

Decentralized Access Control scheme for secure Data Stored in Clouds

1Antorose,2Sharmila,3Vijayavahini

1dazzlingrose.ar@gmail.com,2sharmila11694@gmail.com,3vijivahini28@gmail.com

1Assistant professor,CSE Department,2,3,IVyear,CSE Department

Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College

ABSTRACT:

We propose a new decentralized access control scheme which is used for secure the data storage in clouds. In the proposed scheme, the cloud verifies the authenticity without knowing the user's identity before storing data. Our scheme is also added the 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: Revocation, Authentication, Attribute-based signatures, Attribute-based encryption, Cloud storage.

INTRODUCTION:

Research in 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 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), infrastructures (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.

Recently, Wang *et al.* [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 security. 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.

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 necessary 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 solve 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 (ACL) 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 [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 The extensible Access Control Markup Language (XACML) [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], [13].

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 Drop box, 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 A.

Our Contributions

The main contributions of this paper are the following:

1. Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.
2. Authentication of users who store and modify their data on the cloud.
3. The identity of the user is protected from the cloud during authentication.
- 4) The architecture is decentralized, meaning that there can be several KDCs for key management.
- 5) The access control and authentication are both collusion resistant, meaning that no two users can collude and access the data or authenticate themselves, if they are individually not authorized.
- 6) Revoked users cannot access data after they have been revoked.
- 7) The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information.
- 8) The protocol supports multiple read and write on the data stored in the cloud.
- 9) The costs are comparable to the existing centralized approaches, and the expensive operations are mostly done by the cloud.

The message is from a single user or many users colluding together. For these reasons, a new protocol known as Attribute 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 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.* [14] proposed a decentralized approach, their technique does not authenticate users, who want to remain anonymous while accessing the cloud. In an earlier work, Rujet *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 which enables to authenticate the validity of the message without revealing the identity of the user who has stored information in the cloud. In this version we also address user revocation that was not addressed in [1]. We use attribute based signature scheme [12] to achieve authenticity and privacy. 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 access control without disclosing the identity of the user to the cloud.

Existing work [18], [12], [13], [14], [15], [16] on access control in cloud are centralized in nature. Except [6] and [18], all other schemes use attribute based encryption (ABE). The scheme in uses a symmetric key approach and does not support authentication. The schemes [12], [13], [16] do not support authentication as well. Earlier work by Zhao *et al.* [15] provides privacy

preserving authenticated access control in cloud. its attributes, might no longer be able to write to the cloud. We therefore add this extra feature in our scheme and modify [24] appropriately. Our scheme also allows writing multiple times which was not permitted in our earlier work

RELATED WORK:

There are two classes of ABEs. In Key-policy ABE or KP-ABE (Goyal *et al.* [27]), 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. ([28], [29]), the receiver has the access policy in the form of a tree, with attributes was also a centralized approach. A recent scheme by the same authors [12] 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 who want to remain anonymous while accessing the cloud.

To ensure anonymous user authentication Attribute Based Signatures were introduced by Maji.

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, Greenet *et al.* [33] 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 key distribution center makes it less robust than decentralized approaches? Both these approaches had no way to

BACKGROUND:

In this section, we present our cloud storage model, adversary model and the assumptions we have made in the paper. Table I presents the notations used throughout the paper. We also describe mathematical background used in our proposed solution.

Formats of access policies:

Access policies can be in any of the following formats: 1) Boolean functions of attributes, 2) Linear Secret Sharing Scheme (LSSS) matrix, or 3) Monotone span programs. Any access structure can be converted into a Boolean function.

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 adversaries do not tamper with data so that they can keep the system functioning normally and remain undetected. Users can have either read or write or both accesses to a file stored in the cloud.

