

Decentralized authentication of data stored in clouds

¹Besta Chandrakala & ²Raja kumari

Abstract —

In this paper, we propose the secure data storage in clouds for a new decentralized access. The cloud verifies the authenticity of the series without knowing the user's identity in the proposed scheme. Our feature is that only valid users can able to decrypt the stored information. It prevents from the replay attack. This scheme supports creation, modification, and reading the data stored in the cloud and also provide the decentralized authentication and robust. It can be comparable to centralized schemes for the communication of data, computation of data, and storage of data.

Keywords— Access control; authentication of user; attribute-based signature; attribute-based encryption; cloud storage

INTRODUCTION Clouds can provide many types of services like applications (e.g., Google Apps, Microsoft online) ,infrastructures (e.g., Amazon's EC2, Eucalyptu, Nimbus), and platforms to help developers write applications (e.g., Amazon's S3, Windows Azure).The data stored in clouds is highly sensitive, for example, medical records and social networks. The user validity is who stores the data is also verified. The cloud is also prone that modification of data and server colluding attacks. The data needs to be encrypted means to provide secure data storage. Newly, Wang et al. [2] addressed secure and dependable cloud storage. The clouds should not know the query but should be able to return the records that satisfy the query with security and privacy protection in clouds by using a encryption [3][4]. The user is able to decoding the result, but the cloud does not know what data it has operated on. In such cases, it should be possible for the user to verify that the cloud returns correct data. Access control is essential when unauthorized users tries to access the data from the storage, so that only authorized users can access the data. It is also significant to verify that the information

comes from a reliable source. We need to solve the problems of access control, authentication, and privacy protection by applying suitable encryption techniques given in [5] [6] [7]. There are three types of access control: user-based access control(UBAC), role-based access control (RBAC), and attribute-based access control (ABAC). In UBAC, the access control list contains the list of users who are authorized to access data. This is not possible in clouds where there are many users. In RBAC users are classified based on their own roles. Data should be accessed by users who have matching roles. The roles are declare by the system. For an example, only faculty members and senior secretaries might have access to data but not the junior secretaries. ABAC is more extended in scope, in which users are given attributes, and the data has attached access policy. Only users with valid set of attributes and satisfying the access policy, can access the data. Only when the users have matching set of attributes, they have decrypting the information stored in the cloud. The merits and demerits of RBAC and ABAC are discussed in [7]. There has been some related

work on ABAC in clouds for authentication (for example, [8], [9], [10], [11]). Our contributions in this paper are multirole. a. To identify whether the user is protected from the cloud during authentication. b. The architecture is decentralized, meaning that there should be several KDCs for key management. c. The access control data and authentication are both collusion resistant, that means two users can collude and access data or authenticate themselves, if they are individually not authorized. d. Revoked users cannot be access the data after they have been revoked. e. The proposed system is resilient to replay attacks. A writer those attributes and keys have been revoked cannot write back stale information.

f. The protocol has supported multiple read and write on the data stored in the cloud.

RELATED WORK The authors [12] take a centralized technique where a single key distribution center (KDC) distributes secret keys and attributes to all the users. Unfortunately, a single KDC is not only a single data of failure but difficult to maintain because of the large number of users that are supported in a cloud environment. The receiver receiving the attributes and secret keys from the attribute authority and is able to decrypt the information if it has matching attributes. All the technique take a centralized approach and allow only one KDC, which is a single point of failure. Chase [13] proposed a scheme in which there are several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys of the users. However, the presence of one proxy and one KDC makes it less robust than decentralized approach. A new scheme given by Maji et al. takes a decentralized approach and provides authentication without disclosing the identity of the users.

