IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 11, November 2015.
www.ijiset.com
ISSN 2348 – 7968

Secure Data Storage in Clouds by Using Decentralized Access Control Scheme

N. Suneel Kumar[1], A. NarayanaRao[2] M.Tech., (Ph.D.)

[1] Computer Science and Engineering, Shree Institute of Technical Education (Affiliated to J.N.T.U. Anantapur), Tirupathi, Andhra Pradesh, India.

[2] Associate Professor, HOD. Dept. of Computer Science and Engineering, Shree Institute of Technical Education (Affiliated to J.N.T.U. Anantapur), Tirupathi, Andhra Pradesh, India.

Abstract

This paper focuses on real-world applications of a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. In the proposed scheme, the cloud verifies the authenticity of the series without knowing the user's identity before storing data. Decentralized access control 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. The scheme also address user revocation. Moreover, authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized. The communication, computation, and storage overheads are comparable to centralized approaches.

Keywords: Access Control, Authentication, Attribute-based signatures, Attribute-based Encryption, Cloud storage.

1. Introduction

Cloud computing:

Cloud computing is receiving a lot of attention from both academic and industrial worlds. In cloud computing, users can outsource their computation and storage to servers (also called clouds) using Internet. This free users from the hassles of maintaining resources on-site. Clouds can provide several types of services like applications (e.g., Google Apps, Microsoft online), infra- structures (e.g., Amazon's EC2, Eucalyptus, Nimbus), and platforms to help developers write applications (e.g., Amazon's S3, Windows Azure).

Much of the data stored in clouds is highly sensitive, for example, medical records and social networks. Security and privacy are, thus, 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 cloud can hold the user accountable for the data it outsources, and likewise, the cloud is itself accountable for the services it provides. The validity of the user who stores the data is also verified. Apart from the technical solutions to ensure security and privacy, there is also a need for law enforcement.

Access control in clouds is gaining attention because it is important that only authorized users have access to valid service. A huge amount of information is being stored in the cloud, and much of this is sensitive information. Care should be taken to ensure access control of this sensitive information which can often be related to health, important documents (as in Google Docs or Drop box) or even personal information (as in social networking). There are broadly 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 feasible in clouds where there are many users. In RBAC users are classified based on their individual roles. Data can be accessed by users who have matching roles. The roles are defined by the system. For 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, satisfying the access policy, can access the data.

Access control is also gaining importance in online social networking where users (members) store their personal information, pictures, and videos and share them with selected groups of users or communities they belong to. Access control is very important that only the authorized users are given access to those information. A similar situation arises when data is stored in clouds, for example, in Drop box, and shared with certain groups of people.

ISSN 2348 – 7968

Cloud computing is the use of delivered as a service over a network (typically the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers. The goal of cloud computing is to apply traditional computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer computing resources (hardware and software) that are Internet). The name comes from the common with a user's data, software and end networks of server computers.

The goal of cloud computing is to apply traditional. supercomputing, or high performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games. The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

2. THEORETICAL BACKGROUND

Existing work on access control in cloud are centralized in nature. Except and, all other schemes use ABE. The scheme in uses a symmetric key approach and does not support authentication. The schemes do not support authentication as well. It provides privacy preserving authenticated access control in cloud. However, the authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users.

2.1 Disadvantages

- The scheme in uses asymmetric key approach and does not support authentication.
- Difficult to maintain because of the large number of users that are supported in a cloud.

In this paper, we propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. In the proposed scheme, the cloud verifies the authenticity of the series without knowing the user's identity before storing data. Our 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.

2.1.1 Attribute-Based Signature Scheme

ABS scheme has the following steps:

2.1.1.1 System Initialization

Select a prime q, and groups G1 and G2, which are of order q. We define the mapping ˆe:G1 ×G1 → G2. Let g1, g2 be generators of G1 and hj be generators of G2, for j [tmax], for arbitrary tmax. Let H be a hash function. Let A0 = ha0 0, where a0 Z q is chosen at random. (TSig, TV er) mean TSig is the private key with which a message is signed and TV er is the public key used for verification. The secret key for the trustee is TSK = (a0, TSig) and public key is TPK = (G1, G2, H, g1, A0, h0, h1, htmax, g2, TV er).

2.1.1.2 User Registration

For a user with identity Uu the KDC draws at random Kbase G. Let K0 = K1/a0 base. The following token γ is output γ = (u, Kbase, K0, ρ), where ρ is signature on u||Kbase using the signing key TSig.

2.1.1.3 KDC Setup

We emphasize that clouds should take a decentralized approach while distributing secret keys and attributes to users. It is also quite natural for clouds to have many KDCs in different locations in the world. The architecture is decentralized, meaning that there can be several KDCs for key management.

