USER ANONYMOUS AUTHENTICATION SCHEME FOR DECENTRALIZED ACCESS CONTROL IN CLOUDS USING MULTIPLE KDC

KOMAL L. SAWANT[1], TRUPTI G. GAYKAR[2], PRAGATI T. RODE[3], ASHWAJIT S. DANGRE[4]

[1] *Computer Engineering, SGOI COE, Maharashtra, India*
[2] *Computer Engineering, SGOI COE, Maharashtra, India*
[3] *Computer Engineering, SGOI COE, Maharashtra, India*
[4] *Computer Engineering, SGOI COE, Maharashtra, India*

ABSTRACT

Abstract- *Now a days, the cloud computing concept is developing rapidly and taking considerable attention. It is an effective way for reducing energy costs, so efficiency of the data centers can be improved. In this paper we propose securing data on cloud using new decentralized access which supports anonymous authentication. In proposed scheme cloud verifies authenticity of the user and grants access to an valid user through who can get accessed to the data information stored in cloud and also encrypt or decrypt the data information stored in the cloud. For which we propose new feature based on KDC (Key Distribution center) which supports creation, modification, reading and writing of data stored on cloud and also prevent replay attack.*

Keyword: *Access control, Authentication, Attribute-based signatures, Attribute-based encryption and Cloud storage.*

1. INTRODUCTION

In Cloud computing is big source of large-scale distributed computing. It has moved computing and data away from desktop and PCs, into larger data center the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand of user. This can work for allocating resources to users. Figure 1.1 shows the cloud. Cloud computing provides the different types of services that are based in pay-as-go model. User can take services on cloud ranging from web application to scientific application. The services are delivered over internet to the customer or user. Cloud consists of servers and number of resources. Clouds can provide several types of services like applications (e.g., Google Apps, Microsoft online), infra-structures (e.g., Amazon’s EC2, Eucalyptus, Nimbus), and platforms to help developers write applications (e.g. Amazon’s S3, Windows Azure).Data stored on cloud is highly sensitive like medical record & social network

.Security Privacy are import issues in cloud computing .Indeed cloud storage is more adaptable how the security & protection are accessible for the outsourced data turned into real concern .The three major issues are availability ,confidentiality and integrity.

1.1 User's anonymity in cloud computing

User privacy is also required in cloud that means other users do not know the identity of others users which can be achieved by maintaining privacy or anonymity of user's .The validity of users who stored the data on cloud is also verified. It is also important to verify the information comes from a reliable source

Fig 1.A Cloud

1.2 Access control

There are three types of access control: user-based access control (UBAC) , role-based access control (RBAC) , and attribute-based access control (ABAC) .In UBAC , the access control list contains the list of authorized users has access to data but this is not possible in cloud because there are many users so that it is difficult to maintain list of authorized users. In RBAC users are classified according to their own roles which are declared by system. ABAC is more extended in scope, in which users are given attributes, and access policy. Only users with valid set of attribute and satisfying the access policy, can access the data. And have decrypting the information stored in cloud.

1.3 Encryption in cloud computing

The cloud is also prone to data eversion and server colluding attacks. The averse can compromise storage servers in server colluding attack, so the server can modify data files even though the servers are internally coherent. The data needs to be encrypted to provide secure data storage. However, the data is often modified and this dynamic property Needs to be taken into account while designing useful secure storage techniques.

1.4 Security and privacy protection on cloud data

Public key cryptographic techniques is used to Authenticate user in cloud computing. Many holomorphic Encryption techniques have been optional sure that the cloud is not able to read the data while performing computations on the data. By using this encryption scheme, the cloud receives cipher text of the data .On received cipher text computations are performed and then it returns the encoded value to user then the user is able to decode the result or encoded value, although the cloud does not even know what data it has operated on.

2. LITERATURE SERVEY

Attribute based encryption was proposed by Sahai and Waters [25]. In ABE, set of attributes are given to user and its associated unique ID. There are two subtype of ABEs. In Key-policy ABE or KPABE (Goyal et al.) for encrypting data the access policy is given to sender. If anyone want to write a data but his/her attributes and keys have been revoked so they can't write back stale information. From the attribute authority receiver can get attributes and secret keys and so that receiver can decrypt the information if and only if they have matching set of attributes. In Cipher text-policy, CP-ABE ([24], [23]), the access policy is given to receiver in the form of a tree, this attributes are treated as leaves and monotonic access structure with AND, OR gates.

