International Journal of Research in Advent Technology, Vol.6, No.11, November 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

Efficient Access Control Scheme with Multiple Attribute Authorities for Public Cloud Storage

Mr. Ramesh[1], Prof. Girish Kumar D[2],
Student, M.tech(Computer Networking)[1], Assistant Professor, Computer Science Department[2], ,BITM[1,2],Ballari[1,2]
Email: rameshbitm16@gmail.com[1], get.girivar@gmail.com[2]

Abstract- Information get to control is a testing issue out in the open distributed storage frameworks. Cipher content Policy Attribute-Based Encryption (CP-ABE) has been embraced as a promising method to give adaptable, fine-grained and secure information get to control for distributed storage with genuine however inquisitive cloud servers. In any case, in the current CP-ABE plans, the single characteristic expert must execute the tedious client authenticity check and mystery key dissemination, and thus it results in a solitary point execution bottleneck when a CP-ABE conspire is received in an extensive scale distributed storage framework. Clients might be stuck in the trusting that an extensive stretch will get their mystery keys, in this manner bringing about low-effectiveness of the framework. In this paper, we propose a novel heterogeneous structure to expel the issue of single-point execution bottleneck and give a more productive access control plot with an inspecting component. Our system utilizes different ascribe experts to share the heap of client authenticity confirmation. Dissimilar to other multiauthority get to control plots, every one of the experts in our plan deals with the entire trait set independently. To improve security, we additionally propose an inspecting instrument to distinguish which AA (Attribute Authority) has erroneously or noxiously played out the authenticity confirmation technique. Examination demonstrates that our framework ensures the security necessities as well as makes extraordinary execution enhancement for key age.

Index Terms-Cloud storage, Access control, Auditing.

1. INTRODUCTION

Cloud storage is one of the important service paradigm in ac cloud computing [1]-[4]. There are several benefits by using a cloud storage some of few are it provides a accessibility, reliability, rapid deployment and the stronger protection. The above mentioned benefits brings new challenges on data access control which is a critical issue ensure the data security. The cloud storage which is operated by a cloud services provider are outside the trusted domain of the data owners, the traditional access control methods will not suitable in cloud storage environment. In cloud storage environment data access control has become one of the big challenging issues. CP-ABE has a advantage that it provides a data owners direct control power based on the access policies, because to provide flexible, fine grained and secure access control for cloud storage systems. User can decrypt chipper text to plain text only if the access structure satisfy, it is possible only by that the attribute must associated with the user's secret key. CP-ABE has developed for cloud in two types of group's first single authority [5]-[9], and multi-authority [10]. But the existing CP-ABE has a lots of disadvantage they are neither stronger nor efficient in key generation. In CP-ABE a single authority must take care of all the responsibility, if the system crash/offline this authority will make the key generation unavailable during this period. In our proposed project similar problem exist, but each multiple attribute manage disjoint attribute set.

1.1 *Problem Statement*

In cloud storage multiple authority performing same operation and it will be difficult to identify misbehaving authority, that who have made the mistake in while performing a secret key generation and distribution. For example, a performing authority may distribute secret keys beyond user's legitimate attribute set (without informing to the central authority CA). Such a security problem will makes this straight forward idea hard to meet the security requirement of access control for public cloud storage.

1.2 *Objective Of The Project*

Multiple attribute authority will allow the user to request the secret key without waiting in the queue has mentioned in the existing system each authority will individual handle the process. A single CA(central authority take care of all the responsibility and it will monitor the multiple authority, if any authority is misbehaved that authority will be deleted and the secret key which is generated for user that will be canceled

1.3 *Proposed Statement*

The problem in the CP-ABE existing can be reduce by proposing the stronger and efficient key generation with a CA and multiple AA's. The work load and verification of user's is shared by each authority, and further verification is done by CA. By proposing single CA with multiple attribute authorities can be overcome. In our proposed project user can decrypt chipper text to plain only when the associated attribute with the secret must satisfy the access structure. In our proposed project includes an auditing mechanism by using this a CA can identify misbehaving AA.

