International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-5 Issue-1, October 2015

Secure Data Storage in Cloud using Centralized Access Control with Anonymous Authentication

Athira R, Lekshmy D Kumar

Abstract— Cloud computing's multi-tenancy feature which provides privacy, security and access control challenges because of sharing of physical resources among untrusted tenants. Much of the data stored in clouds is highly sensitive particularly in the case of medical records and social networks. Security and privacy are very important issues in cloud computing. In one hand, the user should authenticate itself before initiating any transaction, and on the other hand, it must be ensured that the cloud does not tamper with the data that is outsourced. User privacy is also required so that the cloud or other users do not know the identity of the user. The validity of the user who stores the data is also verified. In order to achieve safe storage, a suitable encryption technique with key management should be applied before outsourcing the data. A new decentralized access control scheme is implemented for secure data storage in clouds, which supports anonymous authentication. In this scheme, the cloud verifies the authenticity of the user without knowing the user's identity before storing data. The scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents replay attacks and supports creation, modification and reading data stored in the cloud. User revocation is also addressed. Moreover, the authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized.

Index Terms—Attribute based encryption, Access control, Authentication.

I. INTRODUCTION

Cloud computing is the emerging technology today. It delivers a wide range of resources like computational power, computational platforms, storage and application to users via internet. There are several definitions for cloud computing, But the standard definition was given by NIST(National Institute of Standards and Technology) as "Cloud Computing is model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources(eg: networks, servers, storages, applications and services)that can be rapidly provisioned and release with minimal management effort or service provider interaction"[1]. Research in cloud computing is receiving a lot of attention from both academic and industrial world as it reduce the costs in organizations and at the same time offer on demand resources and computation without requiring to create an IT infrastructure. Cloud provide services such as Amazon Web Services(AWS) or Microsoft Azure, provide in cloud a means for organizations to instantly provision and deprovision virtual machines(VM) depending on their needs, just paying for what they use. In cloud computing users can outsource their computation and storage to ensure servers using internet. This frees users from the hassles of maintain resources on-site. Therefore cloud service provider must provide the trust and security, as there is valuable and sensitive data in large amount stored on the clouds. The data must be encrypted before uploading to the cloud by using some cryptographic algorithms for protecting the confidentiality of the stored data. Security and privacy are the most important issue in cloud computing because most of the data stored in cloud is highly sensitive. For this reason the user need to authenticate itself before initializing any transaction and must

Confirm that the cloud does not tamper with the data outsourced. Another important issue is user privacy that is the identity of each user is protected from the cloud and their users. Cloud itself is liable for the services it provide, the validity of the users and the user authentication for the data outsourced. To provide a secure data storage in cloud the data must be encrypted before outsourcing. On the other hand a proficient search on the encrypted data is also important. Nonetheless a searchable encryption [2][3]mechanism is used where the cloud should not know the query but should be able to return the record that satisfy the query[4].The explicit records are returned only when searched with correct keywords. Therefore the main issue is that the records must contain the keywords to ensure the search. The security and privacy protection in cloud include authentication of users using public key cryptographic [5] techniques and using homomorphic encryption [6][7] techniques to ensure that cloud cannot read the data while performing computation on them. In the traditional methods the data is encrypted with user's public key and the data owner encrypt the data using this public key and then upload the files to the cloud. If users want to download the file then they must use their generated secret key to decrypt the file. The problems of using this method is that the owner has to get the public key of the user and the same data is encrypted with different public keys this results in storage overhead. This method is not suitable for the situations where the same data is used by multiple users because each time the data is encrypted using the user's public key. In order to overcome this issue a relatively recent approach known as "Attribute Based Encryption (ABE)" that reconsiders the concept of public key cryptography is used.

In ABE the identity is defined as a set of attributes for eg, roles and messages can be encrypted with respect to subsets of attribute. The key issue is that someone should only be able to decrypt a cipher text if the person holds a key for "matching attributes" where user keys are always issued by some trusted party.

Manuscript published on 30 October 2015.
* Correspondence Author (s)
Athira R*, Department of Computer Science, University of Kerala/ SCT College of Engineering, Trivandrum, India,.
Lekshmy D Kumar, Department of Computer Science, University of Kerala / SCT College of Engineering, Trivandrum, India.

