International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.3, 2015

RESEARCH ARTICLE

ISSN: 2321-7758

SCALABLE AND SECURE DATA SHARING IN CLOUD STORAGE USING REVOCABLE DATA ACCESS CONTROL WITH CP-ABE

ANANTHI.A.R[1]., SASIDEVI.J[2]

[1,]PG Student, [2]Professor Department of Computer Science & Engineering
Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu, India

Article Received: 29/04/2015 | Article Revised on:04/05/2015 | Article Accepted on:09/05/2015

ANANTHI.A.R

ABSTRACT

A decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. In Existing system, access control schemes designed for clouds which are centralized. A centralized approach where single key distribution center (KDC) distributes secret keys and attributes to all users.It prevents replay attacks and supports creation, modification, and reading data stored in the cloud, also address user revocation. A decentralized access control using two types protocols are, 1. Attribute Based Encryption (ABE) and 2. Attribute Based Signature (ABS).

KEYWORDS: Cloud Storage, CP-ABE, Revocable Data Access Control, Secure Data, Sharing, attribute-based signatures, attribute-based encryption.

I INTRODUCTION

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without IaaS are virtual servers leased by Amazon, Rackspace, GoGrid, etc..

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 users who are authorized to access data. This is not possible in clouds where there are many users. In RBAC users are classified based on their own roles. Data should be accessed by users who have matching roles. The roles are declare by the system. For an example, only faculty members and senior secretaries might have access to data but not the junior secretaries..

ABAC is more extended in scope, in which users are given attributes, and the data has attached access policy. Only users with valid set of attributes and satisfying the access policy, can access the data. Only when the users have matching set of attributes, they have decrypting the information stored in the cloud.

Third service model is Infrastructure as a Service (IaaS). Cloud infrastructure services or Infrastructure as a Service (IaaS) delivers a computing infrastructure, typically a virtualization environment, as-a-service. Examples of IaaS are virtual servers leased by Amazon, Rackspace, GoGrid, etc.

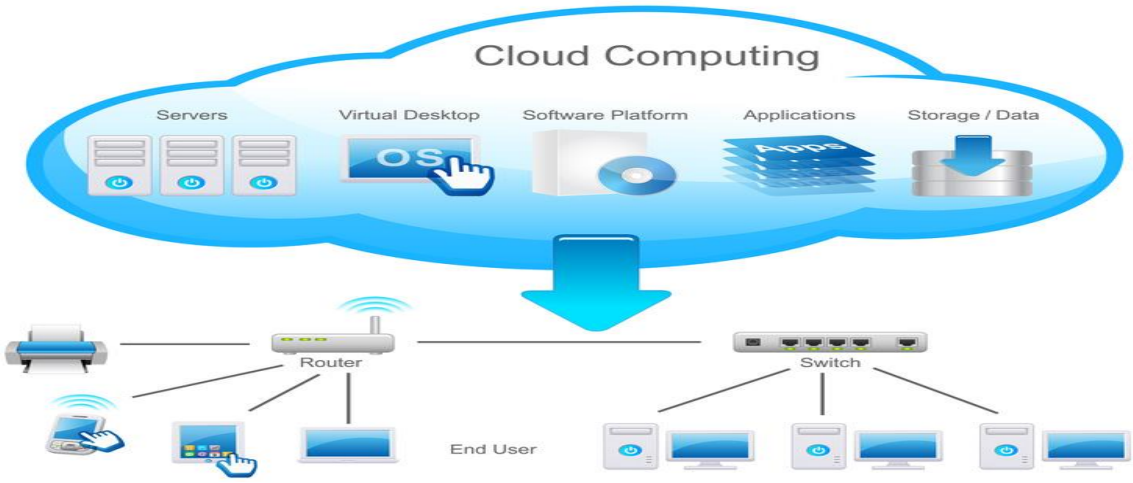

Figure: Cloud Computing

Our contributions in this paper are multirole.

a. To identify whether the user is protected from the cloud during authentication.
b. The architecture is decentralized, meaning that there should be several KDCs for key management.
c. The access control data and authentication are both collusion resistant, that means two users can collude and access data or authenticate themselves, if they are individually not authorized.
d. Revoked users cannot be access the data after they have been revoked.
e. The proposed system is resilient to replay attacks. A writer those attributes and keys have been revoked cannot write back stale information.

