# Security Provision for Data Stored in Cloud using Decentralized Access Control with Anonymous Authentication

Afsha Pathan
Department of Computer Engineering
JSCOE, Pune, India

M. D. Ingle
Department of Computer Engineering
JSCOE, Pune, India

## ABSTRACT
As we store more sensitive data on cloud and also share it with third party. Though cloud storage provides on-demand services to multiple users but there are several issues related security. So in this paper we are trying to address security related issues of cloud, so that users can store any confidential data on cloud without any fear and enjoy the services provided by cloud. In this paper we have proposed a new decentralized access control mechanism to secure data stored on cloud. This proposed scheme will support anonymous authentication scheme. In that system will initially verifies the identity of the user .Here Authentication and access control schemes are decentralized unlike other systems which are centralized. Also here we are storing access policies on cloud in hash codes so that we can hide it from third parties. The proposed scheme also addresses the problem of replay attack and user revocation. So the proposed scheme is more robust than the existing scheme as it doesn't store the authentication information on cloud itself.

## Keywords
Cloud computing, Privacy Preservation, Access control, Anonymous Authentication, Decentralized

## 1. INTRODUCTION
Researches in cloud computing is receiving much more attention from different fields including education, banking, academics, medical etc. User stores their data into cloud that offers them a great convenience because they don't have to care about complexities of storing data into physical hardware. The pioneer vendors, Amazon Simple Storage Service (S3), and Amazon Elastic Compute Cloud (EC2) are both well-known examples of cloud computing. While these internet-based online services do provide huge amounts of storage space and customizable computing resources.

Cloud computing offers a great advantage for Private organisation and Government organisation. Because they can store their sensitive data on cloud and can protect their data from other users. As it involves deploying groups of remote servers and software networks that allows users to store data in centralized manner and on demand access to computer services and resources. Thus it frees the user from maintaining unnecessary sensitive information.

Though cloud computing provides lots of services but there are some issues related to security and privacy of data stored. Many user store their data into clouds but some cloud providers behave unfaithfully towards users, they compromise their data to third party. Much of the data in cloud is sensitive and this type of data need to kept confidential. Sensitive data can be Patients medical record, Government organisation's data etc. For privacy issue in cloud, user's identity is hidden from the cloud.

To keep the sensitive data secure the data is converted into cipher text format, and that cipher text is stored into cloud rather than original data. So that cloud unaware of original data and can only see cipher text data. Thus whenever user wants to access data on cloud he'll just send a query to cloud and cloud will return the record that satisfies the query. In this whole process cloud doesn't knows the actual query. [2] Discussed the challenges in security of cloud storage and achieved a trusted cloud storage which addresses various issues in cloud computing via technical and policy based approaches.

Here to get the exact idea of proposed system we are taking one example, consider a situation an employee John working in some government organisation i.e E.g. Income-Tax office. He's having some proofs of malpractice done by some reputed authority. So John wants to publish these proofs to higher authorities, his colleagues of the same organisation. While doing all these things he needs to remain anonymous. Here Access control takes main role. So that we can decide who to give the access of which data. But simultaneously we need to check that whether the information is coming from valid source or not.

Access control of data involves retrieval of secured data, so that the accessing data like sensible data should be much care taken. It allows the owner of data to specify the users who can access the data. So unauthorized users are taken away from accessing the data. That is the reason access control is achieving much more attention to achieve the security. There are three different types of access control mechanism: user based access control (UBAC), role based access control (RBAC), attribute based access control (ABAC).In first mechanism, user based access control owner of the data need to main a list of user that he wish to give the access to the data stored on cloud. So only the users included in list will get the access to the data. But this seems not a feasible scheme because there may be thousands of users in a system. In second mechanism, [2] role based access control the users are specified along with their individual roles. And access is given to the matching roles for e.g. Roles can be employees in finance department, or only administrator department. The third scheme is ABAC, attribute based access control users are given some attribute and the data has attached some access policy. Then users with matching attribute can only get access. For e.g. Access only get to the manager having experience more than 10 years, or an employee's having experience of 4-5 years. Paper [5] discusses the advantages

and disadvantages of RBAC and ABAC. Proposed scheme uses ABAC to specify for access control mechanism.

