Volume 5, Issue 3, March 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Identity Based Authentication for Data Stored in Cloud

R. Vishnu Sekhar, N. Nandini, D. Bhanumathy, M. Hemalatha
CSE, Pondicherry University,
India

Abstract: ***We propose a new identity and authentication of data stored in cloud scheme for secure data storage in clouds that support anonymous authentication. The main aim of our project is to secure and protect the data which come under the property of users. Our scheme also added feature of access control in which only users who are valid are able to decrypt the stored information. Cloud data security is a major concern for the client while using the cloud services which are provided by the service provider. An efficient and secure dynamic auditing scheme is desired to convince data owners that the data is correctly stored in the cloud. There can be some security issues between the client and the service provider. To resolve this issue, a TP (third party) can be used as an auditor.***

Keywords: ***Authentication, Third party audit, cloud storage, cloud service provider, Access control.***

I. INTRODUCTION

Cloud computing is a recently developed computing terminology based on utility and consumption of computing resources. Cloud computing involves for deploying groups of remote servers and software networks that allow centralized data storage and online access to computer services and resources cloud can be classified as private cloud, public cloud and hybrid cloud. Cloud computing rely on sharing of resources to achieve coherence and economies of scale. Users access cloud computing using networked client devices such as desktop, laptops, and smartphones etc. some of these devices cloud clients relies on cloud computing for all or a majority of their applications so as to be basically useless without it. Browser based chromebook is the examples of thin clients. Many cloud applications do not require specific software on the client and use a browser to interact with cloud application. There are three models in cloud computing, deployment models and service models. Public cloud, private cloud and hybrid clouds are comes under the deployment models. Platform as a service, infrastructure as a service, application as a service are comes under the service models.

Private cloud is operated only for a single organization, it can be managed by third party and hosted either internally or externally. When the services are rendered over a network that is open for public use is called as public cloud. Public cloud resources may be offered on a pay per usage model. The hybrid cloud is combination of two or more cloud it may be public or community, private. The important issues in cloud computing are security and privacy. Cloud computing posses privacy concerns because the service provider can access the resources that is on the cloud at any time. Solutions to privacy contains policy and legislation as well as end users for how data is stored.

While sharing IT infrastructure in cloud computing is most efficient and provides flexibility for the clients [1]. In recent system adding and removing a user and to prevent forward secrecy and backward secrecy. The keys collaborate with attributes must be changed and the files must be re-encrypted also the new keys must be re-distributed [4].

Now IT infrastructure is under the control of the cloud provider, the user has not only to trust the security mechanisms and also configuration of the cloud provider, but also the cloud service provider itself [1]. Secure outsourcing of arbitrary computation and data storage is especially difficult to fulfill if a cloud client does not trust the cloud provider at all. There are the proposals for cryptographic methods. It has allow to perform particular computations on encrypted data [5].

There are three types of access control: user based access control(UBAC), role based access control (RBAC) and attribute baswesd access control (ABAC). The access control list which contains the list of users who are authorized to access data is called user based access control. It is not feasible in clouds because there are many users. In role based access control Users are classified based on their own roles. The user who have matching roles can be able to access the data. The system defined the role. In attribute based access control in which users are given attributes and the data has attached access policy. The user can access the data who have valid set of attributes and also satisfying the access policy. Records are encrypted under some access policy such as attribute based encryption and stored in the cloud. Trusted platform module(TPM) cannot perform arbitrary secure computation on data, it has protect cryptographic keys and perform the pre defined cryptographic operations like encryption, decryption[5]. Cloud storage enables users to store their data remotely and enjoy the on-demand high quality cloud applications without the burden of hardware and software management [3].

Cipher text-attribute based encryption(CP-ABE) is the attribute based cryptosystems offers a way to encrypt a file for multiple users according to their privileges [4]. Data owner is the user who wants to outsource their data into the cloud and also responsible for encrypting files and generating access structure policies [4].

User can check the data integrity based on two-party storage auditing protocols. In cloud storage system, it is improper to let either side of cloud service providers or user conduct such auditing, because no one could be guaranteed to provide unbiased auditing result. For this situation, *third party auditing* is a choice for the storage auditing in cloud computing. A third party auditor (auditor) that has expertise and capabilities can do a more efficient work and convince both cloud service providers and user. For the third party auditing in cloud storage systems, it has several important requirements which have been proposed in some previous works. The auditing protocol should have the following properties:

1. *Confidentiality:* The auditing protocol should keep user's data confidential against the auditor.
2. *Dynamic Auditing:* The auditing protocol should support the dynamic updates of the data which is stored in the cloud.
3. *Batch Auditing:* The auditing protocol should also be able to support the batch auditing for multiple user and multiple clouds.

