THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

Attribute Based Access Control Scheme for Secure Cloud Storage

Gore Swati S.
ME Student, Department of Computer Engineering
Sharadchandra Pawar College of Engineering, Dumbarwadi, Otur, Pune, India
Deokate Gajanan S.
Assistant Professor, Department of Computer Engineering
Sharadchandra Pawar College of Engineering, Dumbarwadi, Otur, Pune, India
Gumaste S. V.
Head of Department, Department of Computer Engineering
Sharadchandra Pawar College of Engineering, Dumbarwadi, Otur, Pune, India

Abstract:
Cloud computing allows user to remotely store their large amount of data into the clouds and can access when required. So here, security and privacy are very necessary terms in cloud computing. Providing access to valid users is very important as much of the data can be sensitive. This paper presents access control system for data stored in clouds that describes anonymous authentication and attribute based access control. In this scheme, the cloud first authenticates the user without knowing his/her identity before storing data in the cloud. This system has the feature of access control in which only authorised users can encrypt the stored information by using ABE (Attribute-Based Encryption) and paillier Encryption. Using this system user can upload his data securely in cloud, can update and can download at any time. This system is efficient as important data is stored securely. This system can be used for many applications such as storing health care information, in online social networking sites where users store their personal information and share with selected groups of users to whom they want to share. This system also handles user revocation when users are revoked from the system.

Keywords: *Cloud Storage, Attribute Based Access Control, Authentication, Attribute Based Signatures(ABS), Homomorphic encryption*

1. Introduction

Today due to the advances in networking technology and an increase in the need for computing resources have prompted many organizations to outsource their storage and computing needs. This computing model is commonly referred to as cloud computing and includes various types of services such as: infrastructure as a service (IaaS), where a customer makes use of a service provider's computing resources over the internet, platform as a service (PaaS), where a customer use the provider's resources to run custom applications; and finally software as a service (SaaS), where customers use software that is runing on the providers infrastructure.

It is important thing to preserve the security of data and privacy of users. Cloud should ensure that the users trying to access data and services are authorized users. Authentication of users can be achieved using different public key cryptographic techniques. Users should ensure that the cloud is not tampering with their data and computational results. It might also be important to hide the users identity for privacy reasons. For example, while storing medical records, the cloud should not be able to access records of a particular patient, given the particular identity. Users should also ensure that the cloud is able to perform computations on the data, without knowing the actual data. One way to carry computation on data by hiding the data from cloud, is by the use of homomorphic encryption techniques [10]. User sends homomorphicly encrypted messages, while the cloud without knowing the actual data performs computations on these encrypted messages and returns the results to the user.

Consider now the following situation. Patients store their medical records in the cloud. Different users can access different data fields of that information. Here the same data fields might be accessed by a selective group of people which is authorized set. For example, medical history of patients and drug dosses can be accessed by doctors and nurses, but not by hospital management staff. In online social networking , where owners are members of the networking site, they keep their personal details, pictures, music videos in the cloud and other members can view them depending upon their access rights. A member can post a message or upload a picture at any time, which will be visible only to the friends and certain selected communities that he/she belongs to, but it is not accessible to the rest users. It is important to protect privacy of these data from the cloud. Giving access rights to some authorised users and preventing other users from accessing the data, is called access control. There are three types of access control schemes.

1) User-based access control (UBAC)- In this type a list is maintained known as the access control list which contains the list of users who are authorized to access data. This is not feasible in clouds when number of user increases. In cloud computing, such lists can be much long and often dynamic, which will make handling such lists extremely difficult. Each time the list is to be checked to see if the user is valid or not. This results in a huge computation and storage costs.

2) Role-based access control (RBAC)- In this type users are classified based on their individual roles. Data is accessed by users only who have matching roles. The roles can be defined by the system itself. For example, only senior members and secretaries might have access to data but not the junior members.

3) Attribute-based access control (ABAC)- ABAC is more efficient for use, in which users are provided with attributes, and the data has access policy with it. Only users who are having valid set of attributes, satisfying the required access policy, can access the data. For instance, in the above example some records might be accessible by senior members with more than 10 years of experience or by secretaries with more than 8 years experience.