II Background

Assumptions:

a. Users can have either read or write or both accesses to a file stored in the cloud.

b. All communications between users/clouds are secured by the secure shell protocol technique, SSH.

Formats of Access Policies:

a. Boolean functions of attributes,

b. Linear secret sharing scheme (LSSS) matrix of the data [1], or

c. Monotone span programs.

Any access structure can be converted into a Boolean function. An example of a Boolean function is ((a1 Λ a2 Λ a3) V (a4 Λ a5)) Λ (a6 V a7)), where a1,a2, . . . , a7 are attributes.

Let Y : {0; 1}n →{0; 1} be a monotone Boolean function.. A monotone span program for Y over a field IF is an l *t matrix M with entries in IF, along with a labeling function a : [l]→[n]that associates each row of M with an input variable of Y, such that, for every (x1, x2, . . . , xn) ε {0 , 1}n.

a. Distributed access control of the data stored in cloud. Only authorized users with valid attributes can access the data.
b. Authentication of users only store data and modify their data on the cloud.
c. The costs are comparable to the existing centralized approaches, it's very expensive operations are mostly done by the cloud.

III RELATED WORK

A) Public key cryptography

Public-key cryptography, also known as **asymmetric cryptography**, is a class of cryptographic algorithms which requires two separate keys, one of which is *secret* (or *private*) and one of which is *public*. Although different, the two parts of this key pair are mathematically linked. The public key is used to encrypt plaintext or to verify a digital signature; whereas the private key is used to decrypt cipher-text or to create a digital signature. The term "asymmetric" stems from the use of different keys to perform these opposite functions, each the inverse of the other – as contrasted with conventional ("symmetric") cryptography which relies on the same key to perform both.

Public-key algorithms are based on mathematical problems which currently admit no

efficient solution that are inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. It is computationally easy for a user to generate their own public and private key-pair and to use them for encryption and decryption. The strength lies in the fact that it is "impossible" (computationally infeasible) for a properly generated private key to be determined from its corresponding public key. Thus the public key may be published without compromising security, whereas the private key must not be revealed to anyone not authorized to read messages or perform digital signatures.

Message authentication involves processing a message with a private key to produce a digital signature. Thereafter anyone can verify this signature by processing the signature value with the signer's corresponding public key and comparing that result with the message. Success confirms the message is unmodified since it was signed, and – presuming the signer's private key has remained secret to the signer – that the signer, and no one else, intentionally performed the signature operation. In practice, typically only a hash or digest of the message, and not the message itself, is encrypted as the signature.

Many homomorphic encryption techniques have been suggested to ensure that the cloud is not able to read the data while performing computations on them. Using homomorphic encryption, the cloud receives ciphertext of the data and performs computations on the ciphertext and returns the encoded value of the result. The user is able to decode the result, but the cloud does not know what data it has operated on. In such circumstances, it must be possible for the user to verify that the cloud returns correct results. Accountability of clouds is a very challenging task and involves technical issues and law enforcement. Neither clouds nor users should deny any operations performed or requested. It is important to have log of the transactions performed; however, it is an important concern to decide.

B) LIMITATIONS

CP-ABE is more appropriate for data access control of cloud storage systems, as users may hold attributes issued by multiple authorities and data owners may also share the data using access policy defined over attributes from different authorities. For example, in an E-health system, data owners may share the data using the access policy ‘‘Doctor AND Researcher’’, where the attribute ‘‘Doctor’’ is issued by a medical organization and the attribute ‘‘Researcher’’ is issued by the administrators of a clinical trial. However, it is difficult to directly apply these multi-authority CP-ABE schemes to multi-authority cloud storage systems because of the attribute revocation problem.