All this attend to take a centralized approach and allow only one KDC, which is a single point of failure. Chase [22] proposed a multi-authority ABE, as the name suggest there are multiple KDCs are used (coordinated by a trusted authority) this multiple KDCs distribute attributes as well as secret keys to users. Multi-authority ABE protocol was studied in [10], which required no trusted authority which requires every user to have attributes from at all the KDCs. Recently, A fully decentralized ABE was proposed by Lewko and Waters [12] proposed where users could have zero or more attributes from each authority It did not need a trusted server. 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, Green et al. [13] proposed to outsource the decryption task. In this proxy server are used for decryption, so that the user can compute with minimum resources (for example, hand held devices like smart phones). In which, the presence of one proxy and one key distribution center makes it less robust and less secure than decentralized approaches. In this both approaches had no way to authenticate anonymously users. Yang et al. [10] presented a modification of authenticate users, who want to remain anonymous while accessing the cloud.

To assure anonymous user authentication Attribute Based Signatures were introduced by Maji et al. [21].In this also centralized approach are used. A recent scheme by the same authors takes a decentralized approach and provides authentication without discovering the identity of the users. Milos Stojmenovi et al [6] proposed a decentralized access control and also provides user anonymity but cloud knows the access policy stored for each record which is stored on cloud.

3. PROPOSED SYSTEM

Our contributions in this paper are the following:

- In decentralized access control of data stored in cloud that means only authenticated users with valid attributes set are allowed to access data on cloud.

- The identity of the user is hidden from the cloud during authentication. Decentralized access means there can be multiple KDCs for key management task.

- The access control and authentication are both Collusion resistant, if there are two users and they are not individually authorized cannot collude and access data or authenticate themselves.

In this users are rejected when not having matching set of attributes and key or not have access to data stored in cloud .The revoked or rejected users cannot access data after they have been revoked. The proposed scheme is resilient to replay attacks. A writer has been revoked when he/she does not have matching attributes and keys and cannot write back stale information. These supports multiple read and write on the data stored in the cloud. The costs are comparable to the existing centralized approaches, and the expensive operations are mostly done by the cloud**.** Cloud does not know the access policy for each record stored in cloud using cloud handler which stores access policy and also does encryption and decryption.

The architecture is decentralized, which means there can be number of KDC's for key management. In KDC's database article id and corresponding key's has been stored. User Registration information and particular article id has been stored in cloud handler database. Every user's need to register and login themselves. Suppose there are two users, user1 is a publisher and user2 is reader or writer. Consider user1 is publishing the article so that user1 become a publisher of that article. When user1 publish his article the article id has been generated for that article and that article id has been send to the KDC's. for that id the KDC generate read key and write key. Using read key article is being encrypted. Encrypted article is uploaded successfully to the cloud server using cloud handler. Now consider another user, say user2 who wants to read or write the articles published by another users. so that user2 needs to send request to cloud handler for list of article .

Before sending list of article to users the privilege tree has been checked and according to privilege tree the article list is sent to user2. After that user choose the article from article list to read or write . The particular article id is sent to the cloud handler and KDCs as well.