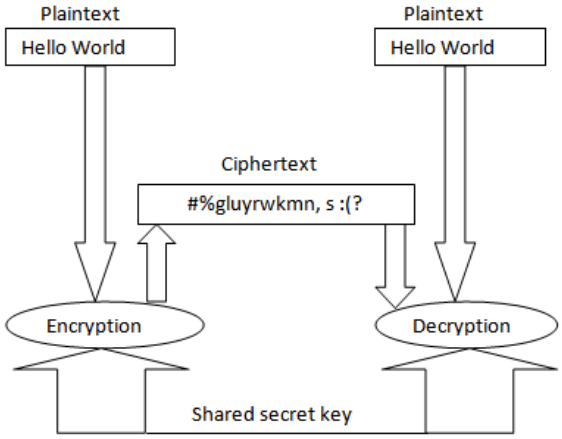

Fig: Encryption and Decryption

II. Related Works

ABE was proposed by Sahai and Waters [8]. In ABE, in addition to its unique ID the user has a set of attributes. There are mainly two classes of ABEs. In key-policy ABE or KP-ABE (Goyal et al. [8]), to encrypt data, the sender has an access policy. The attributes and keys of a writer has been revoked .He cannot write back stale information. The attribute authority sends attributes and secret keys to receiver. He is able to decrypt information if it has matching attributes. In Ciphertext-policy, CP-ABE ([7], [8]), the receiver has the access policy in the form of a tree. The attributes as leaves and monotonic access structure with AND, OR and other threshold gates. Centralized approach is taken by all the different approaches. Here allow a single KDC(key distribution center), which is a single point of failure. Chase [9] proposed another scheme a multiauthority ABE, in which multiple KDC authorities (coordinated by a trusted authority) are used which distribute attributes and secret keys to users. Multiauthority ABE protocol is clearly described in [10] and [11], which required no trusted authority. Every user require attributes from all the KDCs. Recently, Lewko and Waters [12] proposed a fully decentralized ABE where it does not require a trusted server and users could have zero or more attributes from each authority. Decryption is done at user's end and computation overhead is high. So, this technique might be inefficient when users access using their mobile devices. To overcome this problem, Green et al.[13] proposed a technique to outsource the decryption task to a proxy server, so that the user can compute with minimum resources (for example, hand held devices). However, the presence of one proxy and one KDC makes it less efficient than other decentralized approaches. Both these approaches had no way to authenticate users, anonymously. Yang et al. [14] presented a modification for [15], authenticating users, where user remains anonymous while accessing the cloud server. For ensuring anonymous user authentication ABSs were introduced by Maji et al. [13]. This was also a centralized approach. A recent scheme by Maji et al. [16] takes a decentralized approach and provides authentication where the identity of the users is not revealed. But it is prone to replay attack

III. System Overview

The main contributions of this paper are the following:
1. Distributed access control of data stored in cloud.
2. Authorized users with valid attributes can access them.
2. Authentication of users who store and modify their data on the cloud.
3. During authentication the identity of the user is protected from the cloud.
4. The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized.
5. Revoked users cannot access data after they have been revoked.
6. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information.

Assumptions of this paper are the following:
1. The cloud is honest-but-curious, which means that the cloud administrators can be interested in viewing user's content, but cannot modify it. This is a valid assumption that has been made in [12] and [13]. Honest-but-curious model of adversaries do not tamper with data so that they can keep the system functioning normally and remain undetected.
2. Users can have either read or write or both accesses to a file stored in the cloud.
3. All communications between users/clouds are secured by secure shell protocol, SSH.

A privacy-preserving Key Distribution Center (KDC) scheme to protect the user's privacy is proposed. In this scheme, all the user's secret keys are tied to his identifier to resist the collusion attacks while the multiple authorities cannot know anything about the user's identifier. In order to provide a secure storage and access of the data Cipher Text Policy Attribute Based Encryption protocol is used.

There are three types of users known as creator, Reader and writer and each of them is assigned specific access conditions based on which they can create, read, write or modify the data stored in the cloud. Here the creator is the person who has the privilege to create a data and store it in the cloud. At the time of encryption of the data the set of attributes are assigned to the data so that if a user want to access the data, then he/she must have the set of matching attributes only then they can decrypt the data. The reader is the person who has only read access and cannot add or modify the data. Write is given the write access and can add or modify the data stored in the cloud.