The authors had propose a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers, but this method may be leak the data content to the auditor because it requires the server to send the linear combinations of data blocks to the auditor. The authors extended their dynamic auditing scheme to be privacy-preserving and support the batch auditing for multiple users.

The auditing protocol lets the server computes the proof as an intermediate value of the verification, the auditor can directly use this intermediate value to verify the correctness of the proof. It can greatly reduce the computing loads of the auditor by moving it to the cloud server[7].

User revocation is a challenge issue in one-to-many communication systems. In attribute based systems, this issue is more difficult since each attribute is conceivably shared by multiple users. Revocation of any one owner(user) would affect others who share his attributes in the cloud. Moreover, user revocation in attribute based systems has flexible and occur in different granularities[6].

A. Our contributions

1. Valid users who store and modify their data on the cloud.
2. During authentication, the identity of user is protected from the cloud.
3. It has the costs when compared to the existing centralized approaches, and the expensive operations are executed by the cloud.
4. The access control and authentication are both collusion resistant, it means that no two users can collude and access data or authenticate themselves, if they are not individually authorized.
5. Only valid users with valid set of attributes can access the data stored in cloud.

II. RELATED WORKS

In ABE, the user has a set of attributes in addition to its individual ID. There are two classes of ABEs. In key-policy ABE (KP-ABE), the sender has an access policy to encrypt data. The writer whose attributes and keys which has been revoked cannot write back stale information. The receiver has receive attributes and secret keys from the attribute authority and it is able to decrypt information if it has matching attributes. In Ciphertext-policy, CP-ABE , the receiver has the access policy in the form of a tree, where 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.We proposed a multiauthority ABE ,in which several KDC authorities which distribute attributes and secret keys to users. Multiauthority ABE protocol, which required no trusted authority which requires every user to have attributes from at all the KDCs. Recently, Lewko and Waters has proposed a fully decentralized ABE where users could have zero or more attributes from each authority and did not require a trusted server. In this situation, decryption at user's end is computation intensive. So, this technique may be inefficient when users access using their mobile devices. To get over this problem, Green had propose to outsource the decryption task to a proxy server, so the user can compute with minimum resources. However, the presence of single proxy and one KDC makes it less robust than decentralized approaches. Both approaches had no way to authenticate users, anonymously. Authenticate users, who need to remain anonymous while accessing the cloud.

III. ALGORITHM DEFINITION

Our proposed scheme is composed of 7 algorithms, they are Setup, Enc, KeyGen, ReKeyGen, ReEnc, ReKey, and Dec. The algorithm which are Setup, KeyGen, and ReKeyGen are performed by the authority while ReEnc and ReKey are executed by proxy servers. Enc and Dec are called by encryptors and decryptors re-spectively. Note that, in our scheme we define a system wide version information ver indicating the evolution of the system master key as follows: initially it is set to 1; when-ever an attribute revocation event occurs and the system master key is redefined, it increases by 1. The system public key, ciphertexts, user secret keys, and proxy rekey's are all tagged with the version information indicating which version of system master key they comply with.

Setup(1^{λ}) It takes as input the security parameter 1^{λ} and outputs the system master key MK and public parameters P. ver is initialized as 1.

Enc(M, S, P) It takes as input a message M , an access structure S, and current public parameters P, and outputs a cipher

text C .

KeyGen(MK, A) It takes as input current system master key MK and a set of attributes A that describes the key. It outputs a user secret key SK within variety of (ver, A, D, $\bar{}$D = $\{D_i, F_i\}_{i\in S}$).

ReKeyGen(γ, MK) It takes as input an attribute set γ that includes attributes for update, and current master key MK. It outputs the new master key MK′, the new public key P′ (computation of P′ can be delegated to proxy servers), and a set of proxy re-key's r for all the attributes in the attribute universe U . ver is increased by 1. Note that, for attributes in set U −γ, their proxy re-key's are set as 1 in r.

ReEnc(C, r, β) It takes as input a ciphertext C , the set of proxy re-key's r having the same version with C, a set of attributes β which includes all the attributes in C's access structure with proxy re-key not being 1 in r. It outputs a re-encrypted ciphertext C′ with the same access structure as C .

ReKey (D, r, θ) It takes as input the component D of a user secret key SK, the set of proxy re-key's r having the same version with SK, and a collection of attributes θ which has include all the attributes in SK with proxy re-key not being 1 in rk. It outputs updated user secret key components $\bar{}$D′.