The token verification algorithm verifies the signature contained in γ using the signature verification key TV er in TPK. This algorithm extracts Kbase from γ using (a, b) from ASK[i] and computes Kx= K1/(a+bx) base , x J[i, u]. The key Kx can be checked for consistency using algorithm ABS.KeyCheck (TPK, APK[i], γ, Kx), which checks ˆe (Kx, AijBxij) = ˆe (Kbase, hj), for all x J[i, u] and j [tmax].

2.1.1.4 Attribute Generation

The token verification algorithm verifies the signature contained in γ using the signature verification

ISSN 2348 – 7968

key TVer in TPK. This algorithm extracts Kbase from γ using (a, b) from ASK[i] and computes Kx=Kbase, x€ J[I, u]. The key Kx can be checked for consistency using algorithm ABS:KeyCheck(TPK,APK[i],γ;Kx).

2.1.1.5 Sign

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 cipher text C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the cipher text 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.

2.1.1.6 Verify

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.

2.1.1.7 Proposed Privacy Preserving Authenticated Access Control Scheme

In this section, we 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 one protocol ABS we will first discuss our scheme in details and then provide a concrete example to demonstrate how it works. We refer to the Fig. 1. 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 (here 2), 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 Fig. 1, 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 , 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.

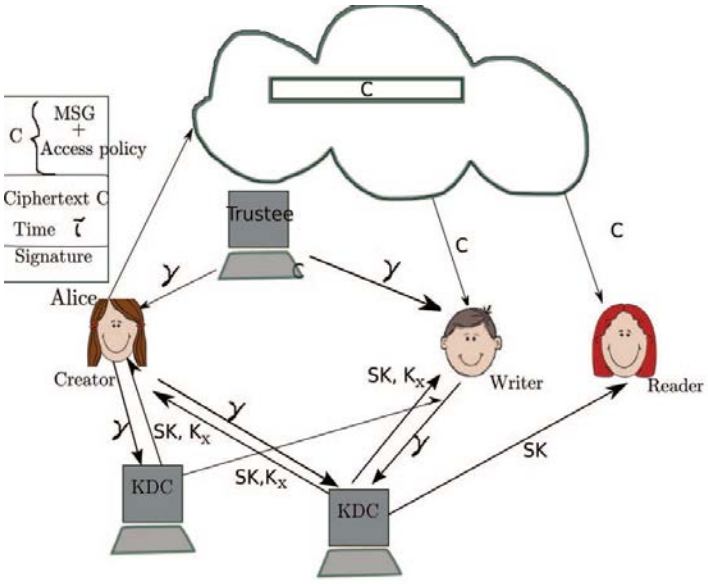

Fig. 1 Secured Cloud Storage Model/Architecture.

2.2 Advantages

1. Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.

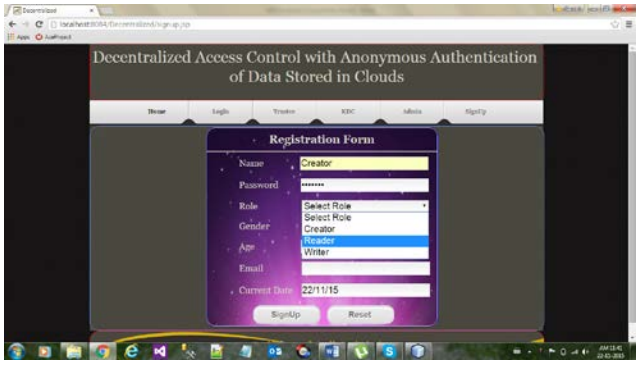

Fig. 2 Creating Access/permissions to the user according to their Roles.

ISSN 2348 – 7968

2. Authentication of users who store and modify their data on the cloud.

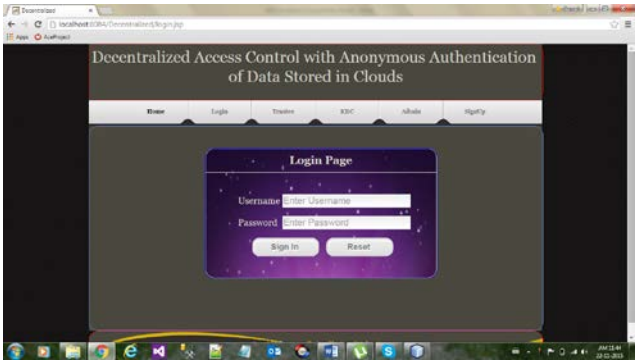

Fig. 3 Authenticate Users login by providing User Credentials.

