International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 3, Issue 11, November 2014 **ISSN 2319 - 4847**

A Secure Decentralized Access Control Scheme for Data stored in Clouds

Priyanka Palekar[1], Abhijeet Bharate[2], Nisar Anjum[3]

[1] SKNSITS, University of Pune

[2]SKNSITS, University of Pune

[3] SKNSITS, University of Pune

ABSTRACT

Cloud computing's multi-tenancy feature, which provides privacy, security and access control challenges, because of sharing of physical resources among untrusted tenants. In order to achieve safe storage, policy based file access control, policy based file assured deletion and policy based renewal of a file stored in a cloud environment, a suitable encryption technique with key management should be applied before outsourcing the data. In this paper a decentralized access control scheme is proposed for secure cloud storage by providing access to the files with the policy based file access using Attribute Based Encryption (ABE) scheme. The proposed scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud. Moreover, the authentication and access control scheme is decentralized and robust, unlike other access control schemes designed for clouds which are centralized.

Keywords:- Access control, authentication, attribute-based encryption, cloud storage.

1. INTRODUCTION

Now a days cloud computing is a rationally developed technology to store data from more than one client. Cloud computing is an environment that enables users to remotely store their data. Remote backup system is the advanced concept which reduces the cost for implementing more memory in an organization. It helps enterprises and government agencies reduce their financial overhead of data management. They can archive their data backups remotely to third party cloud storage providers rather than maintain data centers on their own. An individual or an organization may not require purchasing the needed storage devices. Instead they can store their data backups to the cloud and archive their data to avoid any information loss in case of hardware / software failures. Much of the data stored in clouds is highly sensitive, for example, medical records and social networks. Even cloud storage is more flexible, how the security and privacy are available for the outsourced data becomes a serious concern. 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 Dropbox) or even personal information (as in social networking). It is just not enough to store the contents securely in the cloud but it might also be necessary to ensure anonymity of the user. For example, a user would like to store some sensitive information but does not want to be recognized. The user might want to post a comment on an article, but does not want his/her identity to be disclosed. However, the user should be able to prove to the other users that he/ she is a valid user who stored the information without revealing the identity. There are cryptographic protocols like ring signatures [13], mesh signatures [14], group signatures [15], which can be used in these situations. Ring signature is not a feasible option for clouds where there are a large number of users. Group signatures assume the preexistence of a group which might not be possible in clouds. After comparing the drawbacks of all the cryptographic protocols mentioned above, a new protocol known as attribute-based signature (ABS) has been proposed in this paper. ABS was proposed by author Maji[16]. In ABS, users have a claim predicate associated with a message. The claim predicate helps to identify the user as an authorized one, without revealing its identity. Other users or the cloud can verify the user and the validity of the message stored. ABS can be combined with ABE to achieve authenticated access control without disclosing the identity of the user to the cloud.

2. RELATED WORK

Existing work on access control in cloud are centralized in nature [6], [7], [8], [9], [10], [12], [18]. Except [18] and [12], all other schemes use ABE. The scheme in [18] uses a symmetric key approach and does not support authentication. The schemes [6], [7], [10] do not support authentication as well. Security and privacy protection in clouds are being explored by many researchers. In paper [2], Wang addressed storage security using Reed-Solomon erasure-correcting codes. Authentication of users using public key cryptographic techniques has been studied in [3]. Many homomorphic encryption techniques have been suggested [4], [5] 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. Author Wang, in paper [2] addressed secure and dependable cloud storage. Cloud servers prone to Byzantine failure, where a storage server can fail in arbitrary ways [2]. 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. In paper [9], Zhao 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. Unfortunately, a single KDC is not only a single point of failure but difficult to maintain because of the large number of users that are supported in a cloud environment. Thus, emphasis should be given on that clouds should take a decentralized approach while distributing secret keys and attributes to users. In paper [17], Yang proposed a decentralized approach, their technique does not authenticate users, who want to remain anonymous while accessing the cloud. In an another paper [10], Ruj proposed a distributed access control mechanism in clouds. However, the scheme did not provide user authentication. The other drawback was that a user can create and store a file and other users can only read the file. Write access was not permitted to users other than the creator. In the proposed system, a decentralized architecture is proposed meaning that there can be several KDCs for key management. The main aim of paper is to design a scheme for distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.

3. PROPOSED SYSTEM

Following assumptions are made in the proposed system:

- Users can have either read or write or both accesses to a file stored in the cloud.
- All communications between users/clouds are secured by secure shell protocol, SSH.
- The cloud is honest-but-curious, which means that the cloud administrators can be interested in viewing user's content, but cannot modify it.