Another way to encrypt data is by using public keys of valid users, so that only they can decrypt data using their secret keys. However the same data then must be encrypted many times individually for each user, which may be result in large storage costs. Therefore in this system it is beneficent to use the cryptographic technique called Attribute Based Encryption (ABE) [4] to achieve access control in clouds. Using ABE, owners of the system can encrypt data with attributes that they have and store the information securely in the cloud. The cloud cannot access stored data. Users are provided with different set of attributes and secret keys by a key distribution centre i.e. KDC. Only Users with matching set of attributes can decrypt the stored information. For example, consider a public health records repository [5] system, the medical records have the history of the patients and might be accessed either by medical professionals such as doctors and nurses, researchers and academicians or management authorities such as insurance companies and government policy makers. Different people are allowed to access different records. Each user is given attributes such as the affiliation like hospital names, designation as a occupation and specialization etc. For example, only a psychiatrist or neurologist in Hospital A or B will be able to get the record, such as medical history of a bipolar disorder person. Any other user, for example a hospital staff of hospital A or B, or an eurologist from Hospital C, will not be able to access the record.

2. Literature Survey

Here first consider some existing schemes. Fuzzy-IBE [9] gives rise to two interesting new applications. The first application is an Identity-Based Encryption system that uses biometric identities. That is one can view a user's biometric, for example an iris scan, as the user's identity is described by different attributes and then encrypt data to user using their biometric identity. As many of the biometric measurements are noisy, it is not good to use existing IBE systems. The error-tolerance property of Fuzzy-IBE allows for a private key which is derived from a measurement of a biometric to decrypt a ciphertext encrypted with a slightly different measurement of the same biometric. Fuzzy IBE[9] can also used for an application called attribute-based encryption. In this case party will wish to encrypt a document to all users that have a certain set of attributes. For example, in a computer department, the chairperson might want to encrypt a document to all systems faculty on a hiring committee. In this case document would encrypt to the identity {"hiring-committee","faculty","systems"}. Any user who has an identity that contains all of these attributes could decrypt the document. The advantage of using Fuzzy IBE is that the document can be stored on a simple untrusted storage server instead of relying on trusted server to perform authentication checks before delivering a document.

ABE was proposed by Sahai and Waters [9]. In ABE, a user has a set of attributes with its unique ID. There are two main classes of ABEs. In key-policy ABE or KP-ABE [4], the sender has an access policy to encrypt data. A user whose attributes and keys have been revoked cannot write back any data. The receiver of the system receives attributes and secret keys from the attribute providing authority and is only able to decrypt information if it has matching attributes. In Cipher text-policy or CP-ABE, the receiver has the access policy in the form of a tree, having attributes as its leaves and monotonic access structure with AND, OR and other threshold gates.

2.1. Key-Policy Attribute-based Encryption (KP-ABE)

KP-ABE [4] is a crypto system for fine grained sharing of encrypted data. In KP-ABE cipher text is with attributes and private key are associated with access structures that decides which cipher text a user is able to decrypt. It is used for securing sensitive and important information stored by third parties on the internet. In this system each ciphertext is labelled by the encryptor with a set of different attributes. Each private key is provided with an access structure that specifies which type of ciphertexts the key can decrypt. Note that this setting is same as that of secret sharing schemes. Using different known techniques one can build his/her own secret-sharing scheme that specifies that a set of parties must cooperate with themselves in order to reconstruct a required secret. For example, one can specify a tree access structure where the interior nodes consist of AND and OR gates and the leaves consist of different parties. Any set of parties that satisfy the given tree structure can reconstruct the secret. In this construction each user's key is associated with a tree-access structure where the leaves are associated with attributes. A user can decrypt a ciphertext if the attributes associated with a ciphertext satisfy the key's access structure[4]. The primary difference between this setting and secret-sharing schemes is that while secret-sharing schemes allow for cooperation between different parties.. For instance, if Alice has the key associated with the access structure "P AND Q", and Bob has the key associated with the access structure "Q AND R", system would not want them to be able to decrypt a ciphertext whose only attribute is Q by colluding. To do this, this system adapts and generalizes the techniques to deal with more complex settings. This cryptosystem gives a powerful tool for encryption with fine-grained access control for applications such as sharing audit log information.

2.2. Cipher text Policy Attribute based Encryption (CP-ABE)

CP-ABE [3] is a policy to acquire complex control on encrypted data. This technique is used for keeping encrypted data confidential. In this system, a user's private key is associated with an arbitrary number of attributes expressed as strings. When a party encrypts a message using this system, they have to specify an associated access structure over attributes. A user only is able to decrypt a ciphertext if that user's attributes pass through the ciphertext's access structure. Access structures in this system is described by a monotonic "access tree", where nodes of the access structure are composed of threshold gates and the leaves describes the attributes. There AND gates can be constructed as n of n threshold gates and OR gates as 1 of n threshold gates. Furthermore, this scheme can handle more complex access controls such as numeric ranges by converting them to small access trees.