2. LITRATURE SURVEY

2.1 *Multiple Data Owners With Multiple Keyword Over Encrypted Cloud Data.*

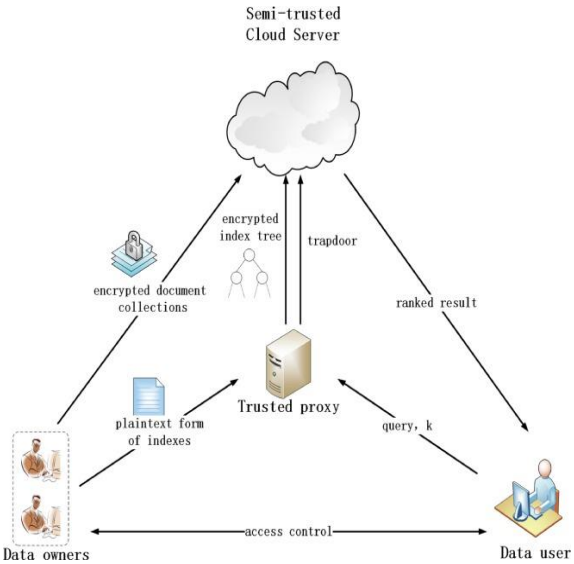

Figure 2.1 Multiple Data Owner

Over encrypted cloud data number of efficient search schemes where proposed by many researchers. The general process of search scheme is divided into five steps: extracting document features, constructing a searchable index, generating search trapdoor, searching the index based on the trapdoor and returning the search results. Search scheme provide different capabilities, including single keyword, multi-keyword, fuzzy keyword search and so on.

2.2 *In a Cloud Computing Privacy Preserving Keyword Search Was Done Over Encrypted Data.*

The typical participants of a secure search system in the cloud involve the cloud server, the data owner, and the data user. When a data user wants to query the outsourced dataset hosted on the cloud server, he/she first either generates a Search control with the keyword of interest (applied to most PKC-based search schemes), or requests such search control by sending a set of intended keywords to the data owner (in the case of SKC-based search schemes). In the latter case, upon receiving the search control generation request, the data owner constructs the search control, and returns it to the user. Then the data user submits the search control to the cloud server. The cloud server will execute the search program with the search control as the input, the search results will be sent back to the user.

3. SYSTEM DESIGN

In System Design has divided into three types like GUI Designing, UML Designing with avails in development of project in facile way with different actor and its utilize case by utilize case diagram, flow of the project utilizing sequence, Class diagram gives information about different class in the project with methods that have to be utilized in the project if comes to our project our UML Will utilizable in this way The third and post import for the project in system design is Data base design where we endeavor to design data base predicated on the number of modules in our project

3.1 *High Level Design*

A high level design provides an overview of a solution, flat form, system, product, service or process. A high level components, interface and networks will usually include in a high level design document that need to be developed. The document may also depict or otherwise referred to work flows or data flows between the component systems.
The developed system is based on providing the stronger security with efficiencies and timing constraints. As well the single CA will take the serious action on the misbehaving authority. The overall developed system architecture is shown in fig 4.3, where each module is explained in the below sections.

3.2 *System Architecture*

The architecture defines connections that take place during a user interaction with the server. Owner upload a file to cloud server. Next the user need to download the file from the server so user need secret key user will request made to attribute authority. Authority will verify and provide a intermediate key and send to the central authority for further verification. After verifying central authority will provide a secret key to user. Next user will download the file from cloud server using secret key.

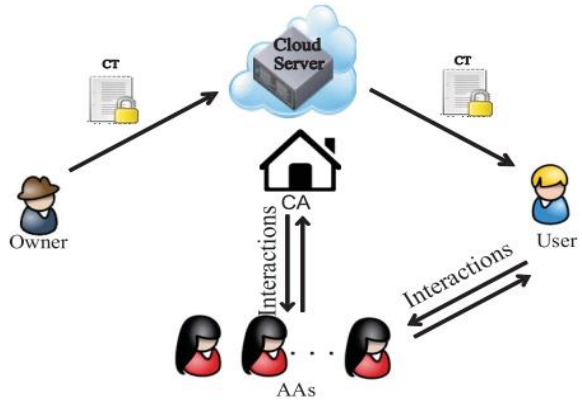

Figure 3.2 System Architecture

3.4 *Detailed Design*

Detailed design gives more details how each component is to operate and they will function together is heavily addressed. In this phase the parts are listed such as prototype, schematics, and they are presented as a theoretical solution to this project. Central Detailed design is especially vital given that it exits at the intersection of so many other development processes.
Outputs from detailed design process are used not only as finished one, but also may take the form of modular components useful for developing other variants.
The modules of the developed project are:

- Central authority
- Attribute authorities
- Data owner
- User
- Cloud server

- ***Central Authority (CA):***

The central authority (CA) is that the administrator of the whole system. CA contains a main responsibility that require to get a public key for every attribute, every user can receive a singular id every attribute authority will receive a singular id. At the time of system initializing every user can get a singular id. A user can create asking to CA for a key and also the CA are accountable to get a secret key. Suppose user can create asking to the CA. A CA can generate a secret key for user supported intermediate key related to user's legitimate attributes that was verified by AA. CA got to pay attention of entire system and wish to stay eye on every AA and have trace UN agency sis misbehaving and corroborative incorrectly a user and issue a attribute sets.

