IJARIIT
INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 3)
Available online at: www.ijariit.com

Efficient and secure data access control with multiple attribute authority in public cloud storage

Chandan G.
gchandan19@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

Achutha H. Bharadwaj
achu.97bharadwaj@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

Bhuvanendra R. Hirewadeyar
r.bhuvanendra@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

Deepak S.
deepak.suresh200@gmail.com
Don Bosco Institute of Technology, Bangalore, Karnataka

B. S. Umashankar
umashankar.cs@dbit.co.in
Don Bosco Institute of Technology, Bangalore, Karnataka

ABSTRACT

One of the main challenges in public cloud storage is accessing the data. There are various techniques to provide flexible, fine-grained and secure control access and one among them is Cipher-Text Policy Attribute-Based Encryption (CP-ABE). In CP-ABE scheme, single attribute authority executes legitimacy verification and secret key distribution and this results in a single point bottleneck approach. When this scheme is applied on a large scale cloud then the user must wait for a longer period to obtain the secret key and performance of the system becomes low. To overcome single point bottleneck problem, a heterogeneous framework is used which employs multiple attribute authorities. This framework shares the load of user legitimacy verification. In the new scheme, central authority acts as the administrator of the system and shares the secret key whereas the multiple attribute authority manages the whole attribute set independently. Proposed scheme enhances more security and improves the efficiency of the system.

Keywords— *CP-ABE, Cloud Storage, Access control*

1. INTRODUCTION

Cloud is mainly used to transfer the data very securely from owner to user. The main advantage of using the cloud in data transfer is, it's scalable, reliable, and accessible and so on. Cloud is handled by the cloud service provider and it is trusted by the data owner. Data access control in the cloud is the main issue.

To solve this issue one of the famous cryptographic technique called Ciphertext-Policy Attribute Encryption (CP-ABE) is used. In CP-ABE scheme, the data owner would encrypt the data and upload it to the cloud. User can access the data from cloud if and only if the secret key given to the user matches with his/her attribute. If it matches, then the user can decrypt the data from the cloud i.e., the user will be able to convert the cipher text into plain text.

CP-ABE has two schemes namely, single authority scheme and multi-authority scheme. In a single authority scheme, as the only single authority will be available there would too much of crash. In this scheme, the user would be provided with a secret key and with the help of this secret key, the user would be able to access the data from the cloud. In a single authority scheme, only one attribute authority would be available to verify the user. When a number of user's increases single authority scheme would fail to authenticate the users and this leads to a single bottom neck approach. This approach leads in low efficiency of the system and verification of the users and owner also fails. Single bottom neck approach also affects the key generation and enforces the user to be active for verification based on some attributes. In a multi-authority scheme, the same problem exists as multi authorities would make use of disjoint attribute set. In this scheme, the secret key would be generated and verification of the user is similar to the single authority scheme. Hence, the problem i.e., single point bottom neck approach still exists in this scheme also.

To overcome the problem of single point bottom neck, instead of using the multi-authority that makes use of disjoint attribute set, multiple attributes can be used. When multiple attributes are used, the load would be shared and verification would be easy when compared to the old scheme. By making use of multiple attributes the malicious user can be restricted from accessing the data from the cloud. By making use of multiple attributes we are increasing the overhead so that the system would be more efficient and making the system robust.

A similar problem is found in Public Key Infrastructure (PKI), to reduce the load on Certificate Authority (CA), multiple Registration Authorities (RA) are used. Multiple RA's would be verifying and once the verification is done the request to certificate would be sent to the CA. Later on, CA would be able to generate the certificate to the user. Lot of loads on CA would be reduced by making use of multiple RA and to keep track of verification CA would store information about which RA verifies.