3. The identity of the user is protected from the cloud during authentication.

Our decentralized access control scheme consists of Attribute-Based Signature Scheme. This paper has the following contributions:

2.3 Data Storage in Clouds

A user Uu first registers itself with one or more trustees. For simplicity we assume there is one trustee. The trustee gives it a token ‘Ù= (u, Kbase, K0, Þ) where _ is the signature on uk, Kbase signed with the trustees private key TSig. The KDCs are given keys PK[i], SK[i]for encryption/ decryption and ASK[i], APK[i] for signing/verifying. The user on presenting this token obtains attributes and secret keys from one or more KDCs. A key for an attribute x belonging to KDC Ai is calculated as Kx = K1base. The user also receives secret keys sk(x, u) for encrypting messages. The user then creates an access policy X which is a monotone Boolean function. The message is then encrypted under the access policy as C = ABE. Encrypt (MSG, XÞ).

2.4 Reading from the Cloud

When a user requests data from the cloud, the cloud sends the ciphertext C using SSH protocol. Decryption proceeds using algorithm ABE. Decrypt (C,{sk (i,u)}).

2.5 Writing to the Cloud

To write to an already existing file, the user must send its message with the claim policy as done during file creation. The cloud verifies the claim policy, and only if the user is authentic, is allowed to write on the file.

2.6 User Revocation

We have just discussed how to prevent replay attacks. We will now discuss how to handle user revocation. It should be ensured 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 Iu possessed by the revoked user Uu is noted and all users change their stored data that have attributes i 2 Iu. In [13], revocation involved changing the public and secret because here different data are encrypted by the same set of attributes, so such a minimal set of attributes is different for different users. Therefore, this does not apply to our model. Once the attributes Iu are identified, all data that possess the attributes are collected.

3. Real World Applications

We now revisit the problem we stated in the introduction. We will use a relaxed setting. Suppose Alice is a law student and wants to send a series of reports about malpractices by authorities of University X to all the professors of University X, Research chairs of universities X; Y; Z and students belonging to Law department in university X. She wants to remain anonymous, while publishing all evidence. All information is stored in the cloud. It is important that users should not be able to know her identity, but must trust that the information is from a valid source. For this reason she also sends a claim message which states that she “Is a law student” or “Is a student counsellor” or “Professor at university X.” The cloud should verify that Alice indeed satisfies this claim. Since she is a law student and is a valid assignment. As a valid user she can then store all the encrypted records under the set of access policy that she has decided. The access policy in case of Alice is later when a valid user, say Bob wants to modify any of these reports he also attaches a set of claims which the cloud verifies. For example, Bob is a research chair and might send a claim “Research chair” or “Department head” which is then verified by the cloud. It then sends the encrypted data to the Bob. Since Bob is a valid user and has matching attributes, he can decrypt and get back the information. If Bob wants to read the contents without modifying them, then there is no need to attach a claim. He will be able to decrypt only if he is a Professor in University X or a Research chair in one of the universities X; Y; Z or a student belonging to Department of Law in university X.

Here it is to be noted that the attributes can belong to several KDCs. For example, the Professors belonging to

ISSN 2348 – 7968

university X have credentials given by the university X, and the Ph.D. degree from a University P, the student counsellor might be a psychologist authorized by the Canadian Psychological Association and assigned an employee number by a university, the research chairs can be jointly appointed by the universities X, Y , Z and the government. The students can have credentials from the university and also a department.

Initially, Alice goes to a trustee, for example, the Canadian health service and presents her a health insurance number or federal agency presents her a social insurance number. Either or both of these trustees can give her token(s) With the token she approaches the KDCs in the university X and department D and obtains the secret keys for decryption and for keys Kx and Ky for signing the assess policy. She can also access the public keys APK[i] of other KDCs.

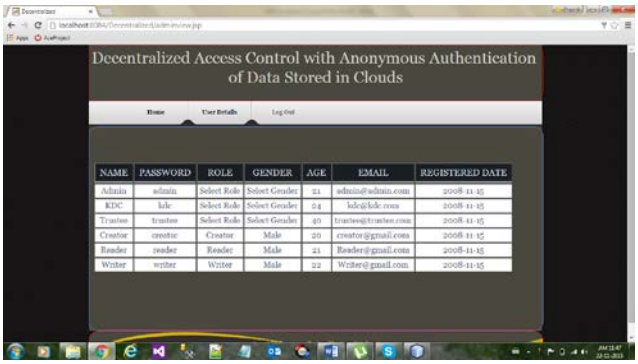

Fig. 4 Admin Role to know User Details/ User Roles.