## 2. PROBLEM DEFINITION

To provide a new and more secured way so that we can save and share the user's files with other people based on their access policy in cloud storage. We are proposing this to provide service with 'Zero-Knowledge' privacy. 'Zero-Knowledge' privacy means the cloud server never knows the plaintext contents of the data it is storing. Therefore, the data is never at risk of being compromised or abused by either internal threats or external hackers. Even though there is physical access to the storage servers, the third party cloud cannot see even the names of client's files and folders. Because all the data stored on cloud is in encrypted form.

## 3. LITERATURE SURVEY

The [1] paper supports decentralized access control which is most robust than other centralized approach. The scheme also supports privacy preservation which is not supported by other scheme. But in this work cloud knows the access policy for each record stored in cloud so that so the proposed scheme avoids this limitation by storing this information on-site.

In Literature work in[4] manages the patient's data on cloud that results in reduction of operational cost. This work supports access control in cloud but it is centralized in nature. And also this scheme doesn't support authentication. Here to ensure that each owner has full control over his data, this scheme leverage attribute-based encryption (ABE) as the encryption primitive, and each owner generates his own set of ABE keys. On the other hand work proposed in [6] and [7] doesn't use the attribute based encryption(ABE).The work in [7] First encrypt every data block with a different symmetric key and adopt the key derivation method so to reduce the number of secrets that the data owner and end users need to maintain. . Second, it adopts over-encryption by the server to achieve data isolation among end users even when they have the same access rights. For the servers that refuse to conduct over-encryption, they propose to use lazy revocation to prevent revoked users from getting access to updated data.

The work in [6] uses Extensible Access Control Mark-up Language (XACML). That is the de-facto standard for authorization. It can use an attribute to indicate the tenant, and use that attribute in the policies. It defines XACML specification that indicates the security domain of the data. It also identifies the administrator and policy that manages the name-space in which the subject id is administered. The scheme [8] also uses decentralized access control but this doesn't authenticate them self before accessing data.

The scheme in [9] proposes a method by which we can preserve the policy of the owner of data .By using this approach the receiver has the access policy in the form of a tree. The tree contains attributes as leaves and monotonic access structure with AND, OR and other threshold gates. Encrypted information can be kept confidential even if the storage server is untrusted.

It is Secure against collusion attacks. It uses centralized scheme that contains single KDC (key distribution centre), that distributes secret keys to users. And user will use that secret key to access the data on cloud. The single KDC is not secure way because it may cause single point failure, it is not an only issue but it also very difficult maintain because there may be thousands of user. So it is not possible for single KDC to serve these thousands of user. The scheme used in [10] uses decentralized access control scheme. This scheme describes several Key Distribution Authorities (coordinated by a trusted authority) which distribute attributes and secret keys to users. Multi-authority Attribute Based Encryption protocol which requires no trusted authority which requires every user to have attributes from at all the KDCs. These KDC can serve any number of users so this scheme is more secure than centralized scheme.

The work proposed in [11] provides flexible distributed storage integrity auditing mechanism, utilizing the homomorphism token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. It Considers the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. It is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