In authority cloud storage systems, users’ attributes can be changed dynamically. A user may be entitled some new attributes or revoked some current attributes. And his permission of data access should be changed accordingly. However, existing attribute revocation methods either rely on a trusted server or lack of efficiency, they are not suitable for dealing with the attribute revocation problem in data access control in authority cloud storage systems

Cloud servers prone to Byzantine failure, where a storage server can fail in arbitrary ways. The cloud is also prone to data modification and server colluding attacks. In server colluding attack, the adversary can compromise storage servers, so that it can modify data files as long as they are internally consistent. To provide secure data storage, the data needs to be encrypted. However, the data is often modified and this dynamic property needs to be taken into account while designing efficient secure storage techniques. Efficient search on encrypted data is also an important concern in clouds. The clouds should not know the query but should be able to return the records that satisfy the query.

This is achieved by means of searchable encryption. The keywords are sent to the cloud encrypted, and the cloud returns the result without knowing the actual keyword for the search. The problem here is that the data records should have keywords associated with them to enable the search. The correct records are returned only when searched with the exact keywords. Security and privacy protection in clouds are being explored by many researchers.

IV SYSTEM MODEL

Here propose our privacy preserving authenticated access control scheme. According to our scheme a user can create a file and store it securely in the cloud. This scheme consists of use of the two protocols ABE and ABS.

We refer to the system Architecture. There are three users, a creator, a reader, and writer. Creator Alice receives a token _ from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc .On presenting her id (like health/social insurance number), the trustee gives her a token. There are multiple KDCs, which can be scattered. For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and signing. In the system Architecture, SKs are secret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X. The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy Y, to prove her authenticity and signs the message under this claim. The ciphertext C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the ciphertext C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message. Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud.

.

