International Journal of Computer Applications (0975 – 8887)
Volume 116 – No. 22, April 2015

Modified Fine-grained Data Access Control Algorithms for File Storage Cloud

Deepak Mehar
[1]M. Tech Scholar,
Department of Computer Science and Engineering, (SATI) Vidisha, MP, India

Gagan Ishwakarma
[2]Lecture,
Department of Computer Science and Engineering, (SATI) Vidisha, MP, India

Y. K Jain
[3]HOD,
Department of Computer Science and Engineering, (SATI) Vidisha, MP, India

ABSTRACT
Cloud Computing is an emerging technology and it is used where computation, data manipulation and information sharing are needed. As number of Cloud users is increasing day by day, which leads to various security issues like unauthorized and unnecessary service access. To provide access control or preventing unauthorized access of data stored at Cloud Server Storage, the existing system applies cryptographic methods by sharing the decryption keys among intended authorized users or members. However, to perform these operations, existing system needs heavy computation power and resources. achieving scalability and access control simultaneously for file storage service in cloud environment is a big concern. The issues are yet not fully resolved. Our scheme proposed the solution for these issues by a mechanism that defines a package of Role based access control, Attribute based access control and security with scalable infrastructure simultaneously.

Keywords
Access Control, Role Based Access Control, Attribute Based Access Control, Scalable.

1. INTRODUCTION
Cloud Computing is a technology in which resources are provided as a service over the network in pay per use fashion. Any company, organization or individual who have sufficient resources to share with other companies, organizations or individuals can be a Cloud Service Provider (CSP) [1], [2]. CSP has to fulfill the promises which they made with their clients or cloud users. Promises such as every time service availability, scalability and assurance of data security. The data security or prevention from unauthorized access is very important issue in cloud environment. If these issues are not resolved properly, it can impede the processing of whole cloud system.

Data security issues are great concern for data owners when they store their confidential information on Cloud Server. Generally, Cloud Servers are outside from trusted domain of cloud user. In practical aspect, data security or data confidentiality may raise a juristic concern [3]; for example, in higher education management cloud, to use and share the protected information such as exam papers, storing this protected information on server is a requirement but its confidentiality is a big concern. Furthermore, sometimes cloud user themselves are content providers which share data or information through Cloud Server and need a fine-grained access control in terms of which data user has what access privilege to which type of data. In case of higher education cloud, the administration department would be the data owner who stores huge amount of data such as student's academic records and teacher's detail in the Cloud Server. Here students, teachers and other staff members would be the data consumer to access available data under security policies admitted by the administration.

In the research area of information security, access control has been evolving since thirty years and various effective techniques have been developed to achieve fine grained access control, which allows scalability and security to specify access privileges to individual user. Traditional access control architecture assumes that the Cloud Server defines data access policies for data owner as well as for data consumers, whereas the Cloud Server and the data owner both are in same trusted domain [5]. However, this assumption has no longer support to cloud architecture because in modern cloud architecture, Cloud Server and data owner not be in same trusted domain. The data resources are not physically under control of data owner. Hence, there are various possibilities of data theft, data misuse, data damage or destroying data by unauthorized user. For the purpose of help to data owner, to achieve fine grained data access control, we proposed a scalable, secure and distributed access control method for cloud computing.

2. RELATED WORK
Mansura, Md. Islam and A. B. M. Ali et al. [4] proposed a data access right model which enables a Cloud user to perform action according to their position in Access Right Tree. This access tree decides the role of user in cloud. The main issue in this model is, if any user authorized as a data owner than he can permit to perform same action on all available files. However, in various situations in cloud it is necessary to limit user's permissions for some files. In our proposed access right model, access permissions are having by file for particular user.

Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou et al. [3] proposed an access control model in which only data owners have unique access on their files, whereas in cloud environment a flexible, scalable and independent access control is necessary. In our scheme, we proposed a novel method of ongoing authorization to overcome from this challenge.

3. DATA OR FILE ACCESS RIGHT MODEL
In our proposed data access model, a set of access rights is organized in tabular structure, mentioned as an Access Right List (ARL). As shown in Table-1, where first column represents a unique code known as Access Level Code (ALC) and rest all columns represents access rights. Here access right columns can contain a Boolean value in form of true or false.