We propose a novel heterogenous called as Efficient and secure data access control with multiple attribute authority in public cloud storage. In this framework, multiple AAs are used for verification of the user and for the generation of keys. Central Authority (CA) is the administrator of the system and is responsible for allowing both owner and user to access the data that is present in the cloud. CA is also responsible for the generation of secret key and distribution in the system. One of the AA is used to verify the user and generates the intermediate key and sends it to the CA. Upon the intermediate key, the CA would generate the secret key to the user. By making use of multiple AAs loads can be shared among AA and single point bottom neck problem would be resolved. AA is not responsible for generating a key, is only responsible for verification of the user. Once the user is verified by the AA, during secret key generation verification of the user is not needed again. By making use of multiple AAs and CA legitimate of user verification is possible and also the malicious user is restricted.

2. PROBLEM STATEMENT

Ciphertext Policy Attribute Based Encryption is one the cryptographic technique which is used in accessing the data in public cloud storage. CP-ABE provides fine grained and flexible access control for data. CP-ABE was proposed by Benthencourt et al. CP-ABE is categorized into two schemes based on authorities.

Single authority scheme, in this scheme only one authority is responsible for generating key effectively and managing the attribute set. Another scheme known as Multi-message Cipher Text Policy Attribute Based Encryption (MCP ABE) is used to encrypt multiple messages with one cipher text and also is scalable. Multi-authority scheme is one of the schemes where it is similar to the single authority scheme and is also used to generate the key to the user and this scheme uses attribute disjoint set. In multi authority scheme, for each attribute, only one authority can issue key. However, in both the schemes single point bottom neck problem still exists.

To overcome single point bottom neck problem and also to achieve robust, efficient performance a new heterogeneous framework called as Efficient and Secure Data Access Control with Multiple Attribute Authority in Public Cloud Storage. Mainly two steps take place in this scheme. The first step is the verification of the user. The second step is to generate the key and distribute it to the verified user. In our proposed scheme, we make use of a single CA and multiple AAs. Multiple AAs will be responsible for verifying the user based on their attribute set independently. After verification, AAs would generate an intermediate key and distribute to the verified user. Later on, CA would generate the secret key to the verified user and distributes them based on the intermediate key.

Since multiple AAs are used for the verification of user's loads on CA is reduced. CA is the administrator of the system and keeps track of AA and makes sure that no AA is generating the secret key to the user without permission. CA also makes sure that malicious users are restricted to access the data in the cloud. CA keeps track which AA verified the user and distributed the intermediate key to the user. This scheme is suitable for large scale system where a large number of users request for accessing the data from public cloud storage.

In our scheme, there are mainly five phases i.e., System initialization, Encryption, Key Generation, Decrypting and Auditing and Tracing.

System initialization: CA is used to choose multiplicative cyclic groups G and G_T for generating a secret key for each attribute. Each AA sends the request to CA during system initialization and CA assigns a unique identity key to each request.

Encryption: This phase is done by the owner where the owner chooses a random number $\kappa \in G_T$ and encrypts the plain text into cipher text. The encrypted text will be done based on a symmetric encryption algorithm. The owner sets policy and based on this policy if the users credential meets then only the ciphertext will be converted to plain text.

Key Generation: Key generation is generated by CA and AA. In the first step, $Uj \rightarrow AAi$ when the user Uj makes a secret key request to AA then the requesting user would be verified to check whether they are the valid user. In the second step, $AAi \rightarrow CA$ user verification will be carried out by AAi. After verification, a time stamp would be there to generate the intermediate key.

Decryption: This phase is done by the user. Once the user is verified and get the intermediate key from AAi and also get the secret key from CA then using this key user will be able to decrypt the encrypted data from the data owner. Using symmetric key algorithm user decrypts the data and obtain the plain text.

Auditing and Tracing: This phase is handled by CA. Once the verification is completed by AA then CA would generate the secret key for the user. Later on, CA would keep track of AA and store some information about AA and restricts the malicious user from entering the system. CA would perform secret key ownership and AA tracing.

3. SYSTEM DESIGN

We describe details regarding system model, security assumptions and requirements of public cloud storage.