Attribute based encryption (ABE) was proposed in [12], it is the attribute based encryption. It is another set of cryptographic techniques that allows the specification of a decryption policy to be associated with a cipher text. In a (cipher text-policy) attribute-based encryption scheme each user in the system is provided with a decryption key that has a set of attributes associated with it (i.e. "credentials"). A user can then encrypt a message under a public key and a policy. Decryption to the data will only allow if the attributes associated with the decryption key match the policy used to encrypt the data. Attributes are qualities of a party that can be established through relevant credentials. ABE is one-to-many encryption technique. Since then, several works used ABE to realize fine-grained access control for outsourced data. Here Users/data are classified according to their attributes, such as professional roles/data types. There are two forms of ABE, KP-ABE (Key-policy ABE) and CP-ABE (Cipher text policy ABE). The First form KP-ABE proposed in [13]. It develops a much richer type of attribute-based encryption cryptosystem and demonstrates its applications. In this system each cipher text is labelled by the encryptor with a set of descriptive attributes. Each private key is associated with an access structure that specifies which type of cipher texts the key can decrypt. Since the access structure is specified in the private key, while the cipher texts are simply labelled with a set of descriptive attributes.

Second form is CP – ABE(Ciphertext policy ABE) proposed in [9] in this scheme the encryptor can fix the policy, which can decrypt the encrypted message. The policy can be formed with the help of attributes. In CP-ABE, access policy is sent along with the cipher text. Here access policy can be expressed using AND, OR Boolean operators. The access policy can be represented by an n-ary tree, the leaf nodes represents the attribute present in the access policy, interior nodes represents the AND, OR operators. Each attribute in the leaf node can take multiple values. The value assigned for the leaf node by the secret sharing method will be distributed to these multiple values effectively.

## 4. IMPLEMENTATION DETAILS

### 4.1. Mathematical Model

Proposed system can be represented as a set

X = {I, O, SC, FC, C}

Where,

I=set of inputs

O=set of outputs

SC= set of outputs in success cases

FC = set of outputs in failure cases

C = set of constraints

Input set I = It is set of user details, requests of users for tokens, and set of encryption and decryption key.

Output set O = It is set of files downloaded, files uploaded and file modified.

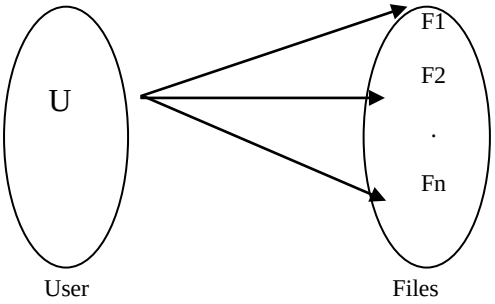


**Fig.1. mapping of user to files**

Figure shows one user is allowed to create/modify/delete multiple files. So above figure shows mapping between user and files one-to-many.

FC = {FDn, FUn, FMn, NULL}

Where,

FDo = invalid set of files downloaded

FUo = invalid set of files uploaded

FMo = invalid set of files modified

NULL represents no output

C = {C1, C2, C3}

Where,

C1 = "Every user must request a token before uploading a file into the cloud"

C2 = "All files must be encrypted before uploading into the cloud"

C3 = "All files must be decrypted after downloading from the cloud"

SC= this set represents the outputs of system in c success cases. This represents the system works properly all the valid user get the token's and required credentials to decrypt the data stored on cloud.

FC = this set represents the outputs of proposed system in failure cases. Here system incorrectly authenticates the valid users.

NP-Complete: - Our application is NP-Complete. Our application is a technique for Decentralized Access Control. In this system the access policy is defined on basis of that user type is identified. The user request for a token to TTP. Operations are categorized on the basis of user type. Thus in our application there is guarantee that we get output. Hence our application is NP-complete.

## 4.2 Proposed System Overview

In this paper we are proposing anonymous authentication and decentralized access control scheme for the user who wish to access the data stored on cloud that will make the cloud storage more secure than previous one. In existing system cloud verifies the authenticity of series of user's without having knowledge identity of user that stores data. The base paper has access policy that is defined for a file so that they can achieve access control. This access policy is maintained in plaintext format. Also the policy is stored on the same cloud where the data/file is stored. This introduces a risk of access policy tampering; as access policy is stored in plain text. Also attacker can easily view contents of access policy so it makes its job easier to attack, as he/she gets an idea for what profile (access policy contains user attributes) he/she has to look for. In order to remove security risk/threat to store the access policy on cloud in plain text format we propose a new way to define and store access policy.