BACKGROUND Assumptions: a. Users can have either read or write or both accesses to a file stored in the cloud. b. All communications between users/clouds are secured by the secure shell protocol technique, SSH. Formats of Access Policies: a. Boolean functions of attributes, b. Linear secret sharing scheme (LSSS) matrix of the data [1], or c. Monotone span programs. Any access structure can be converted into a Boolean function. An example of a Boolean function is $((a_1 \wedge a_2 \wedge a_3) \vee (a_4 \wedge a_5)) \wedge (a_6 \vee a_7)$, where a_1, a_2, \dots, a_7 are attributes. Let $Y : \{0, 1\}^n \rightarrow \{0, 1\}$ be a monotone Boolean function.. A monotone span program for Y over a field IF is an $l \times t$ matrix M with entries in IF , along with a labeling function $a : [l] \rightarrow [n]$ that associates each row of M with an input variable of Y , such that, for every $(x_1, x_2, \dots, x_n) \in \{0, 1\}^n$. a. Distributed access control of the data stored in cloud. Only authorized users with valid attributes can access the data. b. Authentication of users only store data and modify their data on the cloud. c. The costs are comparable to the existing centralized There are three following users, a creator, a reader, and a writer. Creator Alice receives a token γ from the trustee, now it is assumed to be who is honest. 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 define a claim policy Y to prove the authenticity and signs of the message under this claim. The ciphertext C with a signature c is sent to the cloud. The cloud verifies the signature and stores the ciphertext C . When a reader wants to read the message in the cloud sends C . That the user has attributes matching with the access policy, it can be decrypted and get back the original message. Write also proceeds in the similar way as file creation. By designating

the verification of the data 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 decrypting and 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. Data Storage in Clouds: A user U_u have one or more trustees. This is used to prevent to the replay attacks. In this time data is not sent, then the user can write previous stale message back to the cloud with a valuable signature, even when its claim policy and attributes have been revoked. Reading from the Cloud: The user requests data from the cloud, the cloud sends the ciphertext using SSH protocol. Decryption proceeds using algorithm ABE. Writing to the Cloud: 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. User Revocation: It should be ensured that users must not have the ability to access data, even if they possess matching set of attributes.

SECURITY OF THE PROTOCOL We will explain that our scheme authenticates a user who wants to write to the cloud. A user should only write provided the cloud is able to validate it access to the claim. An invalid user cannot receive the attributes from a KDC, if it do not have the credentials from the trustee. If a user's credentials are revoked, then it cannot replace data with previous data, thus preventing replay attacks. Theorem 1. Our access control scheme is secure, collusion resistant and allows access only to authorized users. Theorem 2. Our authentication data is correct, collusion secure, resistant to the replay of attacks, and protects privacy of the user. Next we confirm that only a valid user with valid access claim is only able to store the message in

the cloud. This is taken from the functions given in [24]. A user who wants to create a file and tries to make a wrong access claim, cannot do so, since it will not have attribute keys K_x from the related KDCs. Since the message is encrypted, a user without valid access policy cannot decrypt and change the information

CONCLUSION The conclusion of the paper is to present a decentralized access control technique with anonymous authentication. Its provides user revocation and prevents to the replay attacks. The cloud do not know the identity of the user who store the information, but one and only verifies the user's credentials.

REFERENCES

- [1]. S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds," Proc. IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing, pp. 556- 563, 2012.
- [2]. C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.- June 2012.
- [3]. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM, pp. 441- 445, 2010.
- [4]. S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptography and Data Security, pp. 136- 149, 2010.
- [5]. H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud



- Computing,” Proc. First Int’l Conf. Cloud Computing (CloudCom), pp. 157-166, 2009.
- [6]. C. Gentry, “A Fully Homomorphic Encryption Scheme,” PhD dissertation, Stanford Univ, <http://www.crypto.stanford.edu/craig>, 2009.
- [7]. 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.
- [8]. 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.
- [9]. 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.
- [10]. G. Wang, Q. Liu, and J. Wu, “Hierarchical AttributeBased Encryption for Fine-Grained Access Control in Cloud Storage Services,” Proc. 17th ACM Conf. Computer and Comm. Security (CCS), pp. 735-737, 2010.
- [11]. F. Zhao, T. Nishide, and K. Sakurai, “Realizing FineGrained 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.
- [12]. A. Sahai and B. Waters, “Fuzzy Identity-Based Encryption,” Proc. Ann. Int’l Conf. Advances in Cryptology (EUROCRYPT), pp. 457-473, 2005.
- [13]. M. Chase, “Multi-Authority Attribute Based Encryption,” Proc. Fourth Conf. Theory of cryptography (TCC), pp. 515-534, 2007.