The proposed privacy preserving authenticated access control scheme uses two protocols ABE and ABS. These protocols are explained in Sections 4.1 and 4.2, respectively. In the proposed scheme a user can create a file and store it securely in the cloud. The details of the proposed scheme are shown in Figure 1. The detailed description of model is as follows:

- There are three users, a creator, a reader, and writer.
- Creator Alice receives a token γ 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 γ.
- 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 Figure 1, SKs are secret keys given for decryption, K_x are keys for signing. The message MSG is encrypted under the access policy χ.
- 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.

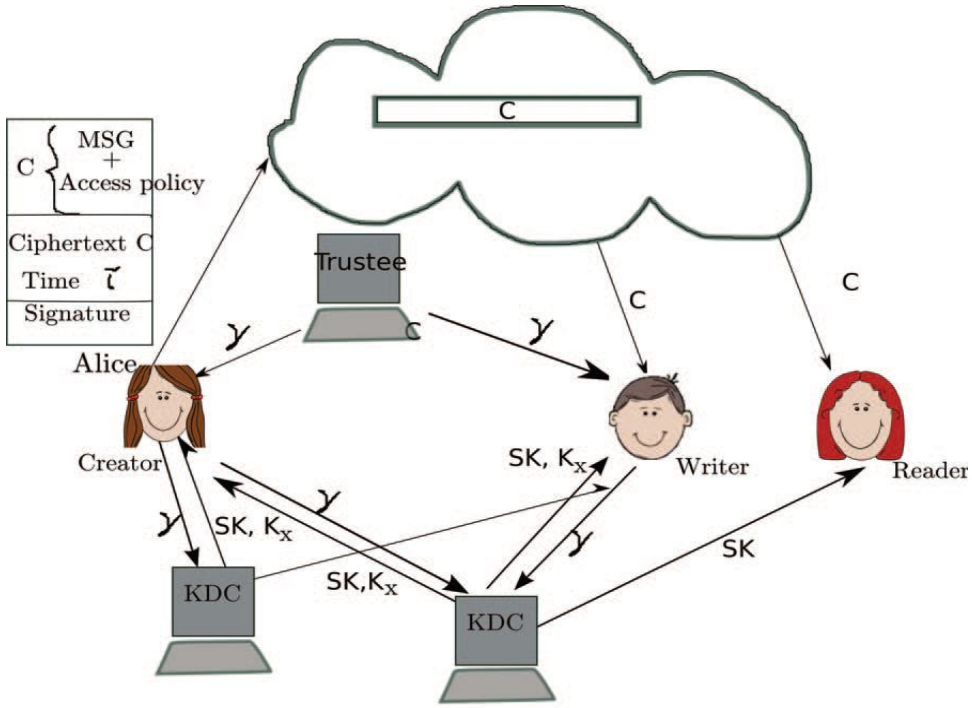

Figure 1. Cloud Storage Model

- 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.

4.ALGORITHMS USED

After surveying the topics related to the proposed scheme, we have found two algorithms i.e. ABE and ABS [19] [20] feasible which will be implemented in Java. As the implementation will be done in Java all the object oriented features and security will be seen in the implementation. This will make the proposed system more efficient.

4.1 Attribute Based Encryption (ABE)

ABE has following steps:

System Initialization

Select a prime q, generator g of G_0, groups G_0 and G_T of order q, a map $e: G_0 \times G_0 \rightarrow G_T$, and a hash function $H: \{0,1\}^* \rightarrow G_0$ that maps the identities of users to G_0. The hash function used here is SHA-1. Each KDC $A_j \in \mathcal{A}$ has a set of attributes L_j. The attributes disjoint ($L_i \bigcap L_j = \phi$ for $i \neq j$). Each KDC also chooses two random exponents $\alpha_i, y_i \in \mathbb{Z}_q$. The secret key of KDC $\mathcal{A}_j$ is

$$SK[j] = \{\alpha_i, y_i, i \in L_j\}. \quad (1)$$

The public key of KDC $\mathcal{A}_j$ is published

$$PK[j] = \{e(g,g)^{\alpha_i}, g^{y_i}, i \in L_j\}. \quad (2)$$

Key Generation and Distribution by KDCs

User U_u receives a set of attributes $I[j,u]$ from KDC $\mathcal{A}_j$, and corresponding secret key $sk_{i,u}$ for each $i \in I[j,u]$

$$sk_{i,u} = g^{\alpha_i} H(u)^{y_i}, \quad (3)$$

where $\alpha_i, y_i \in SK[j]$. 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.