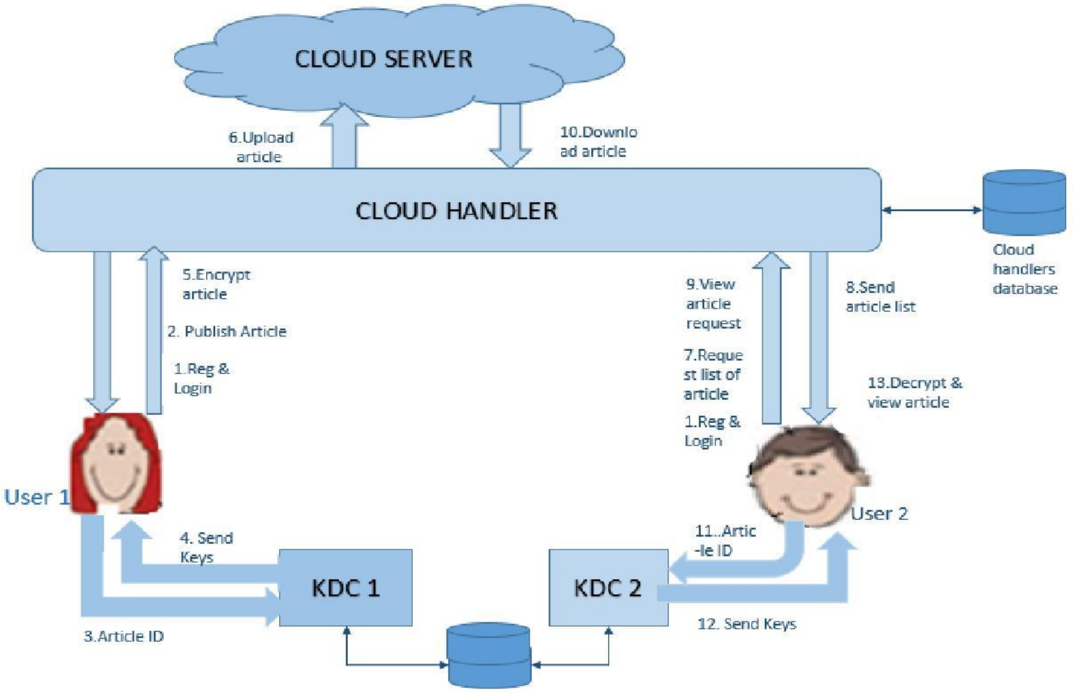


Fig.2. System Architecture

Cloud handler checks the article id and download particular article from cloud server which is in encrypted format. To decrypted article KDC provides corresponding read key and using the read key the article is decrypted then original article is shown to the user2. If user want to update or write that article the KDC will provides corresponding write key so user can write an article successfully.

3.1 User Accessibility

Every user's need to register and login themselves. When user publish his article the article id has been generated for that article and that article id is send to the KDC's. for that id the KDC generates read key and write key. Using read key article is being encrypted. Encrypted article is uploaded successfully to the cloud server using cloud handler. Now consider another user, who wants to read or write the articles published by another users. so that user needs to send request to cloud handler for list of article .Before sending list of article to users the privilege tree has been checked and according to privilege tree the article list is sent to user. After that user choose the article from article list to read or write . The particular article id is sent to the cloud handler and KDCs as well.

Cloud handler checks the article id and download particular article from cloud server which is in encrypted format. To decrypted article KDC provides corresponding read key and using the read key the article is decrypted then original article is shown to the user. If user want to update or write that article the KDC will provides corresponding write key so user can write an article successfully.

3.2 Authentication using KDC

Then KDC given a user id to a user, the user will Enrolled the personal details to KDC's that given an Input as user name,user id, and password etc. The KDC will be verify the user details and it will insert it in a Database

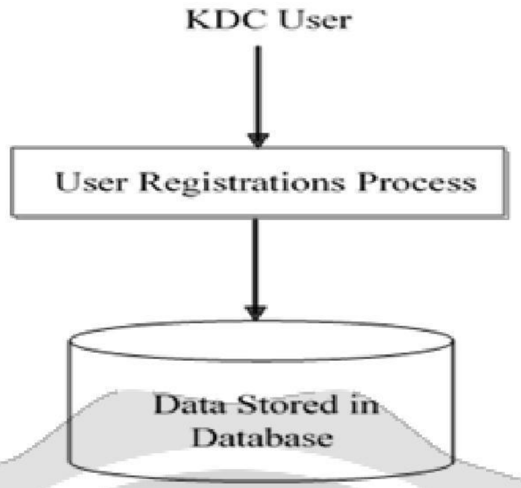


Fig.3. KDC Authentication

3.3 User Revocation

It should be ensured that users must not have the ability to access data, even if they have matching set of attributes which is required to access data on cloud. It is just not enough to store the information on cloud securely but also be important to ensure that user's identity must be protected from other users. For example, a user store some sensitive information on cloud without revealing his/her identity. However, the user have to prove to the other users that he/ she is a valid user who stored the information without revealing the identity. Other users or the cloud can verify the user and the validity of the message stored on cloud.

3.4 Privilege Trees Tp

In our work, encryption policy is described with a tree called access tree. Each non-leaf node of the tree is a threshold gate, and each leaf node is described by an attribute. One access tree is required in every data file to define the encryption policy.

In this paper, we extend existing schemes by generalizing the access tree to a privilege tree. The privilege in our scheme is defined as similar to the privileges managed in ordinary operating systems. A data file has several operations executable on itself, and each of them is allowed only to authorized users with different level of qualifications. For example, {Read_mine, Read_all, Delete, Modify, Create}is a privileges set of students' grades. Then, reading Alice's grades is allowed to her and her professors, but all other privileges should be authorized only to the professors, so we need to grant the "Read_mine" to Alice and all other to the professors. Every operation is associated with one privilege , which is described by a privilege tree Tp. If a user's attributes satisfy Tp, he is granted the privilege p. By doing so, we not only control the file access but also control other executable operations, which makes the file controlling fine-grained and thus suitable for cloud storage service.