- ***Attribute Authorities (AAS):***

The (AAs) will take responsible of generate intermediate keys for user and verifying user legitimacy. A multiple (AA) has capability to manage disjoint attribute set, each authority will share the responsibility of verifying user's legitimacy, each authority perform this verification for each individual user of any. When AA is selected to verify the user legitimacy it will verify manual and generate a intermediate key associated with the attributes that as legitimacy verified. Only when the AA has verified the user next CA will process the further operation.

- ***Data Owner:***

The data owner (Owner) defines the access policy regarding who will get a access to every file, and encrypts the file underneath the outlined policy. List of all every owner encrypts his/her information with a cruciform cryptography algorithm rule. The owner formulates access policy over an attribute set and encrypts the symmetric key under policy according to public keys obtained from CA. At that time owner sends the full encrypted information and also the encrypted cruciform key to the cloud server to be kept within the cloud.

- ***User:***

The data consumer (User) is assigned a global user identity Uid by CA. The client has an arrangement of characteristics and is outfitted with a mystery key related with his/her property set. Client will effortlessly get the encoded information from te cloud server. Be that as it may, the client can decode the encoded information if and just if his/her quality set fulfills the entrance approach inserted in the scrambled information.

- ***Cloud Server:***

The cloud server gives an open stage to proprietors to store and offer their encoded information. The cloud server doesn't lead information get to control for proprietors. The scrambled information put away in the cloud server can be downloaded uninhibitedly by any client

4. DATAFLOW DIAGRAM

4.1 *User And Owner Interaction With Cloud*

Client and proprietor connection is appeared in the beneath dataflow outline. We can see that proprietor will transfer the document to the cloud server by scrambling the record utilizing two layer activity. Client will download the document by utilizing the mystery key.

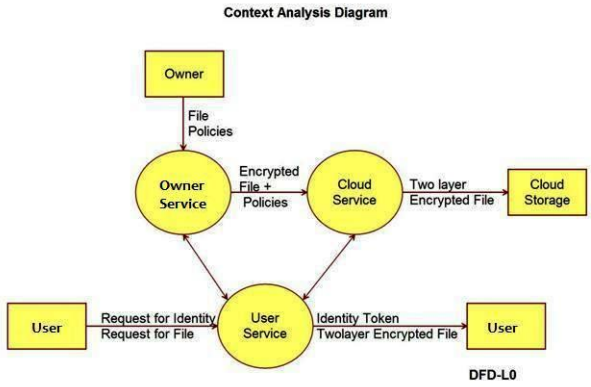

Figure 4.1 User Owner Interaction With Cloud

4.2 *Owner Uploading File In Cloud Storage*

Below diagram shows the interaction of owner uploading a file in cloud storage. Owner will upload a file by specifying the attribute details and encrypt the file using symmetric key and public key, it use two layer operation to encrypt a file. After encrypting a file owner will upload a file in cloud storage.

.

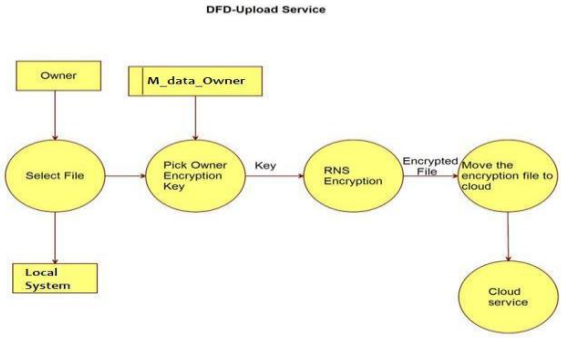

Figure:-4.2 Owner Uploading File

5. IMPLEMENTATION

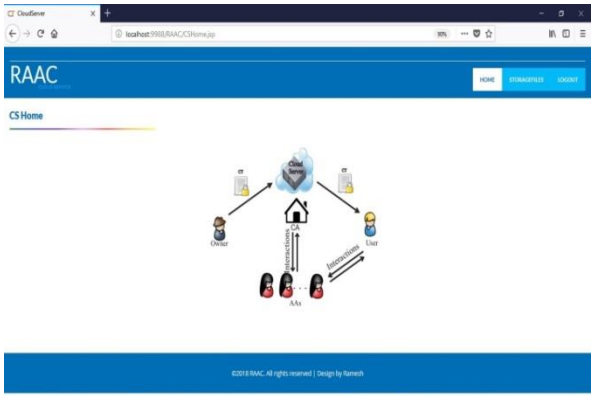

Fig 5. Cloud Server Home Page

Algorithm used in Project

1) Cipher text-Policy Attribute-Based Encryption (CP-ABE).
2) Advanced Encrypted Standard (AES).
3) Digital Signature Algorithm (DSA).