True value denotes permission to perform that particular action and false value denotes not permitted. These access rights or actions can be read, write, edit, move, share, download, temporary delete and permanently delete the file. In ARL, each unique ALC have distinct set of Boolean values in its correspondent columns. Eq. (1) describes the relationship ARL as following:

$$C \overset{grant}{\Longrightarrow} A$$
(1)

Here C is a unique code in access right list and A is a set of access rights which is corresponding to code C. Eq. (1) refers that if any file is entitled for C, it will set limits to access limits to requesting user according to access rights from set A. For example, if any file is accessible for any user and this file entitled an ALC which have access rights for read only permission then this file only can be readable by this particular user and may have different permissions for another user.

Table-1: Access Right List (ARL)

ALC	Read	Write	Edit	Move	Share	Download	Temporary Delete	Permanent Delete
1	T	T	T	T	T	T	T	T
2	T	T	T	T	T	T	T	F
3	T	T	T	T	T	T	F	F
4	T	T	T	T	T	F	F	F
5	T	T	T	F	T	T	F	F
6	T	T	F	F	T	T	F	F
7	T	F	T	F	T	T	F	F
8	T	F	F	F	F	F	F	F

4. PROPOSED SCHEME
To achieve fine grained access control in file storage cloud, we utilize and uniquely combine following access control methodologies: Attribute Based Access Control (ABAC) [6] [7], Access Right List (ARL), and Role Based Access Control (RBAC) [8]. More specifically, we provide an access rules set to each user corresponding to their selected plan for using cloud services and also provide a set of attribute to each file that decides access rights to particular user.

4.1 System Model
Similar to [9] our proposed model has one or many cloud server, one or many clients and an optional third party auditor shown in figure 1.

i. Cloud server is an entity which is managed by Cloud Service Provider to provide promised services to cloud users. These services can be storage space, data security and computation resources.

ii. Auditor could be an autonomous body and self-directed, unaffiliated with the company being audited or captive auditors, as well as some are elected public officials [10]. The auditor can also be used to improve services of cloud by constantly monitoring it. Making profit through serving customers is the main objective of a CSP.

iii. A cloud user (we will not differentiate cloud user, data consumer and client hereafter) an entity, who has data to be stored in the cloud and relies on the cloud for data storage and computation, can be either enterprise or individual customers.

There are mainly two parts of cloud server in our setup; first one is the data server that store Access Right List (ARL), user attributes, file attributes and other access policies. While the other one is the file server which is used to store files, which are uploaded by cloud user.

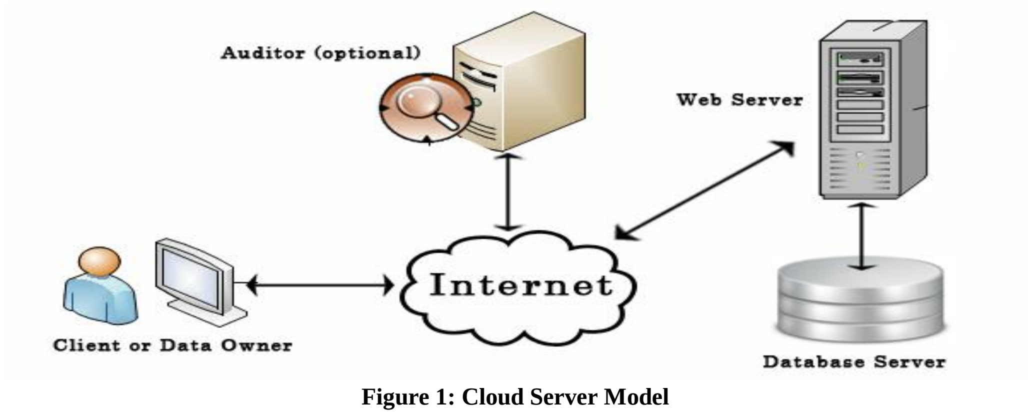

Figure 1: Cloud Server Model

4.2. Activities in System

In proposed scheme, attribute based Access control is combined with Role based access control to create access permissions for files which are stored in encrypted format by using attribute based encryption method. There are several activities in cloud as follows:

4.2.1. User Creation

When new users arrive to the cloud server then our scheme requests him to provide their required information. This information stored as an attribute set into the data server in encrypted form. The scheme use Attribute Based Encryption to encrypt user's information. This encryption prevents unauthorized access of their information by third party and even from Cloud Service Provider (CSP). This information will used for user authorization purpose and will used for creation permission set to use available services.