2.3. Multi-Authority Attribute-Based Encryption (MA-ABE)

MA-ABE [8] method allows any number of independent authorities to monitor attributes and provide secret keys. An encryptor in a system can choose, for each authority, a number d_k and a set of attributes, he can then encrypt a message such that a user can only decrypt if he has at least d_k of the given attributes from each authority k. Chase's scheme [8] is capable of handling disjoint sets of attributes that are distributed with multiple authorities. In this scheme, an encrypting party specifies a set of attributes A_s with the attributes in A_s being controlled by several authorities. Let A_t are be the set of attributes controlled by authority t. Then the ciphertext C associated with the attribute set A_c can only be decrypted by those users u with a set of attributes A_u for which the cardinality of the intersection $A_u \cap A_t \cap A_s$ exceeds the respective threshold d_t [8], for each authority t. one of the main challenges in implementing such a multi-authority attribute based encryption scheme is the prevention of collusion attacks [8] among different users that obtain secret key components from different authorities. Note that it is important that there is no communication between the individual authorities. To come out from these difficulties, Chase's scheme depends on a trusted central authority. The resulting scheme is able to tolerating multiple corrupted authorities, but the honesty of the central authority remains of vital importance since, by the constriction from [4], the trusted authority has the capability of decrypting every ciphertext.

3. Algorithms

3.1. ABE(Attibute Based Encryption)

It works under the following stages [3].

- Setup: This is a random algorithm that takes no input other than security parameter. It outputs the public parameters P and a master key K.
- Encryption: This is a random algorithm that takes as input a message m, a set of attributes n, and the public parameters P. It outputs the ciphertext C.
- Key Generation: This is a random algorithm that takes as input an access structure A, the master key K and the public parameters P. It outputs a decryption key D.
- Decryption: This algorithm takes as input the ciphertext C that was encrypted under the set n of attributes, the decryption key D for access control structure A and the public parameters P. It outputs the message M if $n \in A$.

3.2. ABS (Attibute Based Signature)

An Attribute-Based Signature (ABS) scheme is depend on a possible attributes A and message space M, and consists of the following four algorithms [3].

- ABS.TSetup (to be run by a signature trustee): Generates public reference information TPK.
- ABS.ASetup (to be run by an attribute-issuing authority): generates a two keys PK and SK
- ABS.AttrGen: On input (SK, $A \subseteq A$), outputs a signing key SK.
- ABS.Sign: On input (PK = (TPK, PK), SK, $m \in M$, Y), where Y(A) = 1, outputs a signature σ.
- ABS.Ver: On input (PK = (TPK, PK), m, Y, σ), outputs a boolean value 0 or 1.

3.3. Paillier Encryption

The Paillier Cryptosystem is well known Homomorphic encryption. It is a asymmetric algorithm for public key cryptography.
It works as follows.

3.3.1. Key Generation

- Select two large prime numbers p and q arbitrary and independent of each other such that gcd(n, Φ (n)) = 1, where Φ (n) is Euler Function , Φ (n) =((p-1)(q-1)) and n=ab.
- Calculate RSA modulus n = pq and Carmichael's function is given by λ = lcm (p-1, q-1).
- Select g called generator where $g \in Z_n^{2*}$ Select α and β randomly from a set Z_n^* then calculate g = $(\alpha n + 1)\ \beta^n mod\ n^2$.
- Compute the following modular multiplicative inverse μ = (L (g^λmod $n^2)^{-1}$ mod n. Where the function L is defined as L(u) = $(u-1)/n$.

The public (encryption) key is (n and g).
The private (decryption) key is (λ and μ).

3.3.2. Encryption

a. Let mes be a message to be encrypted where mes $\in Z_{n.}$
b. Select random r where r $\in Z_n^{*}.$
c. The cipher text can be calculated as:
Cipher text = $g^{mes}\cdot r^n$.mod n^2.

3.3.3. Decryption

a. Cipher text c $\in \mathbb{Z}_n^{2*}$
b. Original message: mes = L (cipher $^{\lambda}$ mod n^2).μ mod n.