Dec(C, P, SK) It takes as input a ciphertext C, public parameters P, and the user secret key SK having the same version with C . It outputs the message M if the attribute set of SK satisfies the ciphertext access structure.

IV. PROPOSED DECENTRALIZED ACCESS CONTROL WITH ANONYMOUS AUTHENTICATION FOR DATA STORED IN CLOUD

In this section, we propose our decentralized access control with anonymous authentication for data stored in cloud. According to our scheme a owner can create a file and store it securely in the cloud. This theme consists of use of the two protocols CP-ABE and auditing. We'll first discuss our scheme in details and so offer a example to demonstrate how it has work. We tend to visit the Fig. 1. There are 3 users, a creator, a reader, and writer.

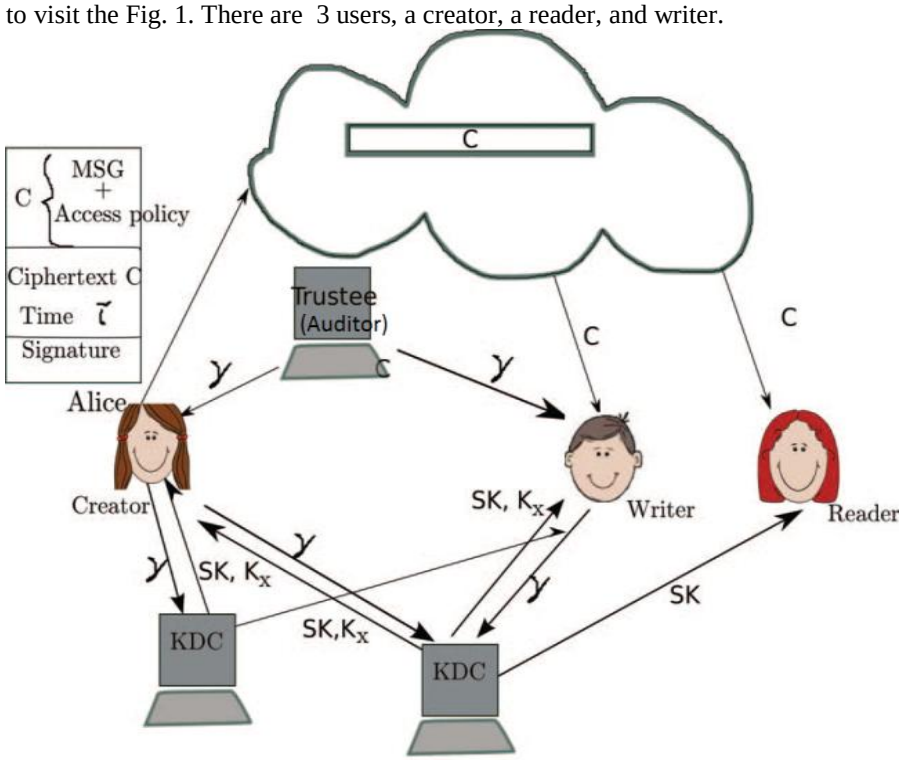

Fig 1. Our secure and authenticated cloud storage model

Creator Alice receives a token γ from the trustee, who is assumed has honest. A trustee(Auditor) may be someone just like the central United Nations agency manages welfare numbers etc. On presenting her id (like health/social insurance number), the trustee provides her a token γ. There are multiple KDCs , which might be scattered. For example, these may be servers in several components of the globe. A creator on presenting the token to 1 or more KDCs receives keys for encryption/decryption and sign language. In the Fig. 1, SKs are secret keys given for decoding, Kx are keys for sign language. The message MSG is encrypted underneath the access policy X. The access policy decides United Nations agency will access the information hold on within the cloud. The creator decides on a claim policy Y, to prove her believability and signs the message underneath this claim. The ciphertext C with signature is c, and is distributed to the cloud. The cloud has verify the signature and stores the ciphertext C. Then the trustee United Nations agency is audit the data which is stored in the cloud. Auditing the data is very useful while we fetches our data from the cloud. Once a reader wants to read, the cloud sends C. If the user's attributes matching with access policy, it will decipher and acquire back original message. Write income within the same means as file creation. By designating the verification method to the cloud, it has relieve the individual users from time intense verifications. Once a reader wants to read some information that hole on within the cloud, it tries to decipher it using the secret keys it receives from the KDCs. If it's enough attributes matching with the access policy, then it decrypts the data stored in the cloud.

