ISSN: 2321-7782 (Online)
Volume 2, Issue 11, November 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com
Special Issue: International Conference on Advanced Research Methodology Held by The International Association for Scientific Research & Engineering Technology, India

Anonymous Authentication for Decentralized Access Control of Cloud data

Hemalatha[1]
M.Tech Student
Department of Computer Science and Engineering
Siddharth Institute of Engineering and Technology

V. Balaji[2]
Assistant Professor
Department of Computer Science and Engineering,
Siddharth Institute of Engineering and Technology,

P. Nirupama[3]
Professor & Head
Department of Computer Science and Engineering
Siddharth Institute of Engineering and Technology

Abstract: ***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.***

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. We also address user revocation. Moreover, our 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: Privacy Preserving, Anonymous authentication,

I. INTRODUCTION

Research in cloud computing is receiving a lot of attention from both academic and industrial orlds. In cloud computing, users can outsource their computation and storage to servers (also called clouds) using Internet. This frees 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 comput- ing. 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.

Recently, Wang et al. [2] addressed secure and depend- able 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.

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 [3], [4]. 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. Wang et al. [2] addressed storage security using Reed-Solomon erasure-correcting codes. Authentication of users using public key crypto- graphic techniques has been studied in [5]. Many homo- morphic encryption techniques have been suggested [6], [7] 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 how much information to keep in the log. Accountability has been addressed in TrustCloud [8]. Secure provenance has been studied in [9].

Considering the following situation: A law student, Alice, wants to send a series of reports about some malpractices by authorities of University X to all the professors of University X, research chairs of universities in the country, and students belonging to Law department in all universities in the province. She wants to remain anonymous while publishing all evidence of malpractice. She stores the information in the cloud. Access control is important in such case, so that only authorized users can access the data. It is also important to verify that the information comes from a reliable source. The problems of access control, authentication, and privacy protection should be solved simultaneously. We address this problem in its entirety in this paper.

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). 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 (introduced by Ferraiolo and Kuhn [10]), 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. For instance, in the above example certain records might be accessible by faculty members with more than 10 years of

research experience or by senior secretaries with more than 8 years experience. The pros and cons of RBAC and ABAC are discussed in [11]. There has been some work on ABAC in clouds (for example, [12], [13], [14], [15], [16]). All these work use a cryptographic primitive known as attribute- based encryption (ABE). The eXtensible access control markup language [17] has been proposed for ABAC in clouds [18].

An area where access control is widely being used is health care. Clouds are being used to store sensitive information about patients to enable access to medical professionals, hospital staff, researchers, and policy makers. It is important to control the access of data so that only authorized users can access the data. Using ABE, the records are encrypted under some access policy and stored in the cloud. Users are given sets of attributes and corresponding keys. Only when the users have matching set of attributes, can they decrypt the information stored in the cloud. Access control in health care has been studied in [12] and [13].

Access control is also gaining importance in online social networking where users (members) store their personal information, pictures, videos and share them with selected groups of users or communities they belong to. Access control in online social networking has been studied in [19]. Such data are being stored in clouds. It 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 Dropbox, and shared with certain groups of people.

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 [20], mesh signatures [21], group signatures [22], 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 pre- existence of a group which might not be possible in clouds. Mesh signatures do not ensure if the message is from a single user or many users colluding together.

For these reasons, a new protocol known as attribute-based signature (ABS) has been applied. ABS was proposed by Maji. 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.

Existing work [12], [13], [14], 15], [16], [18], on access control in cloud are centralized in nature. Except [18], all other schemes use ABE. The scheme uses a symmetric key approach and does not support authentica- tion. The schemes [12], [13], [16] do not support authentica- tion as well. Earlier work by Zhao et al. [15] 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.

We, therefore, 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. Although Yang et al. [22] proposed a decentralized approach, their technique does not authenticate users, who want to remain anonymous while accessing the cloud. In an earlier work, Ruj et al. [16] 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 preliminary version of this paper [1],

we extend our previous work with added features that enables to authenticate the validity of the message without revealing the identity of the user who has stored informa- tion in the cloud. In this version we also address user revocation, that was not addressed in [1]. We use ABS scheme to achieve authenticity and privacy. Unlike, our scheme is resistant to replay attacks, in which a user can replace fresh data with stale data from a previous write, even if it no longer has valid claim policy. This is an important property because a user, revoked of its attributes, might no longer be able to write to the cloud. We, therefore, add this extra feature in our scheme and modify appropriately. Our scheme also allows writing multiple times which was not permitted in our earlier work [16].

The paper is organized as follows: Related work is presented in Section 2. background and assumptions are detailed in Section 3. We present our privacy preserving access control scheme in Section 4 The security is analyzed in Section 5. We conclude in Section 6.

II. Related Work

ABE was proposed by Sahai and Waters [17]. In ABE, a user has a set of attributes in addition to its unique ID. There are two classes of ABEs. In key-policy ABE or KP-ABE (Goyal et al. [18]), the sender has an access policy to encrypt data. A writer whose attributes and keys have been revoked cannot write back stale information. The receiver receives attributes and secret keys from the attribute authority and is able to decrypt information if it has matching attributes. In Ciphertext-policy, CP-ABE ([19], [20]), the receiver has the access policy in the form of a tree, with attributes as leaves and monotonic access structure with AND, OR and other threshold gates.