4.2.2. Permission Granting

After creating a user on cloud server, the scheme decides to grant permission set to this user according to their selected plan for using services. This permission set contain access policies to use services such as how much storage space will be provided, what bandwidth will be allotted etc.

4.2.3. User Authentication

Making decisions for user authorization is a continuous process as a cloud environment is dynamic. Before accessing the data from cloud server user verification is necessary but not sufficient [4]. The continuous verification for each request made by user is required to achieve a fine grained access control. For user verification our proposed scheme make decisions that user is allowed or not to access the requested service. This decision making process starts from the verification of user identity at the time of user arrives and continue to for perform each activity in cloud and ends with aborting by user. A log for these activities is recording simultaneously. The time period from identity verification to aborting by user is called a session. Furthermore, the figure 2 has shown these authentication steps:

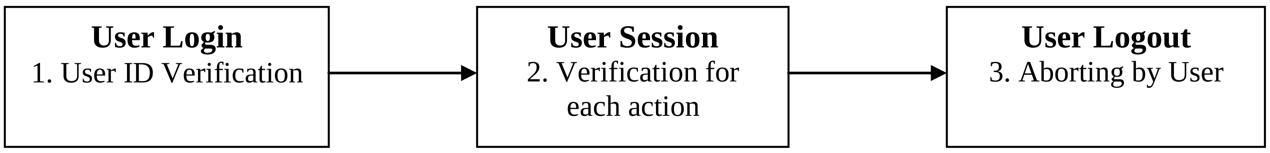

Figure 2: Three steps of user verification

NOTIFICATION	DESCRIPTION
Req()	User Request with essential parameters.
U_{id}	User ID of requesting user.
SecKey	Secrete key or password of requesting user.
AA	Authentication agent.
AP	Access control policies or access rights according to selected plan.
R_{id}	Role of user.
OP	Operation or action to be performs.
F_{id}	File requesting by user to perform action.
SM	Session manager (server software).
ALC	Access level code (defined in Data Access Right Model)
S_{id}	Session for U_{id}.
FAA	File authentication agent.
F_w	File to be uploading.
UNQ_{id}	File unique id.
F_{id*}	File after modification.
SL	List of user for file sharing.

These authorization steps can be describes by following algorithms:

Algorithm 1: User Id Verification or pre authorization process.

```
1. User submit request Req($U_{id}$, SecKey , D).
2. AA check for $U_{id}$ existence.
3. If $U_{id}$ exist extract value of SecKey correspondent to this $U_{id}$ and go to 5th step.
4. Else return to submit new request.
5. AA performs validation and matches SecKey provided by requesting user and SecKey exist into database.
6. If SecKey match verified AA permit to enter into cloud and load correspondent role of user and go to 8th step.
7. Else return to submit new request.
8. AA extracts AP from database for this applied role of user.
9. If everything is correct then user logged in successfully and a session will start.
```

Algorithm 2: User On-going authorization process.

```
1. User submit request Req($U_{id}$, $R_{id}$, OP, $F_{id}$).
2. AA validates for $U_{id}$ status, user logged in or note from the SM.
3. If $U_{id}$ logged in then go to 5th step.
4. Else access denied and returns to submit request again.
5. AA extract access level attributes for $U_{id}$ correspondent to R.
6. AA check and validate whether $U_{id}$ has sufficient permission or not to perform OP.
7. If $U_{id}$ has permission to perform OP then go to 9th step.
```

```
8.  Else access denied and returns to submit request again.
9.  AA check $F_{id}$ exist or not on server.
10. If $F_{id}$ exists go to 12th step.
11. Else return to request another file.
```

```
12. Extract access rights for correspondent ALC which is available as an attribute of $F_{id}$.
13. AA checks whether access rights allow user to perform OP on $F_{id}$ or not.
14. If $U_{id}$ is allowed then OP executes.
```

Storing a File: User who is verified for authorization and has sufficient rights to write can store a file on cloud server. Our proposed scheme has a unique and stepwise process to complete file storing procedure. Algorithm 2.1 shown file storing steps as follows:

