingly stores the Ciphertext. The Ciphertext C is sent to the reader when reader wants to read the data in cloud. If the other than creator has access policy with matching attributes then the reader can decrypt and read the message. For writer operation takes place as file modification, making. The user sends the message with claim policy and it is authenticated by cloud if the user is authenticated then that user is permitted to write to existing file.

3.2.2.1. Data Storage in Clouds

A user Uμ need to first registers itself with one or more trustees that are available. For simplicity assume there is one trustee. The KDCs are given keys ASK[i], APK[i] for signing/verifying and PK[i], SK[i] for encryption/ decryption. The user on presenting this token obtains secret keys attribute sets from one or more KDCs. Key for an attribute x belonging to KDC Ai. The user also receives secret keys $sk_{x,u}$ for encrypting stored messages. For accessing message that will be stored on cloud policy created by user that is access policy x which is a uniformity Boolean function. The message is then encrypted under the access policy.

The end user also constructs a claim policy Y to enable the cloud to authenticate the user. The creator/owner does not send the message MSG as it is, but uses the time stamp and creates $H(C) \parallel \tau$ this is done to prevent replay attacks. If the time stamp is not sent, then the respective user can write previous old message back to the cloud server with an authorized signature, even when its attributes and claim policy have been revoked. The original work suffers from replay attacks. A writer can send its message and correct signature even when it no longer has access rights [11]. In existing scheme a writer's rights have been revoked cannot create a new signature with new time stamp and cannot write back old information. After that it signs the message and calculates the message signature.

3.2.2.2. Modifying Data to the Cloud

For writing to an already existing file, the user must send its message with the claim policy as done during file creation. Verification of the claim policy is done by cloud, and only if the user is authorized, is allowed to write on the file.

3.2.2.3. User Revocation

Existing work defined how prevents replay attacks. Existing work gives thought about how to handle user revocation. Work should ensure that users must not have the ability to access data, even if they possess matching set of attributes to prevent data. For this reason, the owners should change the stored data and send updated data to other users. Set of attributes Iμ possessed by the revoked user Uμ is noted and users change their stored information that has attributes $i \epsilon I_{\mu}$. In [10], revocation involved changing the public and secret keys of the minimal set of attributes defined by creator which are required to decrypt the data. This approach is not considered because here different data are encrypted by the same set of attributes (it should not encrypt data by same attributes for different users), so different user have different minimal set of attributes. Therefore, this does not apply to model which is already there for anonymous authentication of data. Once the attributes I_{μ} are identified, all data that possess the attributes are collected. So that without the public key, the private key and hence the data file remain encrypted and are deemed to be unreachable. Thus, the main security property of file is deletion provided that even if a cloud provider maintains expired file copies in its storage, it does not remove it and those files remain encrypted, unrecoverable [1]. Policies based file assured deletion [1], [10] for better access to the files and delete the files which are decided no more. Developed system also has the added new feature of fine grained access control in which only authorized users are able to decrypt the loading information. The system prevents replay attacks and supports construction, modification, and reading data collected in the cloud.

4. Security of the Protocol

Existing security of cloud storage protocols are:

Theorem1. Existing protocol is secure (no outsider or cloud can decrypt ciphertexts), collusion resistant, allows access only to authorized end users [13].

Proof. System showed that no unauthorized user can access data from the cloud. Existing system first proved the validity of scheme. A user can decrypt data if and only if it has a matching set of attributes as per defined. This follows from the fact that access structure S (and hence matrix R) is formed if and only if there exists a set of rows X in R, and linear constants. System next observes that the cloud is not able to decode stored data. This is because it does not possess the secret keys sk_i, u. Even if it colludes with other users, it cannot decode data which the users cannot themselves decrypt, because of the above reason (same as collusion of users). The KDCs are not owned by clouds and are located in different servers. For this reason, even if some (but not all) KDCs are compromised, the cloud server cannot decode data.

Theorem 2. System's authentication scheme is precise, conspiracy secure, resistant to replay attacks, and protects privacy of the user [13].

Proof: Only valid users registered with the trustee(s) and receive attributes, keys from the KDCs. A user's token is K_{base}, K_0 where signature on μ, K_{base}, K_0 with TSig belonging to the trustee. An invalid user with a different user-id not allowed creating the same signature because it does not know TSig.

5. Conclusions

In decentralized access every system has the access control of data. We reviewed that cloud servers are secured storage where the anonymous authentication is used for users, so that only the permitted users can access or modify data. Decryption of data can be viewed only by authorized users. It supports prevention of replay attack, creation of data/information, modifying the data by unknown users, and reading data stored in Cloud. The accessing Cloud data for authorized users in decentralized network is very useful and robust there for overall communication storage has been developed by comparing to the Centralized approaches. In this review paper decentralized access schemes gives authentication to creator with KDC, more secure data storage n cloud companied with the centralized scheme.

REFERENCES

[1] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Computing," Proc. First Int'l Conf. Cloud Computing (CloudCom), pp. 157-166, 2009.

[2] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in *IEEE Trust Com*, 2011.

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

[4] S. Seenu Iropia and R. Vijayalakshmi (2014), "Decentralized Access Control of Data Stored in Cloud using Key-Policy Attribute Based Encryption" in preceedings: International journal of Inventions in Computer Science and Engineering ISSN (print):2348-3431.

[5] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc. Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), pp. 457-473, 2005.

[6] 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. Secu-rity (CCS), 2010.

[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy, pp. 321-334, 2007.

[8] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp 343-352, 2009.

[9] M. Chase, "Multi-Authority Attribute Based Encryption," Proc. Fourth Conf. Theory of Cryptography (TCC), pp. 515-534, 2007.

[10] S. Yu, C. Wang, K. Ren, and W. Lou, Attribute Based Data Sharing with Attribute Revocation, || Proc. ACM Symp. Infor-mation, Computer and Comm. Security (ASIACCS), 2010.

[11] H. K. Maji, M. Prabhakaran, and M. Rosulek, “Attribute-based signatures: Achieving attribute-privacy and collusion resistance,” IACR Cryptology ePrint Archive, 2008.

[12] 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.

[13] 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.