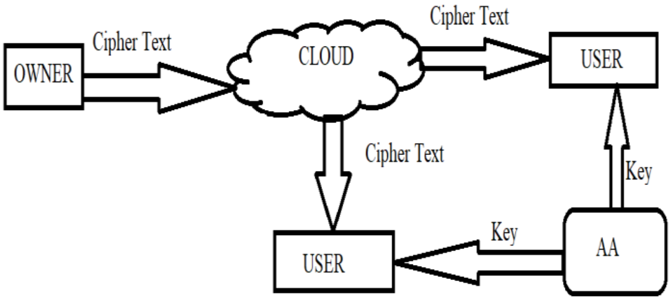


Fig. 1: System architecture

3.1 System model

3.1.1 Central Authority (CA): CA is the central administrator of the system and is responsible for the construction of the system and setting up the parameters and is also used to generate a public key. CA is responsible for generating a secret key for the user based on the key they receive and the users are verified by the AA. CA also traces which AA is not performing up to the mark.

3.1.2 Attribute Authorities (AAs): AA is used for verification of user independently. After verification of each user, an intermediate key is generated associated with attributes.

3.1.3 Data owner: Owner defines the access policy and encrypts the file. The owner defines the policy and later on sends the encrypted data to the cloud. CA would generate the public key and send it to the owner and later on owner sends the data.

3.1.4 Data consumer: User would receive the secret key from CA and can download the encrypted data from the cloud server. To download the data from the cloud server user needs to decrypt the encrypted data and must match the policy defined by the owner.

3.1.5 Cloud server: Cloud server is used to transfer the data from owner to use safety. Owner encrypts the data and stores it in the cloud and the user decrypts and downloads it freely.

3.2 Security assumptions and requirements

In the new scheme, the cloud server is used to transfer the data and is used to execute the task correctly. CA is the administrator of the system where CA has to stay online for generating the keys. AAs is used for verification of the user. But AA cannot be fully trusted so we require manual verification. The user cannot access the data until unless the attributes are matched. Owner is used to uploading the data and is secured by the policy. To ensure to access the data in cloud storage then it must follow two security requirements:

3.2.1 Data confidentiality: Data access must be prevented from unauthenticated users.

3.2.2 Collusion resistance: Unauthorized user will not be able to collide with the authorized user and the user who is not verified alone cannot access the data from the cloud.

4. IMPLEMENTATION

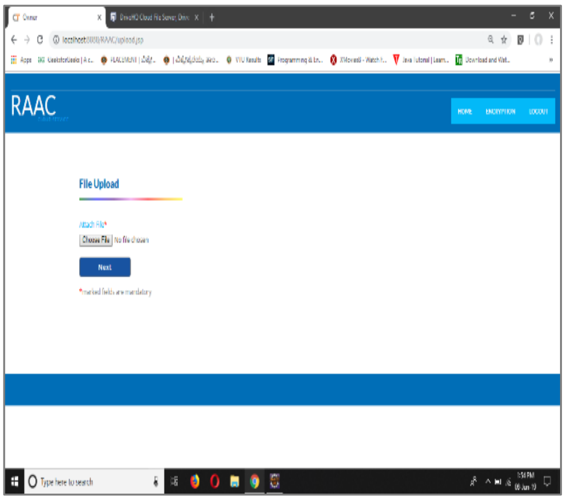
AES algorithm is used to encrypt the data and also helps in decrypting the data from the cloud.

4.1 Pseudo code for AES algorithm

```
Step 1:  function AES(byte in[16], byte out[16], key_array
             round_key[Nr+1])
Step 2:  byte state[16]
Step 3:  state=in
Step 4:  AddRoundKey(state, round_key[0])
Step 5:  For i=1 to Nr-1 do
             SubBytes(state)
             ShiftRows(state)
             MixColumns(state)
             AddRoundKey(state, round_key[i])
Step 6:  SubBytes(state)
Step 7:  ShiftRows(state)
Step 8:  AddRoundKey(state, round_key[Nr])
Step 9:  Out=state
Step 10: Return out
```

5. RESULT

The proposed system makes use of CP-ABE scheme to verify the user and allows them to access the data in the cloud. This scheme provides more security and improves the efficiency of the system.


