International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 25 Issue 6 – MAY 2018.

SECURITY AND MAINTENANCE IN INTERCLOUD ENVIRONMENT FOR MULTICLOUD DATA INTENSIVE APPLICATION

J.Lavanya [#1] and B.Rajmohan [*2]

[#] *M.E - II Year,Dept of CSE, Adhiparasakthi Engineering College, Melmaruvathur.India*

[*] *Assistant Professor/CSE, Adhiparasakthi Engineering College, Melmaruvathur.India*

***Abstract*— Cloud computing is a platform for storing the data and hence there should be some care taken towards the disaster that happens in the cloud. A simple recovery service from disaster is proposed using multiple cloud service providers. These multiple cloud service providers are set aside at different locations so that if any of the cloud fails the data could be retrieved from the remaining cloud interfaces. The proposed technique has greater availability as the data can be recovered from disaster. This paper presents results of the ongoing development of the Inter Cloud Security Framework (ICSF) that is a part of the Inter cloud Architecture Framework (ICAF) and provides an architectural basis for building security infrastructure services for multi-cloud applications. The paper refers to general use case of the data intensive applications that indicate need for multi-cloud applications platforms that will require corresponding multi-cloud security services. When data owner uploads a file, the file would be encrypted using 3DES encryption algorithm and replicated and stored in multiple clouds.**

***Index Terms*—Cloud Computing, Security Framework, 3DES, Intercloud**

I. INTRODUCTION

Cloud computing is a revolutionary computing technique by which computing resources are provided dynamically via Internet and the data storage and computation are outsourced to someone or some party in a 'cloud'. It greatly attracts attention and interest from both academic and industry due to the profitability, but it also has at least three challenges that must be handled before coming to our real life to the best of our knowledge. First of all, data confidentiality should be guaranteed. The data privacy is not only about the data contents. Since the most attractive part of the cloud computing is the computation outsourcing, it is far beyond enough to just conduct an access control. More likely, users want to control the privileges of data manipulation over other users or cloud servers. This is because when sensitive information or computation is outsourced to the cloud servers or another user, which is out of users' control in most cases, privacy risks would rise dramatically because the servers might illegally inspect user's data and access sensitive information, or other users might be able to infer sensitive information from the outsourced computation. Therefore, not only the access but also the operation should be controlled. Secondly, personal information (defined by each user's attributes set) is at risk because one's identity is authenticated based on his information for the purpose of access control (or privilege control in this paper). As people are becoming more concerned about their identity privacy these days, the identity privacy also needs to be protected before the cloud enters our life. Preferably, any authority or server alone should not know any client's personal information. Last but not least, the cloud computing system should be resilient in the case of security breach in which some part of the system is compromised by attack.

II. SYSTEM ARCHITECTURE

In our system, there are four types of entities: N Attribute Authorities (denoted as A), Cloud Server, Data Owners and Data Consumers. A user can be a Data Owner and a Data Consumer simultaneously. Authorities are assumed to have powerful computation abilities, and they are supervised by government offices because some attributes partially contain users' personally identifiable information. The whole attribute set is divided into N disjoint sets and controlled by each authority, therefore each authority is aware of only part of attributes.

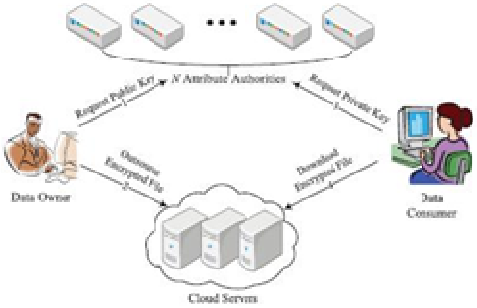

Figure 1.Architecture diagram

A Data Owner is the entity who wishes to outsource encrypted data file to the Cloud Servers. The Cloud Server, who is assumed to have adequate storage capacity, does nothing but store them. Newly joined Data Consumers request private keys from all of the authorities, and they do not know which attributes are controlled by which authorities. When the Data Consumers request their private keys from the authorities, authorities jointly create corresponding private key and send it to them. All Data Consumers are able to download any of the encrypted data files, but only those whose private keys satisfy the privilege tree Tp can execute the operation associated with privilege p. The server is delegated to execute an operation p if and only if the user's credentials are verified through the privilege tree Tp.

III. SYSTEM MODULES

A. *Design Goals*

Our goal is to achieve a multi-authority CP-ABE which: achieves the security defined above; guarantees the confidentiality of Data Consumers' identity information and tolerates compromise attacks on the authorities or the collusion attacks by the authorities. For the visual comfort, we frequently use the following notations hereafter. Ak denotes the k-th attribute authority; Au denotes the attributes set of user u; Auk denotes the subset of Au controlled by Ak; and ATp denotes the attributes set included in tree Tp.

B. *AnonyControl construction Setup*

At the system initialization phase, any one of the authorities chooses a bilinear group G0 of prime order p with generator g and publishes it. Then, all authorities independently and randomly picks vk∈Zp and send Yk = e(g, g)vk to all their authorities who individually compute Y := _k∈ A Yk = e(g, g)_k∈ A vk . Then every authority Ak randomly picks N − 1 integers skj∈Zp(j ∈ {1, . . . , N}{k}) and computes gskj . Each gskj is shared with each other authority Aj. An authority Ak, after receiving N −1 pieces of gsjk generated by Aj.