When the creator presents the id the trustee gives a token 'γ'.This token is then presented to the key distribution centre (KDC).There are several KDC's for example servers in different part of the world. The KDC then pass the key to the creator for encryption and decryption and for signing. The message is then encrypted under the access policy 'X'. This access policy decides who can access the data stored in the cloud.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

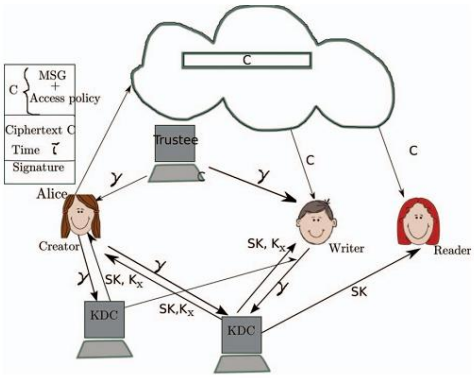

Fig 2: Secure cloud storage

The authenticity of the creator is proved by using the claim policy 'Y' which is decided by the creator and then signs the message under this claim. In order to store the data in the cloud the cipher text along with the signature is send to the cloud and the cloud verifies the signature and stores the data in cloud. When a reader want to read the data the cloud sends the ciphertext and if the user has matching set of attributes then they can decrypt and get the original data .If a reader wants to read data stored in the cloud it uses the secret key to decrypt it.

IV. CIPER-TEXT POLICY

In cpabe attribute-based encryption a user's private key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a ciphertext if and only if his attributes satisfy the policy of the respective ciphertext. Policies may be defined over attributes using conjunctions ,disjunctions and (k,n) threshold gates, i.e, k out of n attributes have to be present. For e.g. if there is a universe of attributes {A,B,C,D} and two users user1 and user2 .If user1 receives a key to attributes {A,B} and user2 to attributes {D}. If a ciphertext is encrypted with respect to the policy (A∧C) ∨ D, then user2 will be able to decrypt, while user1 will not be able to decrypt the data because user2 satisfy the policy and user1 does not satisfy. CPABE thus allow to realize implicit authorization, i.e, authorization is included into the encrypted data and only people who satisfy the associated policy can decrypt data. Another important feature is that the users can obtain their private keys after data has been encrypted with respect to policies. So data can be encrypted without the knowledge of the actual set of users that will be able to decrypt, but only specifying the policy which allows to decrypt them. The access policy is selected based on a specific knowledge of the data, it is done by the person who is holding the secret data. The person does not know the exact identities of all other people who should access the data, but only have a way to describe them in terms of descriptive attributes. This eliminates the need to rely on the data storage for preventing unauthorized data access. In this system the public keys are hidden in the ciphertext and the users can only check whether their own keys are eligible to decrypt the ciphertext without knowing other information. An ciphertext-policy attribute based encryption scheme consists of four fundamental algorithms: Setup, Encrypt, KeyGen, and Decrypt.

1. Setup: This is a randomized algorithm that takes a security parameter as input, and outputs the public parameters PK and a master key MK. PK is used for encryption and MK is used to generate user secret keys and is known only to the central authority.

2. Encryption: This is a randomized algorithm that takes as input a message M, an access structure T, and the public parameters PK. It outputs the ciphertext CT.

3. KenGen: This is a randomized algorithm that takes as input the set of a user (say X)'s attributes SX, the master key MK and outputs a secret key SK that identifies with SX.

4. Decryption: This algorithm takes as input the ciphertext CT, a secret key SK for an attribute set SX. If SX satisfies the access structure embedded in CT, it will return the original message M.

CPABE as proposed by J. Bethencourt [10] proceeds as follows [1]:

Let G_0 be a bilinear group of prime order p and let g be a generator of G_0. In addition, let $\hat{e} : G_0 \times G_0 \rightarrow G_1$ denote the bilinear map. A function H : Z ∗ p → G_0 to map any attribute to an element in G0 , where H(i) = g i is used . This crytosystem consists of the following algorithms.

Setup: This algorithm chooses a bilinear group G_0 of prime order with generator g and two random exponents and produces a public key PK

PK=G_o, g, h=g^{β}, f=$g^{1/\beta}$,$e(g,g)^{\alpha}$……(1)

and a master key MK as (β, g^{α})

Encrpyt (PK,M, T) : The encryption algorithm encrypts a message M under the tree access structure T. The algorithm first chooses a polynomial q_x for each node x (including the leaves) in the tree T. These polynomials are chosen in the following way in a topdown manner, starting from the root node R. For each node x in the tree, set the degree d_x of the polynomial q_x to be one less than the threshold value k_x of that node, that is, d_x=k_x -1.