4. Conclusions

In this paper we briefly present first two of these essential pillars: Here, we studied 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. One limitation is that the cloud knows the access policy for each record stored in the cloud. In future, we would like to hide the attributes and access policy of a user.

Here are the results pages according to the modules in the paper:

Fig. 5 Data Stored in Clouds.

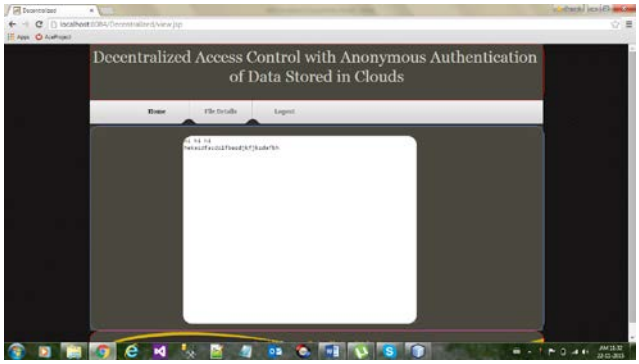

Fig. 6 Reading Data from the Cloud.

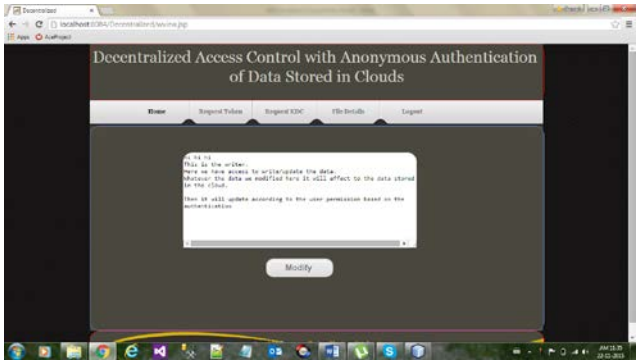

Fig. 7 Writing to the Cloud.

ISSN 2348 – 7968

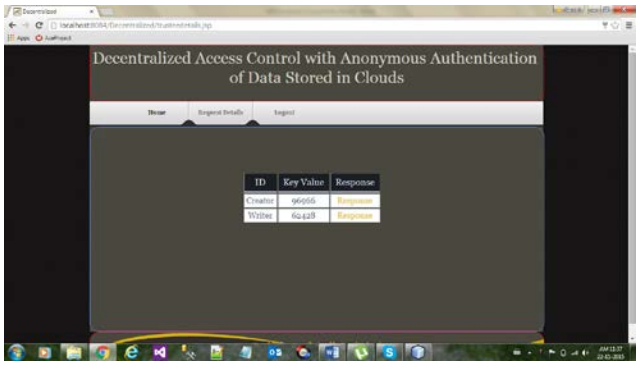

Fig. 8 Trustee Module to generate random Key values.

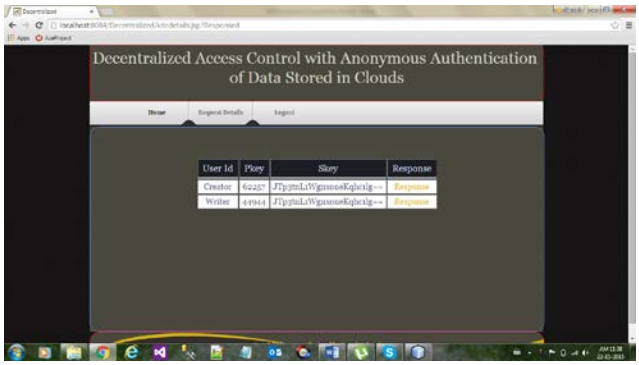

Fig. 9 KDC Module to generate random PKey by Response.

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.

[2] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, “Toward Secure and Dependable Storage Services in Cloud Computing”.

[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy Keyword Search Over Encrypted Data in Cloud Computing.”

[4] H. Li, Y. Dai, L. Tian, and H. Yang, “Identity-Based Authentication for Cloud Computing,” Proc. First Int’l Conf. Cloud Computing (CloudCom).

[5] D.R. Kuhn, E.J. Coyne, and T.R. Weil, “Adding Attributes to Role-Based Access Control.”

[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

N. Suneel Kumar received B.Tech degree in Computer Science and Engineering from the Shree kalahasteeswara institute of Technology affiliated to the Jawaharlal Nehru Technological University Anantapur, in 2012, and M. Tech in Computer Science and Engineering from Shree Institute of Technical Education affiliated to the Jawaharlal Nehru Technological University Anantapur. Interested areas are Cloud Computing and Data Mining. Attended Two National Conferences during 2014 and 2015 and published an International Journal.