Owner
DriveHQ Cloud File Server, Drive
localhost:8080/RAAC/upload.jsp
RAAC
HOME
ENCRYPTION
LOGOUT
Type here to search

File Upload

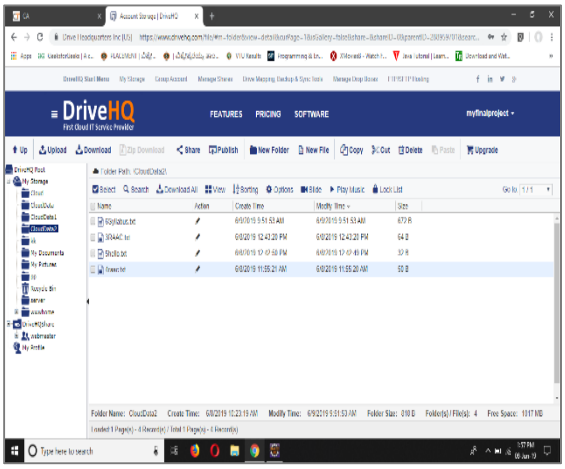
Attach File*

Choose File No file chosen

Next

*marked fields are mandatory

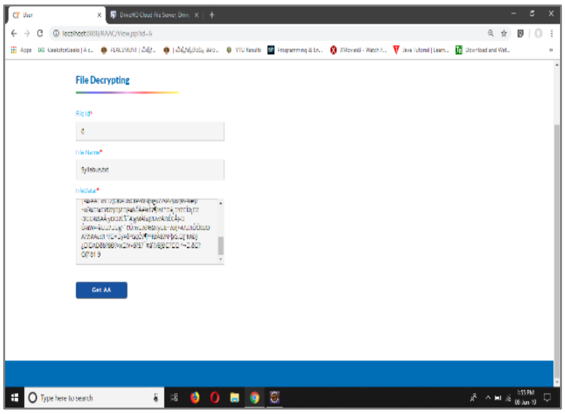
Fig. 2: Owner would upload the data to the cloud



Folder Path: \CloudData2\

Name	Action	Create Time	Modify Time	Size
6Syllabus.txt		6/9/2019 9:51:53 AM	6/9/2019 9:51:53 AM	672 B
3RAAC.txt		6/8/2019 12:43:20 PM	6/8/2019 12:43:20 PM	64 B
5hello.txt		6/8/2019 12:47:50 PM	6/8/2019 12:47:49 PM	32 B
4raac.txt		6/8/2019 11:55:21 AM	6/8/2019 11:55:20 AM	50 B

Fig. 3: Encrypted data would be stored in the cloud


User
DriveHQ Cloud File Server, Drive
localhost:8080/RAAC/View.jsp?id=6
Type here to search

File Decrypting

File Id*

6

File Name*

Syllabus.txt

FileData*

[illegible]

Get AA

Fig. 4: File decryption carried out by the user

6. CONCLUSION

In this paper, we proposed a new framework, called Efficient and Secure Data Access Control in Public Cloud Storage, which eliminates the single-point performance bottleneck from the existing CP-ABE schemes. We make use of one CA and multiple AAs so that our scheme provides a fine-grained and robust and efficient system. Multiple AA carries loads from many users and verify those users and distribute the intermediate key to them. And also, we trace the authority's potential malicious behaviour. Security analysis shows us that our proposed scheme improves security and provides an efficient and robust system. This framework is suitable for large scale data transfer and overcomes the CP-ABE scheme in data access in public cloud storage. Also helps in high-security enhancement with dual encryption mechanism.

7. REFERENCES

[1] 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.

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

[3] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in Proc. Int. Workshop Public Key Cryptogr., 2011, pp. 53–70.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput. Commun. Secure. (CCS), 2006, pp. 89–98.

[5] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi-authority attribute-based encryption without a central authority," Inf. Sci., vol. 180, no. 13, pp. 2618–2632, Jul. 2010.

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

[7] G. Shruthi, Purohit Shrinivasacharya, "A Combined Cipher Text Policy Attribute-Based Encryption and Time Release Encryption Method for Securing Medical Data in Cloud", 2019.