Starting with the root node R the algorithm chooses a random $s \in Z_p$ and sets $q_R(0) = s$. Then, it chooses dR other points of the polynomial q_R randomly to define it completely. For any other node x, it sets $q_x(0) = q_{parent}(x)(index(x))$ and chooses dx other points randomly to completely define q_x. Let, Y be the set of leaf nodes in T. The ciphertext is then constructed by giving the tree access structure T and computing

$$CT=(T, \tilde{C} = Me(g,g)^{\alpha s}, C = h^s,$$

$$\forall y \in Y: Cy = g^{gy(0)}, C'y = H\big(att(y)\big)^{gy(0)} \text{)……(2)}$$

KeyGen (MK, S): The key generation algorithm will take as input a set of attributes S and output a key that identifies with that set. The algorithm first chooses a random $r \in Z_p$, and then random $r_j \in Z_p$ for each attribute $j \in S$. Then it computes the key as

$$SK=(D = g^{(\alpha+\gamma)/\beta},$$

$$\forall j \in S: D_j = g^r.H(j)^{r_j}, D_j^{'} = g^{r_j} \text{……(3)}$$

Decrypt(CT, SK): A recursive algorithm DecryptNode(CT, SK, x) that takes as input a ciphertext CT =(T, $\tilde{C}$,C,$\forall y \in Y$: Cy, C_y'), a private key SK, which is associated with a set S of attributes, and a node x from T. If the node x is a leaf ode then let $i = att(x)$ and define as follows: If $i \in S$, then

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

$$\text{Decrypt Node}(CT, SK, x) = \frac{e(D_i, C_x)}{e(D_i', C_x')} = \frac{e(g^r.H(i)^{r_i}, h^{qx(0)})}{e(g^{r_i}.H(i)^{qx(0)})} \ldots\ldots\ldots\ldots(4)$$

If $i \in S$, then define *DecryptNode*(*CT*,*SK*,*x*) = ⊥. Now consider the recursive case when x is a nonleaf node. The algorithm DecryptNode(CT, SK, x) then proceeds as follows: For all nodes z that are children of x, it calls ecryptNode(CT, SK, z) and stores the output as F_z. Let S_x be an arbitrary k_x sized set of child nodes z such that $F_z \neq \perp$. If no such set exists then the node was not satisfied and the function returns ⊥. Otherwise, compute and return result.

$$Fx = \prod_{z \in Sx} F_z^{\Delta i, S_x'(0)}$$

$$= \prod_{z \in S_x} (e(g,g)^{r.qz(0)})^{\Delta i, S_x'(0)}$$

$$= \prod_{z \in S_x} (e(g,g)^{r.q_{parent(z)}^{(index(z))}})^{\Delta_i, S_x'(0)}$$

$$= \prod_{z \in S_x} (e(g,g)^{r.qx(i)})^{\Delta_i,\, s_x'(0)}$$

$$= e(g,g)^{r.q_x(0)}$$

where $i = index(z), S_x' = \{index(z) : z \in S_x\}$

Now function DecryptNode defined and then defines the decryption algorithm. The algorithm begins by simply calling the function on the root node R of the tree *T*. If the tree is satisfied by S, set $A = DecryptNode(CT,SK,r) = e(g,g)^{rq_{R(0)}} = e(g,g)^{rs}$. The algorithm now decrypts by computing

$$\tilde{C}/(e(C,D)/A) = \tilde{C}/(e(h^s, g^{\alpha+\gamma)/\beta}/e(g,g)^{rs}) = M \ldots(5)$$

V. Experimental Results And Analysis

In this scheme the attribute based encryption is used, which provides a secure storage of the data in cloud. The encryption and decryption is based on the set of attributes of the users, here the time taken for uploading a file is comparatively high. In this scheme a single user that has the admin power can only upload the file and encrypt the file a set of attributes so that if any other users want to download the file they must possess a set of matching attributes. At the time of uploading the admin shares the key with users it wishes to view the file. Only if the key is shared with the user and it has the matching set of attribute they can download the file and view it. The main disadvantage is that only the admin can upload the file and at the time of uploading itself it should share the key with the user.

The problem here is that if after the file uploading the admin wants to share the key with other user then it must upload the file once again and share the key with the user it wishes. This is a time consuming process where the user performs all the actions of uploading and then edit the policy or share the key with the user.