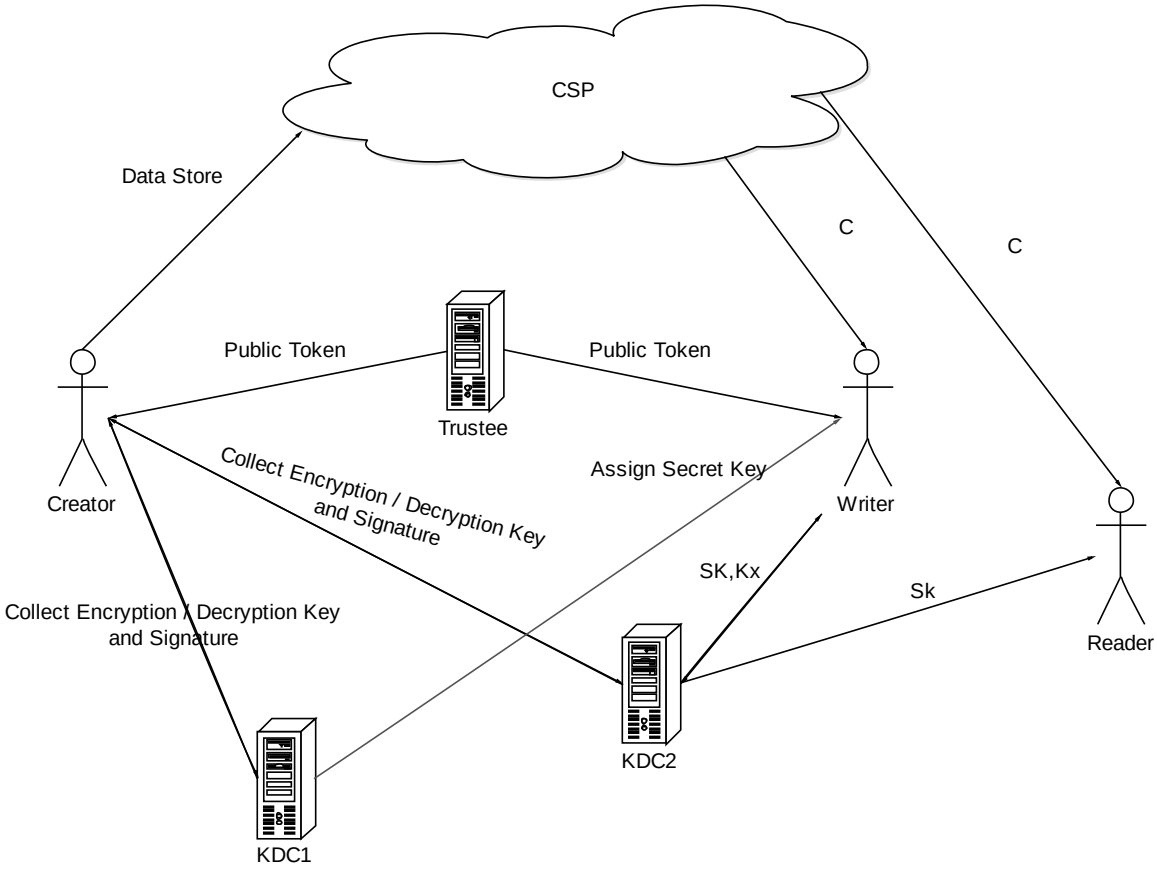

Figure: System Architecture

V PROPOSED SYSTEM DESIGN

In this chapter used to describes the system design how to implement in cloud computing. Here using different phases for the implementation i.e. creator phase, data dynamic phase, read only phase, key distributed center phase, user revocation phase, cloud storage phase. Those are explained below.

A) CREATOR PHASE

In this phase a user Uu first registers itself as a one trustee; the trustee gives it a token and the signature and also signed with the trustees private

key. The user on presenting this token obtains attributes and secret keys from one or more KDCs. After user encrypt the Data using ABE and upload to cloud. Creator to assign writes permission to the anonymous writer to authenticate the writer through Trustee.

B) DATA DYNAMIC PHASE

Data Dynamic Phase used to write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. A user can only write provided the cloud is able to validate its access claim. An invalid user cannot receive attributes from a KDC, if it does not have the credentials from the trustee. If a user's credentials are revoked, then it cannot replace data with previous stale data, thus preventing replay attacks. The Correct authenticated writer will modify the data that process data dynamic these are Insert, Delete, Update Operation will conducted.

C) READ ONLY PHASE

In this phase a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message. Reader it tries to decrypt the data using the secret keys it receives from the KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud.

D) KEY DISTRIBUTED CENTRE PHASE

Key Distributed Centre Phase is used to authenticated user to assign a Trustee token and collect a public and private keys for encryption and decryption of the data. User receives a set of attributes from KDC and corresponding secret key. Note that all keys are delivered to the user securely using the user's public key, such that only that user can decrypt it using its secret key.

E) USER REVOCATION PHASE

In this phase used to ensure that users must not have the ability to access data, even if they possess matching set of attributes. For this reason, the owners should change the stored data and send updated information to other users. The set of attributes possessed by the revoked user is noted and all users change their stored data that have attributes. Revocation involved changing the public and secret keys of the minimal set of attributes which are required to decrypt the data.

F) CLOUD STORAGE PHASE

Cloud Storage phase is used to store data in cloud. Google Cloud Storage allows world-wide storing and retrieval of any amount of data and at any time.

V CONCLUSION

Here 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. This project implements in windows azure cloud (Microsoft Cloud) storage and access through the real time web URL based application. In future we will improve the knowledge to hide the attribute and the Access policies in Decentralized Data Access in secure Cloud. Here proposed a revocable multi-authority CPABE scheme that can support efficient attribute revocation. Construct an effective data access control scheme for multi-authority cloud storage systems and also proved that scheme was provable secure in the random model. The revocable multi-authority CPABE is a promising technique, which can be applied in any remote storage systems.

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

[6]. C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., http:// www. crypto. stanford. edu/craig, 2009.

[7]. A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-Based Cloud Computing," Proc. Third Int'l Conf. Trust and TrustworthyComputing (TRUST), pp. 417-429, 2010.

[8]. R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp. 282-292, 2010.

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

[10]. 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 (Secure Comm.), pp. 89-106, 2010.