Here,
Z_n^{2*}= set of integers co-prime to n^2
Z_n^{*}= set of integers co-prime to n
Z_n= set of integers n

4. Proposed Scheme

4.1. Architecture

According to this scheme of Privacy Preserving Decentralized Access control for Data in Cloud, a user can create his/ file and store it securely in the cloud in encrypted format. Many limitations of existing scheme are removed. Cloud does not know the user and his/her access policy. This scheme consists of use of the ABE and ABS and paillier Encryption,SHA-I.
Now discuss this scheme in details. Refer to the Fig. 1. There are three different users a creator, a reader, and writer. Initially the creator receives a token from the trustee who is assumed to be honest.

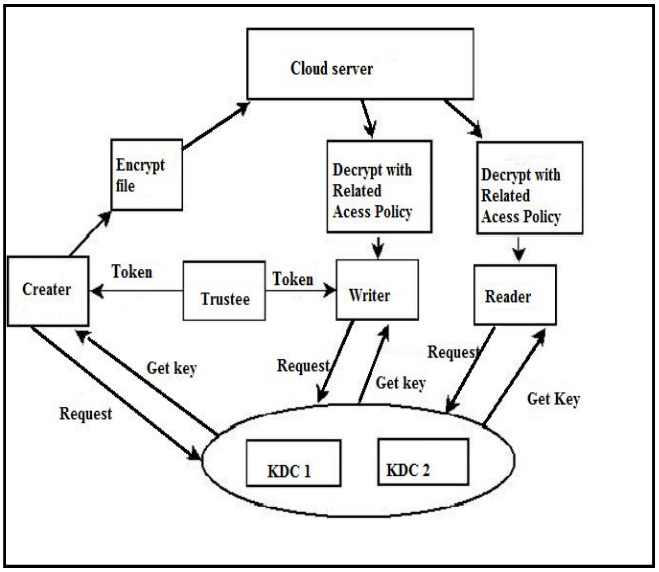

Figure 1: Proposed Architecture

A trustee can be someone like the federal government who is responsible for managing social insurance numbers etc. On presenting her/his id like health/social insurance number, the trustee gives her/him a token.
There are multiple KDCs used in this system, which can be scattered at different location. For example, these can be different servers in different parts of the world. A creator when gives the token to one or more KDCs he/she receives keys for encryption or decryption and signing. The message is encrypted under the access policy. The access policy will decide who can access the data stored in the cloud.
The creator will decide on a claim policy, to prove her/his authenticity and signs the message under this claim. The ciphertext with signature is C created using SHA-I, and given to the cloud. The cloud verifies the signature and it stores the ciphertext C. When a reader wants to read the data in cloud, the cloud sends ciphertext C. If the user has attributes matching with given access policy, he/she can decrypt and get its original message. Writing to cloud also proceeds in the similar way as file creation. By assigning the verification process to the cloud, it will help to relieve the individual users from the required time consuming verifications. When a reader needs to read some data stored in the cloud, it will tries to decrypt it using the secret keys it receives from the KDCs. If there is sufficient number of attributes matching with the access policy, then it decrypts the information stored in the cloud.

4.2. Working

4.2.1. Data Storage in Clouds

Here the KDCs are given keys for encryption and decryption and a key sk for signing and verifying. The users obtain attributes and secret keys from one or more KDCs. The message is encrypted using the equation,
C = ABE.Encrypt(MSG, key)

4.2.2. Reading from Cloud

When a user who wants a data from cloud sends request to cloud, then cloud sends Ciphertext c using SSH protocol. Decryption proceed using equation, ABE.Decrypt(C, $SK_{i,u}$)

4.2.3. Writing to the Cloud

To write a new information to an already existing file the user must send message during file creation. The cloud verifies W_k(writing key) and only if the user is authenticate is allowed to write on the file.

4.3. Comparison with other access scheme

Comparing this scheme with other access control schemes as shown in Table 1 and shown that this scheme supports many features that the other schemes did not support. 1-W-M-R means that only one user can write while many users can read. M-W-M-R means that many users can write and read. Most schemes do not support multiple writes which is supported by this scheme. This scheme is robust and decentralized in nature, most of the others are centralized in nature.

Schemes	**Centralised/decentralised**	**Write /read Access**	**Types of Access Control**	**Privacy Preserving Authentication**	**Revocation**
[5]	centralised	1-W-M-R	ABE	No authentication	NO
[6]	Decentralised	1-W-M-R	ABE	NO	YES
[7]	centralised	M-R-M-W	ABE	Authentication	NO
This	Decentralised	M-R-M-W	ABE, ABS	Authentication	YES

Table 1: Comparison with other schemes

This scheme also supports privacy preserving authentication, which is not supported by other schemes. Most of the schemes do not support user revocation, which this scheme does. Many operations are done by the cloud. If compare the computation load of user during read see that this scheme has comparable costs.