In order to reduce the time of editing the policy and allowing other user to upload the file in the cloud and share the key with other users, a group management scheme is used. In this scheme the write of uploading a file is given to all the group member of the group. Each member of the group has the right to upload a file and share the key with the other users. If the owner of the data wishes to add another user to the group and share the key the user it can just login and edit the policy and the user also has the ability to add or remove the key sharing with the users.

The main aim of using group sharing in cloud computing is to reduce the time of uploading the file and editing the policy. When compared with the previous method the time is comparatively less. There is a constant variation in the graph between the time of the previous method and the current method.

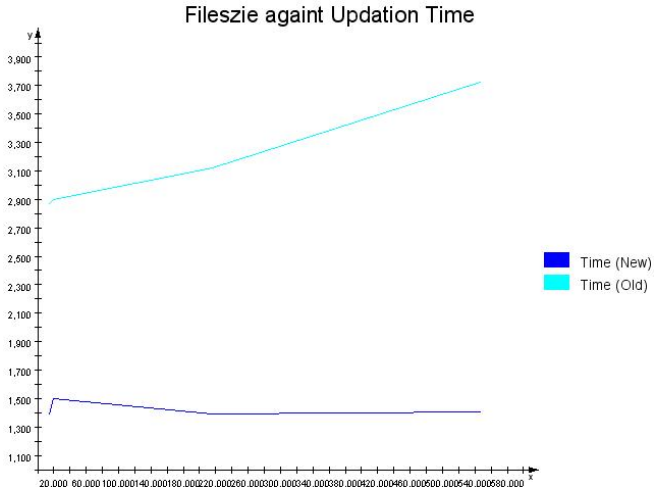

Fig.3 : Performance graph

VI. Conclusion

In this paper a centralized access control with anonymous authentication of data stored in cloud is achieved. The main aim of this work is to maintain the anonymity of the user and to prove that they are valid users. This was achieved and the time of uploading the data was comparatively reduced when compared with other existing methods. Here the key distribution is done in a decentralized way to avoid the loss of the data.

References

1. S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds," Proc.IEEE/ACM Int'l Symp. 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 Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.
3. 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.
4. S. Kamara and K. Lauter, "Cryptographic Cloud Storage,"Proc.14th Int'l Conf. Financial Cryptography and Data Security,pp. 136-149, 2010.
5. C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., http://www.crypto.stanford.edu/craig, 2009.D. Chaum and E.V. Heyst, "Group Signatures," Proc. Ann. Int'lConf. Advances in Cryptology (EUROCRYPT), pp. 257-265, 1991.
6. H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance,"IACR Cryptology ePrint Archive, 2008.[24] H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures," Topics in Cryptology - CT-RSA, vol. 6558, pp. 376-392,2011.
7. A. Beimel, "Secure Schemes for Secret Sharing and Key Distribution,"PhD thesis, Technion, Haifa, 1996.

8. A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc.Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), pp. 457-473,2005.
9. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,"Proc. ACM Conf. Computer and Comm. Security, pp. 89-98, 2006.
10. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based encryption," Proc. IEEE Symp. Security and Privacy, pp. 321-334, 2007.
11. 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.
12. M. Chase, "Multi-Authority Attribute Based Encryption," Proc.Fourth Conf. Theory of Cryptography (TCC), pp. 515-534, 2007.
13. H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure Threshold Multi-Authority Attribute Based Encryption without a Central Authority,"Proc. Progress in Cryptology Conf. INDOCRYPT), pp. 426-436,2008.
14. M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conf.Computer and Comm. Security, pp. 121-130, 2009.
15. K. Yang, X. Jia, and K. Ren, "DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems," IACR Cryptology ePrint Archive, p. 419, 2012.
16. A.B. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," Proc. Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT),pp. 568-588, 2011.

Athira R is currently doing her post-graduation in Computer Science and Engineering at Sree Chitra Thirunal College of Engineering under Kerala University, Trivandrum, Kerala, India. Athira received her under graduation in Computer Science and Engineering from Sun College of Engineering, Nagarcoil under Anna University, Tamil Nadu, India in 2013. Her areas of interest includes network security, distributed computing and cloud computing.

Lekshmy D Kumar is working as Assistant professor at the department of computer science and engineering, Sree Chitra Thirunal College of Engineering, Trivandrum, Kerala. She did her B.Tech degree at Sree Chitra Thirunal College of engineering, Trivandrum from University of Kerala. She did her M.Tech degree from NIT, Surathkal. Now she is also doing research in System Analysis and Computer Applications. She published her research works in many international journals.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)