C. *Achieving Full Anonymity*

We have assumed semi-honest authorities in AnonyControl and we assumed that they will not collude with each other. This is a necessary assumption in AnonyControl because each authority is in charge of a subset of the whole attributes set, and for the attributes that it is in charge of; it knows the exact information of the key requester. If the information from all authorities is gathered altogether, the complete attribute set of the key requester is recovered and thus his identity is disclosed to the authorities. In this sense, AnonyControl is semi anonymous since partial identity information (represented as some attributes) is disclosed to each authority, but we can achieve a full-anonymity and also allow the collusion of the authorities.

D. *Fully Anonymous Multi-Authority CP-ABE*

In this section, we present how to achieve the full anonymity in AnonyControl to designs the fully anonymous privilege control scheme AnonyControl-F. The Key Generate algorithm is the only part which leaks identity information to each attribute authority. Upon receiving the attribute key request with the attribute value, the attribute authority will generate H(att (i))ri and sends it to the requester where att (i) is the attribute value and ri is a random number for that attribute. The attribute value is disclosed to the authority in this step.

By introducing the 1-out-of-k OT in our Key Generate algorithm, the key requester achieves the correct attribute key that he wants, but the attribute authority does not have any useful information about what attribute is achieved by the requester. Then, the key requester achieves the full anonymity in our scheme and no matter how many attribute authorities collude his identity information is kept secret.

IV. MODULE DESCRIPTION

A. *Key Generation and Exchange*

The client made request to the key manager for the public key, which will be generated according to the policy associated with the file. Different policies for files, public key also differs. But for same public key for same policy will be generated. Then the client generates a private key by combining the username, password and security credentials.

Then the file is encrypted with the public key and private key and forwarded to the cloud.

The client can download the file after completion of the authentication process. As the public key maintained by the key manager, the client request the key manager for public key. The authenticated client can get the public key. Then the client can decrypt the file with the public key and the private key. The user's credentials were stored in the client itself. During download the file the cloud will authenticate the user whether the user is valid to download the file. But the cloud doesn't have any attributes or the details of the user.

B. Storing and Splitting

The key generation gets over then the file must be uploaded in multicloud to store in drop box by executing the splitting algorithm in the methods of multi-cloud and now the file stored in multi-cloud with splitting methods.

C. Key Hiding and File Hiding

Once the file will be stored in multi-cloud then the file gets hidden with key. The data owner will generate the file and saved in drop box were as multi-cloud.

D. File Merging and Response Retrieval

File saved in multi-cloud which were in drop box after client send her request to owner to download the file from multi-cloud then owner will check out which client send the request from registration want to download the file then owner send its response with key generation after client received its response from owner now client will download whole file with the help of merging algorithm response retrieval to download and view the file the process based of encryption and decryption algorithm .

V. CONCLUSION

This paper proposes a semi-anonymous attribute-based privilege control scheme AnonyControl and a fully-anonymous attribute-based privilege control scheme AnonyControl-F to address the user privacy problem in a cloud storage server. Using multiple authorities in the cloud computing system, our proposed schemes achieve not only fine-grained privilege control but also identity anonymity while conducting privilege control based on users' identity information. More importantly, our system can tolerate up to N − 2 authority compromise, which is highly preferable especially in Internet-based cloud computing environment. We also conducted detailed security and performance analysis which shows that AnonyControl both secure and efficient for cloud storage system. The AnonyControl-F directly inherits the security of the AnonyControl and thus is equivalently secure as it, but extra communication overhead is incurred during the 1-out-of-n oblivious transfer. One of the promising future works is to introduce the efficient user revocation mechanism on top of our anonymous ABE. Supporting user revocation is an important issue in the real application, and this is a great challenge in the application of ABE schemes. Making our schemes compatible with existing ABE schemes that support efficient user revocation is one of our future works.

REFERENCES

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in CRYPTO. Springer, 1985, pp. 47–53.
[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in EUROCRYPT. Springer, 2005, pp. 457–473.
[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in CCS. ACM, 2006,89–98.
[4] J. Bethencourt, A. Sahai, and B. Waters,"Ciphertext-policy attributebased encryption," in S&P. IEEE, 2007, pp. 321–334.
[5] M. Chase, "Multi-authority attribute based encryption," in TCC. Springer, 2007, pp. 515–534.
[6] M. Chase and S. S. Chow, "Improving privacy and security in multiauthority attribute-based encryption," in CCS. ACM, 2009, pp. 121– 130.
[7] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," Information Sciences, vol. 180, no. 13, pp. 2618–2632, 2010.
[8] V. Bozovi ˇ c, D. Socek, R. Steinwandt, and V.I. Vill ´ anyi, "Multi- ´ authority attribute-based encryptionwithhonest-but-curiouscentral
[9] authority," IJCM, vol. 89, no. 3, pp. 268–283, 2012.
[10] F. Li, Y. Rahulamathavan, M. Rajarajan, and R.-W. Phan, "Low complexity multi-authority attribute based encryption scheme for mobile cloud computing," in SOSE. IEEE, 2013, pp. 573–577.
[11] K. Yang, X. Jia, K. Ren, and B. Zhang, "Dac-macs: Effective data access control for multi-authority cloud storage systems," in INFOCOM.IEEE, 2013, pp. 2895–2903.