5. Performance Analysis

5.1. Time Performance

The performance of this paper was analyzed under various file sizes. At first the time performance of this paper is evolved for different file sizes. Then the cryptographic operation time is evolved.

File Size	**Upload(sec)**	**Download(sec)**
10 byte	2	1
1 kb	7	3
10 kb	9	4
50 kb	15	6
100 kb	22	10

Table2: Time Performance for file upload/download process using paillier algorithm

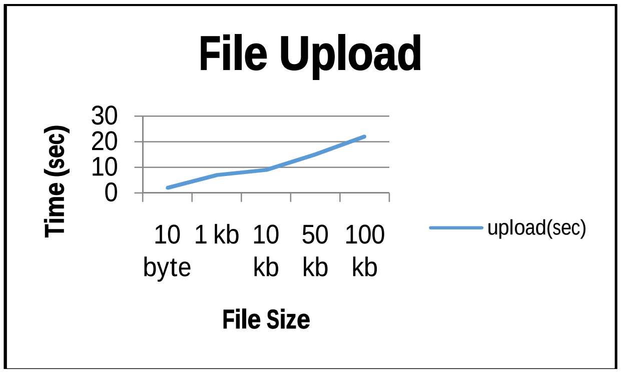

Figure 2: Performance Analysis of File Upload Process

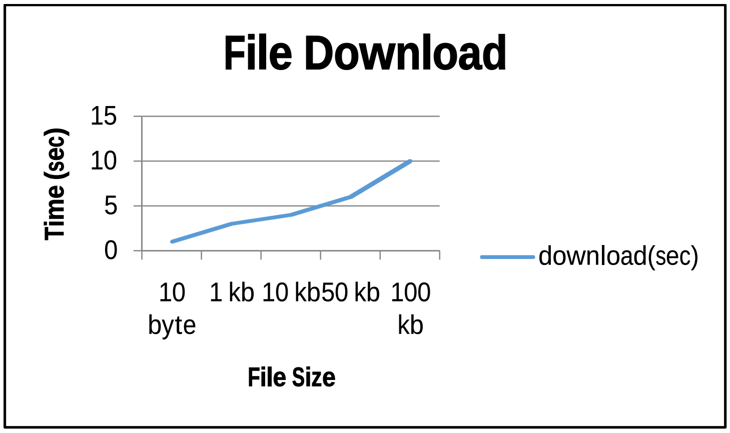

Figure 3: Performance Analysis of File Download Process

6. Conclusion

In this paper security and storage issues are addressed simultaneously. This is a secure cloud storage system using decentralized access control with anonymous authentication. Using this system user can securely upload different files on cloud. Multiple reader, writers are can access data stored in cloud. Revocation is addressed that removes the files of revoked users. The cloud does not know the identity of the user who stores the information. Key distribution is done in decentralized manner using multiple KDC.

7. References

i. 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.

ii. S. Ruj, M. Stojmenovic, A. Nayaks” Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds,” IEEE transactions on parallel and distributed systems, pp-384-394, f 2014

iii. J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute-Based Encryption,” Proc. IEEE Symp. Security and Privacy, pp. 321-334, 2007.

iv. 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.

v. M. Li, S. Yu, K. Ren, and W. Lou, “Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings,” Proc. Sixth Int’l ICST Conf. Security and Privacy in Comm. Networks (SecureComm), pp. 89-106, 2010.

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

vii. F. Zhao, T. Nishide, and K. Sakurai, “Realizing Fine-Grained and Flexible Access Control to Outsourced Data with Attribute-Based Cryptosystems,” Proc. Seventh Int’l Conf. Information Security Practice and Experience (ISPEC), pp. 83-97, 2011.

viii. M. Chase, “Multi-Authority Attribute Based Encryption,” Proc. Fourth Conf. Theory of Cryptography (TCC), pp. 515-534, 2007.

ix. A. Sahai and B. Waters, “Fuzzy Identity-Based Encryption,” Proc. Ann. Int’l Conf. Advances in Cryptology (EUROCRYPT), pp. 457- 473, 2005.

x. C. Gentry, “A Fully Homomorphic Encryption Scheme,” PhD dissertation, Stanford Univ., http://www.crypto.stanford.edu/ craig, 2009