Encryption by Sender

The encryption function is $ABE.Encrypt(MSG, \mathcal{X})$. Sender decides about the access tree $\mathcal{X}$. LSSS matrix R can be derived as described in Section 3.2. Sender encrypts message MSG as follows:

1. Choose a random seed $s \in \mathbb{Z}_q$ and a random vector $v \in \mathbb{Z}_q^h$, with s as its first entry; h is the number of leaves in the access tree (equal to the number of rows in the corresponding matrix R).
2. Calculate $\lambda_x = R_x \cdot v$, where R_x is a row of R.
3. Choose a random vector $w \in \mathbb{Z}_q^h$ with 0 as the first entry.
4. Calculate $\omega_x = R_x \cdot w$.
5. For each row R_x of R, choose a random $\rho_x \in \mathbb{Z}_q$.
6. The following parameters are calculated:

$$\begin{aligned} C_0 &= MSGe(g,g)^s, \\ C_{1,x} &= e(g,g)^{\lambda_x} e(g,g)^{\alpha_{\pi(x)}\rho_x}, \forall x, \\ C_{2,x} &= g^{\rho_x} \forall x, \\ C_{3,x} &= g^{y_{\pi(x)}\rho_x} g^{\omega_x} \forall x, \end{aligned} \tag{4}$$

   where $\pi(x)$ is mapping from R_x to the attribute i that is located at the corresponding leaf of the access tree.
7. The ciphertext C is sent by the sender (it also includes the access tree via R matrix):

$$C = \langle R, \pi, C_0, \{C_{1,x}, C_{2,x}, C_{3,x}, \forall x\} \rangle. \tag{5}$$

Decryption by Receiver

The decryption function is $ABE.Decrypt(C, \{sk_{i,u}\})$, where C is given by (5). Receiver U_u takes as input ciphertext C, secret keys $\{sk_{i,u}\}$, group G_0, and outputs message msg. It obtains the access matrix R and mapping π from C. It then executes the following steps:

1. U_u calculates the set of attributes $\{\pi(x) : x \in X\} \bigcap I_u$ that are common to itself and the access matrix. X is the set of rows of R.
2. For each of these attributes, it checks if there is a subset X' of rows of R, such that the vector $(1, 0, \ldots, 0)$ is their linear combination. If not, decryption is impossible. If yes, it calculates constants $c_x \in \mathbb{Z}_q$, such that $\sum_{x \in X'} c_x R_x = (1, 0, \ldots, 0)$.
3. Decryption proceeds as follows:
   a. For each $x \in X'$, $dec(x) = \frac{C_{1,x} e(H(u), C_{3,x})}{e(sk_{\pi(x),u}, C_{2,x})}$.
   b. U_u computes $MSG = C_0 / \Pi_{x \in X'} dec(x)$.

4.2 Attribute Based Signature

ABS has following steps:

System Initialization

Select a prime q, and groups G_1 and G_2, which are of ord
q. We define the mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Let g_1, g_2
generators of G_1 and h_j be generators of G_2, for $j \in [t_{ma}$
for arbitrary t_{max}. Let $\mathcal{H}$ be a hash function. Let $A_0 = h$
where $a_0 \in \mathbb{Z}_q^*$ is chosen at random. $(TSig, TVer)$ me
$TSig$ is the private key with which a message is signed a
$TVer$ is the public key used for verification. The secret k
for the trustee is $TSK = (a_0, TSig)$ and public key
$TPK = (G_1, G_2, \mathcal{H}, g_1, A_0, h_0, h_1, \ldots, h_{t_{max}}, g_2, TVer)$.

KDC Setup

Choose $a, b \in \mathbb{Z}_q^*$ randomly and compute: $A_{ij} = h_j^a$, $B_{ij} = h_j^b$, for $\mathcal{A}_i \in \mathbb{A}$, $j \in [t_{max}]$. The private key of ith KDC is $ASK[i] = (a, b)$ and public key $APK[i] = (A_{ij}, B_{ij} | j \in [t_{max}])$.

User Registration

For a user with identity U_u the KDC draws at random $K_{base} \in G$. Let $K_0 = K_{base}^{1/\alpha_0}$. The following token γ is output

$$\gamma = (u, K_{base}, K_0, \rho), \quad (6)$$

where ρ is signature on $u \| K_{base}$ using the signing key $TSig$.

Attribute Generation

The token verification algorithm verifies the signature contained in γ using the signature verification key $TVer$ in TPK. This algorithm extracts K_{base} from γ using (a, b) from $ASK[i]$ and computes $K_x = K_{base}^{1/(a+bx)}$, $x \in J[i,u]$. The key K_x can be checked for consistency using algorithm $ABS.KeyCheck(TPK, APK[i], \gamma, K_x)$, which checks

$$\hat{e}(K_x, A_{ij} B_{ij}^x) = \hat{e}(K_{base}, h_j),$$

for all $x \in J[i,u]$ and $j \in [t_{max}]$.