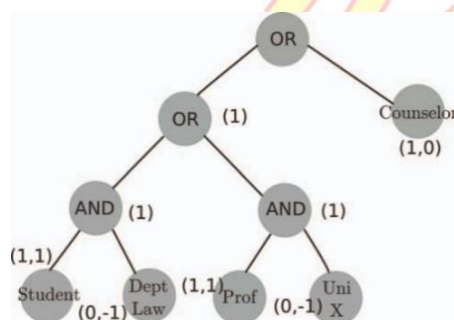


Fig: Examples of claim policy

A. Data storage in clouds:

A user U_u first registers itself with one or more trustees. For simplicity we assume there is one trustee. $C = \text{ABE.Encrypt}(\text{MSG}, X)$ (12) The user also constructs a

claim policy Y to enable the cloud to authenticate the user.

The cloud on receiving the information verifies the access claim using the algorithm ABS.verify . The creator checks the value of $V = \text{ABS.verify}(TP, K, \sigma, c, Y)$. If $V = 0$, then authentication has failed and the message is discarded. Else, the message (C, τ) is stored in the cloud.

B. Reading from the cloud:

When a user requests data from the cloud, the cloud sends the cipher text C using SSH protocol. Decryption proceeds using algorithm.

C. Writing to the cloud:

All communications between users/clouds are secured by Secure Shell Protocol, SSH.

A file and store it securely in the cloud. This scheme consists of use of the two protocols ABE and ABS, as discussed in Section III-D and III-E respectively. We will first discuss our scheme Content, but cannot modify it. This is a valid assumption that has been made in [12], [13]. 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 I_u possessed by the revoked user U_u is noted and all users change their information the data are to be stored in the cloud the cloud provides more security to the user.

The cloud is honest-but-curious, which means that the cloud administrators can be interested in viewing users.

SECURITY OF THE PROTOCOL:

In this section we will prove the security of the protocol. We will show that our scheme authenticates a user who wants to write to the Cloud.

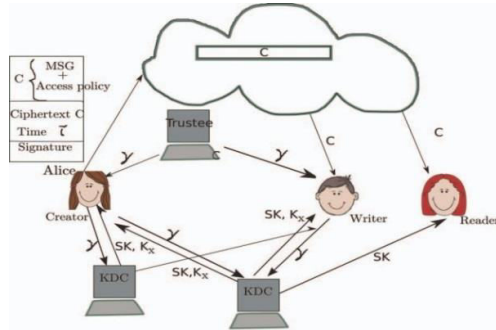


Fig: secure data transmission in cloud

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 attack a University P, the student counselor 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 We will compare our computation costs with existing schemes like [13], [12], [15]

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 Access Control with Authentication for Securing Data in Clouds", *IEEE/ACM International Symposium on 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 T. Services Computing*, vol. 5, no. 2, pp. 220–232, 2012.
- [3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *IEEE INFOCOM*, pp. 441–445, 2010.
- [4] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Financial Cryptography Workshops*, ser. Lecture Notes in Computer Science, vol. 6054. Springer, pp. 136–149, 2010.
- [5] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing," in *CloudCom*, ser. Lecture Notes in Computer Science, vol. 5931. Springer, pp. 157–166, 2009.
- [6] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009, <http://www.crypto.stanford.edu/craig>.
- [7] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-based cloud computing," in *TRUST*, ser. Lecture Notes in Computer Science, vol. 6101. Springer, 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 framework for accountability and trust in cloud computing," HP Technical Report HPL-2011-38. Available at <http://www.hpl.hp.com/techreports/2011/HPL-2011-38.html>.
- [9] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," in *ACM ASIACCS*, pp. 282–292, 2010.
- [10] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," in *15th National Computer Security Conference*, 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, 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," in *SecureComm*, pp. 89–106, 2010.
- [13] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *ACM 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," in *ACM 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," in *ISPEC*, ser. Lecture Notes in Computer Science, vol. 6672. Springer, pp. 83–97, 2011.
- [16] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in *IEEE TrustCom*, 2011.
- [17] <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf>.
- [18]<http://seuresoftwaredev.com/2012/08/20/xacml-in-the-cloud>.
- [19] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-based access control in social networks with efficient revocation," in *ACM ASIACCS*, 2011.
- [20] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *ASIACRYPT*, ser. Lecture Notes in Computer Science, vol. 2248. Springer, pp. 552–565, 2001.