In this system, once user has defined an access policy; we will calculate a hash value (digest using MD5). Once we are done with calculation of hash values, we will now construct a 2D (2-dimensional) array to store calculated hash values. Now this access matrix of hash values will be stored on cloud server along with file. Now as we have stored has values instead of plain text, attacker won't be able to fetch attributes of user. And access policy will remain anonymous.

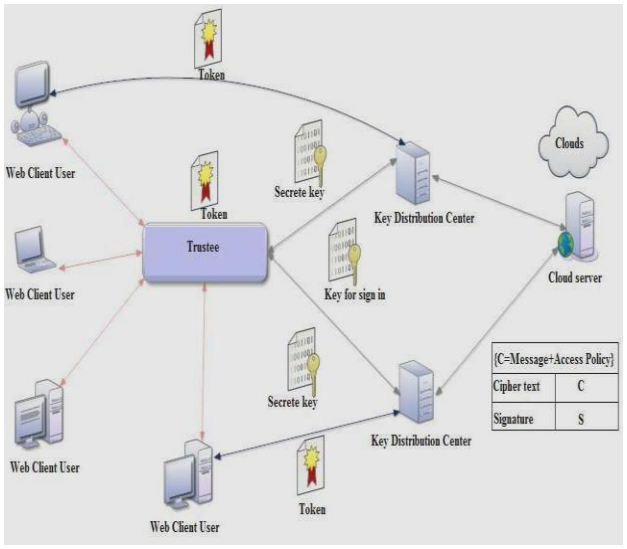


**Fig.2. System Architecture**

Figure shows the architecture of the proposed system. Architecture contains Trustee in the mid, it nothing but some government organization who manages the user's unique identity number for e.g. Pan Card, Aadhar card number. Whenever user wish to access the data on cloud the user need to authenticate himself to trustee with any unique identity number. The trustee verifies the number provided by user, if the number is valid then trustee provides one unique token to user.

After getting that token user goes to KDC i.e. Key distribution centre that provides secret keys to the user. There are multiple KDC's that are geographically located at different places. The user needs to submit the token given by trustee for next verification. The KDC takes the token of user and again varies from trustee to check whether that token is provided by trustee only.

After verification done successfully, KDC provides secret keys, if user is creator then he will get two keys encryption key and decryption key and if the user is reader then he'll get one key i.e. decryption key along with access policy. Then user will submit the secret keys to application server that is in middle of cloud and user and cloud. Through that server the user can access the cloud. All the access policies are saved on application server. It will check requirement of user and also check for file F if the access policy of file F and user matches then only user can access the file F.

### *4.1.1 .User Authentication*

Here user first authenticated by Trustee and receive one unique token T. KDC also authenticates the user and gives encryption key e, decryption key d and access policy x.

### *4.1.2. Writing data to Cloud Storage*

If the user is creator and want to store the file on cloud then he need to encrypt the message and then he can store it on cloud. So user will encrypt message by encryption key.

C = RSA.decrypt (msg, e)

Here we are using RSA algorithm for encryption and decryption. This is asymmetric cryptography algorithm. This provides much secure for data transmission. This encrypted message is then send to application server application server checks the access policy of user and then allowed him to store message on cloud. Also with each message we are sending time stamp value t, to avoid replay attacks. So the encrypted message is forwarded to the cloud server in form:

Message = (C, d, Y)

Where X is the access policy formed by the creator that is stored on the cloud server in the form of hash value.

### *4.1.3. Reading data from cloud Storage:*

If the user is reader then user must have valid decryption key d given by KDC. Then only user can decrypt the message and also the user's access policy must match with message's access policy. Decryption proceeds in following manner.

P = RSA.decrypt(C, d, Y)

Here Y is the access policy of the reader. After sending access policy to cloud server the cloud server will verifies the access policy of reader if it is valid then only he'll is allowed to access the message stored on cloud.