Sign

The algorithm

$$ABS.Sign(TPK, \{APK[i] : i \in AT[u]\}, \gamma, \{K_x : x \in J_u\}, MSG, \mathcal{Y}),$$

has input the public key of the trustee, the secret key of the signer, the message to be signed and the policy claim $\mathcal{Y}$. The policy claim is first converted into the span program $M \in \mathbb{Z}_q^{l \times t}$, with rows labeled with attributes. M_x denotes row x of M. Let π' denote the mapping from rows to the attributes. So, $\pi'(x)$ is the mapping from M_x to attribute x. A vector v is computed that satisfies the assignment $\{x : x \in J[i,u]\}$. Compute $\mu = \mathcal{H}(MSG \| \mathcal{Y})$. Choose $r_0 \in \mathbb{Z}_q^*$ and $r_i \in \mathbb{Z}_q$, $i \in J_u$, and compute:

$$Y = K_{base}^{r_0}, S_i = (K_i^{v_i})^{r_0} \cdot (g_2 g_1^{\mu})^{r_i} (\forall i \in J_u), \quad (7)$$

$$W = K_0^{r_0}, P_j = \Pi_{i \in AT[u]} (A_{ij} B_{ij}^{\pi'(i)})^{M_{ij} r_i} (\forall j \in [t]). \quad (8)$$

The signature is calculated as

$$\sigma = (Y, W, S_1, S_2, \ldots, S_l, P_1, P_2, \ldots, P_t). \quad (9)$$

Verify

Algorithm

$$ABS.Verify(TPK, \sigma = (Y, W, S_1, S_2, \ldots, S_l, P_1, P_2, \ldots, P_t), MSG, \mathcal{Y}),$$

converts $\mathcal{Y}$ to the corresponding monotone program $M \in \mathbb{Z}_q^{l \times t}$, with rows labeled with attributes. Compute $\mu = \mathcal{H}(MSG \| \mathcal{Y})$. If $Y = 1$, $ABS.Verify = 0$ meaning false. Otherwise, the following constraints are checked

$$\hat{e}(W, A_0) \stackrel{?}{=} \hat{e}(Y, h_0), \quad (10)$$

$$\Pi_{i \in I} \hat{e}(S_i, A_{i,j} B_{i,j}^{\pi'(i)})^{M_{ij}} \stackrel{?}{=} \begin{cases} \hat{e}(Y, h_1) \hat{e}(g_2 g_1^{\mu}, P_1), j = 1, \\ \hat{e}(g_2 g_1^{\mu}, P_j), j > 1, \end{cases} \quad (11)$$

where $i' = AT[i]$.

5. CONCLUSION

In this paper, a decentralized access control technique with anonymous authentication is proposed, which provides user revocation and prevents replay attacks. The files are associated with file access policies, that used to access the files placed on the cloud. More security is assured when uploading and downloading of a file to a cloud is performed with standard Encryption/Decryption techniques. 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, the work can be done to hide the attributes and access policy of a user.

REFERENCES

[1] SushmitaRuj, Milos Stojmenovic and Amiya Nayak, "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds", IEEE, 2014.
[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, Apr.- June 2012.
[3] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Computing," Proc. First Int'l Conf. Cloud Computing, 2009.
[4] C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., 2009.
[5] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-Based Cloud Computing," Proc. Third Int'l Conf. Trust and Trustworthy Computing (TRUST), 2010.

[6] 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), 2010.
[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), 2010.
[8] G. Wang, Q. Liu, and J. Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services," Proc. 17th ACM Conf. Computer and Comm. Security (CCS), 2010.
[9] 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), 2011.
[10] 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.
[11] S.SeenuIropia, R.Vijayalakshmi, "Decentralized Access Control Of Data Stored In Clouds Using Key Policy Attribute Based Encryption", International Journal Of Invention In Computer Science And Engineering, 2014.
[12] http://securesoftwaredev.com/2012/08/20/xacml-in-the-cloud, 2013.
[13] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), 2001.
[14] X. Boyen, "Mesh Signatures," Proc. 26th Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), 2007.
[15] D. Chaum and E.V. Heyst, "Group Signatures," Proc. Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), 1991.
[16] H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance," IACR Cryptology ePrint Archive, 2008.
[17] K. Yang, X. Jia, and K. Ren, "DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems," IACR Cryptology ePrint Archive, 2012.
[18] W. Wang, Z. Li, R. Owens, and B. Bhargava, "Secure and Efficient Access to Outsourced Data," Proc. ACM Cloud Computing Security Workshop (CCSW), 2009.
[19] 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.
[20] H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures," Topics in Cryptology - CT-RSA,