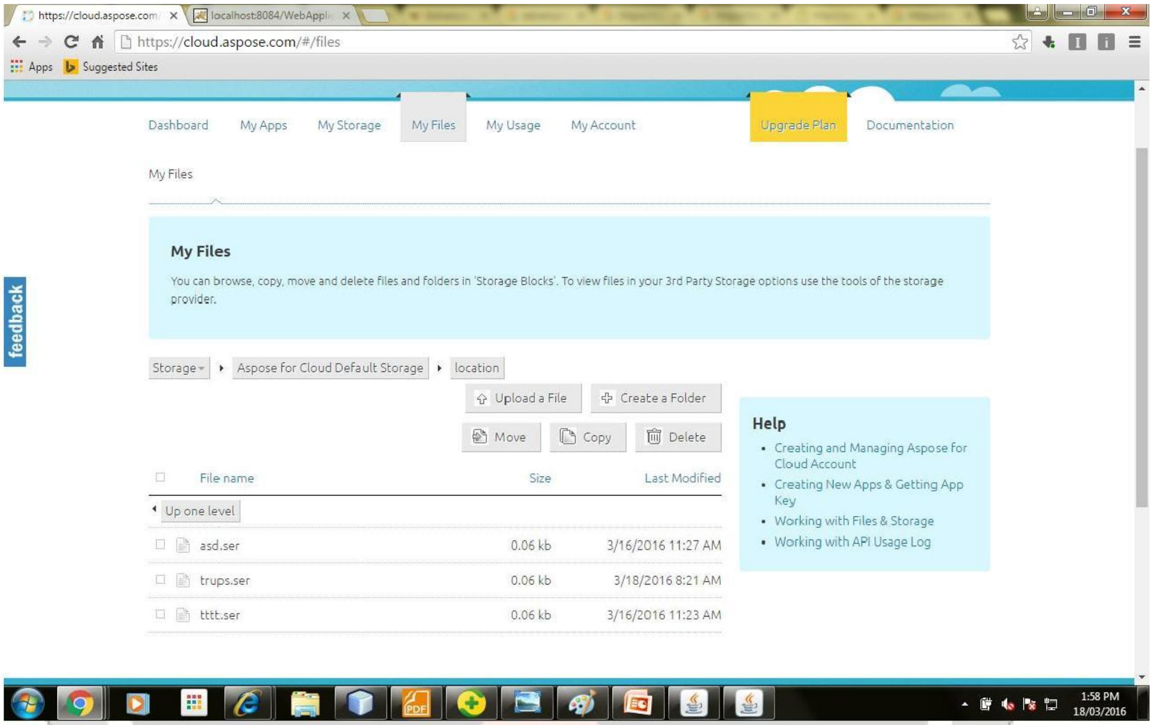
In our scheme, several trees are required in every data file to verify users' identity and to grant him a privilege accordingly.There are supposed to ber these kind of structures, which means there arer different privileges defined for the corresponding data file. The privilege 0 is defined as the privilege to read the file, and other privileges may be defined arbitrarily (the m-th privilege does not necessarily have more powerful privilege than then-th one when m>n). Given a tree, if numx is the number of the nodex's children node and kx is its threshold value $0<kx \leq numx$, then node xis assigned a true value if at least kx children nodes have been assigned true value. Specially, the node becomes an OR gate when kx =1 and an AND gate whenkx =numx.

Satisfying the Privilege Tree:

If a user's attributes set Ssatisfies the privilege treeTp or the nodex, we define it as Tp(S)=1orx(S)=1 respectively. Tp(S)is calculated recursively as follows. If xis a leaf node, x(S)=1 if and only ifatt(x)∈S.Ifx is a non-leaf node, x(S)=1 only when at leastkx child nodes return 1. For the root node Rp ofTp, Tp(S)=1 only ifRp(S)=1.

4. EXPERIMENTAL RESULTS

On a cloud server data is stored in a encrypted format. The cloud does not know the identity of the user who stores data. by using Decentralized access cloud overload is reduced. Privileged tree are used to access to the data is controlled. The multiple KDC's reduce the problem of the single point failure and provided the read as well as write key.



My Files

You can browse, copy, move and delete files and folders in 'Storage Blocks'. To view files in your 3rd Party Storage options use the tools of the storage provider.