## 5. RESULT DISCUSSION

We have presented a decentralized access control technique with anonymous authentication, which

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. Key distribution is done in a decentralized way.

Following screenshots represents the results when user login as a reader and when user is writer. When User login as a reader he/she'll get one key i.e. decryption key to decrypt data on cloud and when user login as writer he/she will get two keys encryption key and decryption key.

Our scheme is robust and decentralized; most of the others are centralized. Our scheme also supports privacy preserving authentication, which is not supported by others.

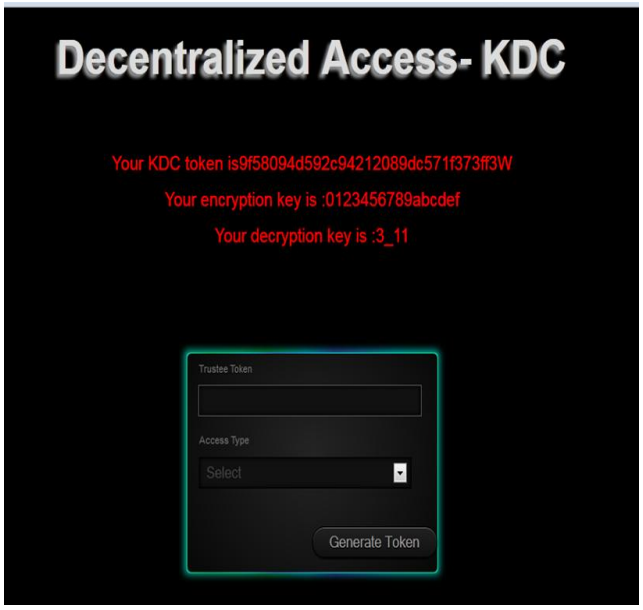


**Fig.3. User as Writer**


**Fig.4. User as Reader**

## 6. CONCLUSION AND FUTURE SCOPE

Thus this paper presents decentralized approach for access control that enables user to decide access policy by their own. It also provide anonymous authentication that enhances the security of data stored on cloud. The proposed scheme prevents the replay attacks and also user revocation. This system provides more security to the data because the access policies for the data are not stored on same cloud but they are stored on cloud server in form of hash values. The proposed system is secure cloud storage using decentralized access control with anonymous authentication.

In Future, I would like to implement the same concept for the public cloud to enhance the security of public cloud. Also we can enhance the system's security by adding feature of key management. In that system will maintain keys of different user.

## 7. ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Prof. M.D. Ingle for his exemplary guidance, monitoring and constant encouragement which helped me in completing this task through various stages. The blessings, help and guidance given by him time to time shall carry me a long way in the journey of life which I am about to embark.

## 8. REFERENCES

[1] 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, VOL. 25, NO. 2, FEBRUARY 2014

[2] R.K.L. Ko, P. Jagad pramana, M. Mowbray, S. Pearson, M.Kirchberg, Q. Liang, and B.S. Lee, "Trustcloud: A Framework for Accountability and Trust in Cloud Computing," HP Technical Report HPL-2011 38, http://www.hpl.hp.com/techreports/2011/HPL-2011-38.html, 2013.

[3] D.F. Ferraiolo and D.R. Kuhn, "Role-Based Access Controls,"Proc.15th Nat'l Computer Security Conf., 1992.

[4] 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'lICST Conf. Security and Privacy in Comm. Networks (SecureComm),pp. 89-106, 2010.

[5] D.R. Kuhn, E.J. Coyne, and T.R. Weil, "Adding Attributes to Role-Based Access Control," IEEE Computer, vol. 43, no. 6, pp. 79-81,June 2010.

[6] http://securesoftwaredev.com/2012/08/20/xacmlin-the-cloud,2013.

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

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

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

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

[11] 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.-June2012.

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

[13] 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