V. SECURITY DEFINITION

We 1st gift the necessities of correctness of our proposed scheme by the following conditions:

(1) Dec(Enc(M,S, P), P, SK) = M, if the attribute set A of SK satisfies S.
(2) Let C′ = ReEnc(Enc(M,S, P), r, β), and SK′ = (ver +1, A,D, ¯D ′ = ReKey(¯D, r, θ)), where ver is the version number of P and r. Dec(C′, P′, SK′) = M, if A′ = A\(β \ θ) satisfies S.
(3) Let T′′ = ReEnc(T′, r′, β′), and SK′′ = (ver + 2, A′,D,ReKey(¯D ′, r′, θ′)). If Dec(C′,P′,SK′) = M and A′′=A′\(β′\ θ′) satisfies S, Dec(C′′,P′′,SK′′) =M.
(4) Inductively we get the statement for (C(n), P(n),SK(n)) of any n.

CPA security of our projected theme beneath the selective structure model [9] can be defined by the following game between an adversary X and a challenger Y.

CPA Security Game Let $^{\lambda}$ be a security parameter. We say that our scheme is secure against chosen plaintext attacks under selective-structure model if no PPT adversary X can win the following game with non-negligible advantage.

Init The adversary X chooses the challenge access structure S*, a version number ver*, and ver* − 1 attribute sets { γ (1), γ (2), ・ ・ ・ , γ (ver*−1)}, and submits them to the challenger Y.

Setup The challenger Y fist runs Setup(1^{λ}) to obtain MK and P for version 1. He then runs ReKeyGen(γi,MK) from i = 1 to ver*−1. Finally, B gives (P, {r(i)}2≤i≤ver*) to X, where r(i) denotes the proxy re-key set for version i3. Note that, X is able to derive P for all the versions with r(i)'s.

Phase 1 The adversary X is allowed to issue polynomial times (in $^{\lambda}$) of queries on generation of secret keys of any version within the range of [1, ver*]. The only restrict is that the attribute set that X submits for each secret key query does not satisfy S*.

Challenge The adversary submits two equal length messages M0 and M1. The challenger flips a random coin b, and encrypts Mb by executing C* ← Enc(M,S*, P), where P is the public parameter for version ver*. The challenge ciphertext C* is passed to the adversary.

Phase 2 Phase 1 is repeated.

Guess The adversary X outputs his guess b0 of b.

The adversary X is advantage in winning this CPA security game is defined as ADVCPA = Pr[b0 = b] – 1/2 .

issue queries on re-encryption of ciphertexts and on update of secret keys. In our security game, the adversary has been given all the proxy re-key's. This means that he is able to answer the two queries by himself. For this sake, we have a tendency to don't embrace the two corresponding oracles in Phase 1. In fact, the adversary X has at least the same capability as proxy servers who passively collect secret keys of unauthorized users. Since we assume proxy servers are honest, we do not consider active attacks from proxy servers by colluding with revoked authorized users.

Definition 1. (CPA SECURITY) We say that our scheme is CPA secure if ADVCPA is negligible (in $^{\lambda}$) for any polynomial time adversary.

VI. COMPARISON WITH OTHER ACCESS CONTROL SCHEMES IN CLOUD

We compare other access control schemes with our scheme and show how our scheme supports many features that the other schemes did not support. 1-W-M-R means that one user can be able to write and many users can be able to read. M-W-M-R means that many users can able to write and read. Most schemes did not support many writes but our scheme support. Compared with other schemes, our scheme is decentralized and robust and other schemes are centralized. Privacy preserving authentication is supported by our schemes. Our scheme support user revocation, most of others are not supported. We have presented the decentralized access control technique which prevents replay attacks and also provides user revocation. In most of the other schemes, cloud knows the access policy for each and every record that are stored in cloud, in our schemes we would hide the attributes and access policy of user.

VII. CONCLUSION

We have presented the identity and authentication of data stored in cloud, which prevents replay attacks. The cloud does not know the user identity who stores information in cloud. Key distribution is done only in decentralized way not centralized.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] J. Hur and D. Kun Noh, "**Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems**," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214-1221, July 2011.

[5] https://www.cs.purdue.edu/homes/jrr/pubs/AdvComp.pdf "**Secure outsourcing of scientific computation**".

[6] 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.

[7] Kan Yang, Xiaohua Jia "**Third-party Storage Auditing Service",** 2014.

[8] 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.

[9] S. Kamara and K. Lauter, "**Cryptographic Cloud Storage**," Proc. 14th Int'l Conf. Financial Cryptography and Data Security, pp. 136- 149, 2010.

[10] 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.