1) Cipher text-Policy Attribute-Based Encryption (CP-ABE)

Although the definitions and constructions of different CPABE schemes are not always consistent, the uses of the access structure in Encrypt and Decrypt algorithms are nearly the same. Here we adopt the definition and construction from [18,22].

A CP-ABE scheme consists of four algorithms: Setup, Encrypt, Key Generation (KeyGen), and Decrypt. Setup(λ, U) $\rightarrow$ (PK,MSK). The setup algorithm takes the security parameter λ and the attribute universe description U as the input. It outputs the public parameters PK and a master secret key MSK.

Encrypt(PK,M,A) $\rightarrow$ CT. The encryption algorithm takes the public parameters PK, a message M, and an access structure A as input. The algorithm will encrypt M and produce a ciphertext CT such that only a user whose attributes satisfies the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains A.

KeyGen(MSK, S) $\rightarrow$ SK. The key generation algorithm takes the master secret key MSK and a set of attributes S as input. It outputs a secret key SK.

Decrypt(PK,CT, SK)$\rightarrow M$. The decryption algorithm takes the public parameters PK, a ciphertext CT which contains an access policy A, and a secret key SK as input, where SK is a secret key for a set S of attributes. If the set S of attributes satisfies the access structure A, the algorithm will decrypt the ciphertext and return a message M.

2) AES Algorithm

1) Key Expansions
- For each round AES requires a separate 128-bit round key block plus one more.
- Initial Round
- Add Round Key with a block of the round key, each byte of the state is combined using bitwise xor.

2) Rounds
- Sub Bytes in this step each byte is replaced with another byte.
- Shift Rows for a certain number of steps, the last three rows of the state are shifted cyclically.
- Mix Columns a mixing operation which operates on the columns of the state, combining the four bytes in each column.

3) Add Round Key
- Final Round (no Mix Columns)
- Sub Bytes

 Shift Rows Add Round Key

6. CONCLUSION

In this paper, to overcome the problem of existing system we have introduced a new frame work called EACS(Efficient Access Control Scheme). In existing system there was single point performance bottleneck to overcome the problem we introduce EACS. By introducing EACS it provides a greater security and also provide a auditable access control with multiple authorities, it provides greater controls for cloud storage. To trace the AAs a new method was introduce called auditing method. This is used to trace the misbehavior AA. The proposed auditing & tracing scheme, no AA could deny its misbehaved key distribution.

ACKNOWLEDGMENT

I would sincerely like to thank our Professor Girish Kumar D, Department of Computer Engineering, BITM, Ballari for his guidance, encouragement and the interest shown in this project by timely suggestions in this work. His expert suggestions and scholarly feedback had greatly enhanced the effectiveness of this work.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-145, 2011.

[2] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 9, pp. 2546–2559, Sep. 2016.

[3] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2016, pp. 1–9.

[4] K. Xue and P. Hong, "A dynamic secure group sharing framework in public cloud computing," IEEE Trans. Cloud Comput., vol. 2, no. 4, pp. 459–470, Oct. 2014.

[5] Y. Wu, Z. Wei, and R. H. Deng, "Attribute-based access to scalable media in cloud-assisted content sharing networks," IEEE Trans. Multimedia, vol. 15, no. 4, pp. 778–788, Jun. 2013.

[6] J. Hur, "Improving security and efficiency in attribute-based data sharing," IEEE Trans. Knowl. Data Eng., vol. 25, no. 10, pp. 2271–2282, Oct. 2013.

[7] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 7, pp. 1214–1221, Jul. 2011.

[8] J. Hong, K. Xue, W. Li, and Y. Xue, "TAFC: Time and attribute factors combined access control on time-sensitive data in public cloud," in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2015, pp. 1–6.

[9] Y. Xue, J. Hong, W. Li, K. Xue, and P. Hong, "LABAC: A locationaware attribute-based access control scheme for cloud storage," in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2016, pp. 1–6.

[10] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in Advances in Cryptology—EUROCRYPT. Berlin, Germany: Springer, 2011, pp. 568–588.