File name	Size	Last Modified
asd.ser	0.06 kb	3/16/2016 11:27 AM
trups.ser	0.06 kb	3/18/2016 8:21 AM
tttt.ser	0.06 kb	3/16/2016 11:23 AM

Help

- Creating and Managing Aspose for Cloud Account
- Creating New Apps & Getting App Key
- Working with Files & Storage
- Working with API Usage Log

Fig 4. Encrypted data stored on cloud

5. CONCLUSIONS

A decentralized access control technique with anonymous authentication, which provides user revocation and prevents replay attacks, is achieved. The cloud does not know the identity of the user who stores information, but only verifies the user's credentials. Key distribution is done in a decentralized way and also hide the attributes and access policy of a user. Cloud does not knows the access policy for each record stored in the cloud. Instead of storing access policy of user on cloud we use cloud handler for that purpose so that cloud never know the access policy of user.

6. ACKNOWLEDGEMENT

We offer special thanks to the Prof. D. V. Dhawase who have guided towards development of our system paper. Also thanks all who helped in the development of system and giving their valuable suggestions. So that we are able to improve our system.

7. REFERENCES

[1]T. LATHA and B.BHARATH KUMAR," Anonymous Authentication for secure decentralized access control in cloud", JULY 2015

[2] J.Ganeshkumar, N.Rajesh, J.Elavarasan, Prof.M.Sarmila, Prof.S.Balamurugan, "A Survey on Decentralized Access Control Strategies for Data Stored in Clouds", January 2015

[3] Investigations on Decentralized Access Control Strategies for Anonymous Authentication of Data Stored In Clouds.Authers: J.Ganeshkumar, N.Rajesh, J.Elavarasan, Prof. M.Sarmilaand Prof. S.Balamurugan ,2015

[4]Pooja R. Vyawhare, Prof.Namrata D. Glues "User Anonymous Authentication Scheme for Decentralized Access Control in Clouds", 2015

[5]Sandhya Medavarapu, and Madhira Srinivas," Authentication for Data Storage in Clouds with Anonymous Decentralized Access Control", 2015.

[6] Decentralized Access Control With Anonymous Authentication of Data Stored in Cloud.Authors:Sushmita Ruj,Milos StojMenovics and Amiya Nayak. IEEE Transactions on Parallel and Distributed Systems, VOL. 25, NO. 2, February 2014.

[7] R.Ranjith and D.Kayathri Devi, "Secure Cloud Storage using Decentralized Access Control with Anonymous Authentication", 2013

[8] Matthew Green, Susan Hohenberger and Brent Waters, "Outsourcing the Decryption of ABE Ciphertexts,"in USENIX Security Symposium, 2011.

[9] C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou, "TowardSecure and Dependable Storage Services in Cloud Computing", IEEE T. Services Computing, vol. 5, no. 2, pp. 220–232, 2012.

[10] Kan Yang, Xiaohua Jia and Kui Ren, "DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems", IACR Cryptology ePrint Archive, 419, 2012.

[11] J. Hur and D. Kun Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems", IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 7, pp. 1214–1221, 2011.

[12] A. B. Lewko and B. Waters, "Decentralizing attribute- based encryption," in EUROCRYPT, ser. Lecture Notes in Computer Science, vol. 6632. Springer, pp. 568–588, 2011.

[13] Matthew Green, Susan Hohenberger and Brent Waters, "Outsourcing the Decryption of ABE Ciphertexts,"in USENIX Security Symposium, 2011

[14] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in IEEE INFOCOM. , pp 441–445, 2010

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

[16] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing," in CloudCom, ser. Lecture Notes in Computer Science, vol.5931. Springer, pp. 157–166, 2009.

[17] X. Liang, Z. Cao, H. Lin and D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption," inACM ASIACCS,pp 343–352, 2009.s

[18] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[19] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and efficient access to outsourced data," in ACM Cloud Computing Security Workshop (CCSW), 2009

[20] H. Lin, Z. Cao, X. Liang and J. Shao, "Secure Threshold Multi-authority Attribute Based Encryption without a Central Authority," in INDOCRYPT, ser. Lecture Notes in Computer Science, vol. 5365, Springer, pp. 426–

436,2008.

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

[22] M. Chase, “Multi-authority attribute based encryption,” in TCC, ser. Lecture Notes in Computer Science, vol. 4392. Springer, pp. 515–534, 2007

[23] J.Bethencourt, A. Sahai, and B. Waters, “Ciphertextpolicy attribute-based encryption,” in IEEE Symposium on Security and Privacy. , pp. 321–334, 2007.

[24] Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in ACM Conference on Computer and Communications Security, pp. 89–98, 2006.

[25] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in EUROCRYPT, ser. Lecture Notes in Computer Science, vol. 3494. Springer, pp. 457–473, 2005.