All the approaches take a centralized approach and allow only one KDC, which is a single point of failure. Chase [21] proposed a multiauthority ABE, in which there are several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys to users. Multiauthority ABE protocol was studied in [22], which required no trusted authority which requires every user to have attributes from at all the KDCs. Recently, Lewko and Waters proposed a fully decentralized ABE where users could have zero or more attributes from each authority and did not require a trusted server. In all these cases, decryption at user's end is computation intensive. So, this technique might be inefficient when users access using their mobile devices. To get over this problem, Green proposed to outsource the decryption task to a proxy server, so that the user can compute with minimum resources (for example, hand held devices). However, the presence of one proxy and one KDC makes it less robust than decentralized approaches. Both these approaches had no way to authenticate users, anonymously. Yang presented a modification of, authenticate users, who want to remain anonymous while accessing the cloud.

To ensure anonymous user authentication ABSs were introduced by Maji. This was also a centralized approach. A recent scheme by Maji et al. takes a decentralized approach and provides authentication without disclosing the identity of the users. However, as mentioned earlier in the previous section it is prone to replay attack.

III. Background

We make the following assumptions in our work:

1. The cloud is honest-but-curious, which means that the cloud administrators can be interested in view- ing user's content, but cannot modify it. This is a valid assumption that has been made in [12] and [13]. Honest-but-curious model of adversaries do not tamper with data so that they can keep the system functioning normally and remain undetected.

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

3. All communications between users/clouds are se- cured by secure shell protocol, SSH.

IV. Proposed 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 two protocols ABE and ABS, as discussed in Sections 3, respectively. 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 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.

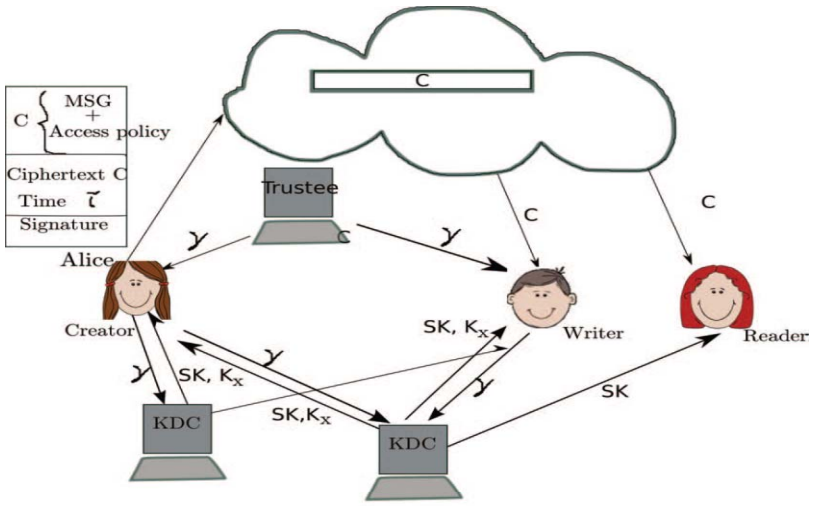

Fig.1 The Secure Proposed model

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.

a) 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 = (u,Kbase,K0), where is the signature on ukKbase signed with the trustees private key TSig (by (6)). 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=K1=ðaþbxÞbase , where (a, b) € ASK[i]. The user also receives secret keys skx;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

$$C = ABE.Encrypt(MSG, \mathcal{X}).$$

The user also constructs a claim policy Y to enable the cloud to authenticate the user. The creator does not send the message MSG as is, but uses the time stamp and creates H(C)||k. This is done to prevent replay attacks. If the time stamp is not sent, then the user can write previous stale message back to the cloud with a valid signature, even when its claim policy and attributes have been revoked. The original work by Maji suffers from replay attacks. In their scheme, a writer can send its message and correct signature even when it no longer has access rights. In our scheme a writer whose rights have been revoked cannot create a new signature with new time stamp and, thus, cannot write back stale information. It then signs the message and calculates the message signature.

b) 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.

c) 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 keys of the minimal set of attributes which are required to decrypt the data. We do not consider this approach 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.

V. SECURITY OF THE PROTOCOL

Theorem 1. Our access control scheme is secure (no outsider or cloud can decrypt ciphertexts), collusion resistant and allows access only to authorized users.

Proof. We first show that no unauthorized user can access data from the cloud. We will first prove the validity of our scheme. A user can decrypt data if and only if it has a matching set of attributes. This follows from the fact that access structure S (and hence matrix R) is constructed if and only if there exists a set of rows X0 in R, and linear constants

We next observe that the cloud cannot decode stored data. This is because it does not possess the secret keys ski;u (by (3)). Even if it colludes with other users, it cannot decrypt data which the users cannot themselves decrypt, because of the above reason (same as collusion of users). The KDCs are located in different servers and are not owned by the cloud. For this reason, even if some (but not all) KDCs are compromised, the cloud cannot decode data.

Theorem 2. Our authentication scheme is correct, collusion secure, resistant to replay attacks, and protects privacy of the user.

Proof. We first note that only valid users registered with the trustee(s) receive attributes and keys from the KDCs. A user's token is Kbase;K0 where is signature on ukKbase with TSig belonging to the trustee. An invalid user with a different user-id cannot create the same signature because it does not know TSig.

VI. CONCLUSION

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. 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.

References

1 S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving AccessControl 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 Trustworthy Computing (TRUST), pp. 417-429, 2010.

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

9 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.

10 [10] D.F. Ferraiolo and D.R. Kuhn, "Role-Based Access Controls," Proc.15th Nat'l Computer Security Conf., 1992.

11 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.

12 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.

13 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), pp. 261-270, 2010.

14 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), pp. 735-737, 2010.

15 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.

16 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.

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

18 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.

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

20 X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp 343-352, 2009.

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

22 H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure Threshold Multi- Authority Attribute Based Encryption without a Central Authority,"Proc. Progress in Cryptology Conf. (INDOCRYPT), pp. 426-436,2008.