Algorithm 2.1: Storing a file on cloud server.

```
1.  User submit request Req($U_{id}$, $F_w$, $R_{id}$).
2.  AA extracts access rights from $R_{id}$ for $U_{id.}$
3.  FAA extract and validate file attributes whether $U_{id}$ possess authority to upload this file or not.
4.  If system allows uploading this validated file then go to 6th step.
5.  Else return to submit new request.
6.  Generate $UNQ_{id}$ for this file to check whether the same file is exists or not in cloud storage server.
7.  If file already exist in storage then check file's owner detail whether the file owner is requesting user or not.
8.  Else go to 11th step.
9.  If file owner and requesting user are same then through exception and return to submit new request.
10. Else go to 11th step.
11. Generate $F_{id}$ and insert record into file storage database.
```

The uploaded file is stored in file storage server after completing algorithm 2.1. Format of stored file is shown in figure 2 as follows:

F_{id}	UNQ_{id}	ALC	F_{data}	F_{uid}	F_{info}

Figure 2: Stored file format

Here,

F_{id} = File's serial id into file storage database.

UNQ_{id} = File's unique id which is same for another file which have same attributes.

ALC = Access level code.

F_{data} = Content of file.

F_{uid} = File's owner id.

F_{info} = Attribute of files such as file size, file type, file upload date and time etc.

Algorithm 2.2: Accessing a file from cloud server.

```
1.  User submit request Req($U_{id}$, $F_{id}$, $R_{id}$).
2.  AA validate for $U_{id}$ status whether user logged in or not from SM.
3.  If $U_{id}$ logged in then go to 5th step.
4.  Else return to submit new request.
5.  FAA extracts file access rights for $U_{id}$ and validate whether file allows to access or not to $U_{id}$.
6.  If file allows to modification then go to 8th step.
```

```
7.  Else return to submit new request.
8.  FAA generates a link for file $F_{id}$.
```

Algorithm 2.3: Modifying a file on cloud server.

```
1.  User submit request Req($U_{id}$, $F_{id}$, $R_{id}$).
2.  AA validate for $U_{id}$ status whether user logged in or not from SM.
3.  If $U_{id}$ logged in then go to 5th step.
4.  Else return to submit new request.
5.  FAA extracts file access rights for $U_{id}$ and validate whether file allows to access or not to $U_{id}$.
6.  If file allows to modification then go to 8th step.
7.  Else return to submit new request.
8.  User submit request to save modification Req($U_{id}$, $F_{id^*}$, $R_{id}$).
9.  FAA check whether $U_{id}$ possess right to save modification or not.
10. If $U_{id}$ allows then go to 12th step.
11. Else return to submit new request.
12. Generate new $UNQ_{id}$ and replace old $UNQ_{id}$ and file for $F_{id}$.
```

Algorithm 2.4: Sharing a file with another cloud member.

```
1.  User submit request Req($U_{id}$, $S_{id}$).
2.  AA validates for $U_{id}$ status, user logged in or not from the SM.
3.  If $U_{id}$ logged in then go to 5th step.
4.  Else access denied and returns to submit request again.
5.  FAA extracts file access rights for $U_{id}$ and validate whether file allows to share or not to $U_{id}$.
6.  If file allows to share then go to 8th step.
7.  Else return to submit new request.
8.  User submit request Req($U_{id}$, $F_{id}$, $R_{id}$, SL, ALC).
9.  File alias sent to each user mention in SL.
```

Algorithm 3: User Logout and Session closing or post authorization process.

```
1.  User submit request Req($U_{id}$, $S_{id}$).
2.  AA validates for $U_{id}$ status, user logged in or not from the SM.
3.  If $U_{id}$ logged in then go to 5th step.
4.  Else access denied and returns to submit request again.
5.  AA closes $S_{id}$ and exit.
```

5. ANALYSIS OF FRAMEWORK AND SCHEME

In order to achieving fine grained access control for file storage cloud our scheme has following efficiencies:

1. ***Dynamic Security:*** This feature is main contribution of our proposed scheme. The proposed access control model provides ongoing authorization as well as pre and post authorization mechanism, which ensure data owner that their data cannot be access by any unauthorized person.

2. ***Information Confidentiality:*** Our scheme provides confidentiality policy secure user's personal information, only authorized user can read or use this information.

3. ***Cost Reduction:*** Our scheme eliminates manual policy management. Scheme maintain access right

list dynamically and removes manual identity verification and user authorization. Hence a huge amount of cost would have been reduced.

4. In our proposed scheme each file has individual access rights set for a particular user. More specifically a single file can has multiple access right sets for different users. This mechanism removes role dependencies of file access control.

5. We proposed a novel method for file storage management. In this mechanism our scheme provides a unique identification for a particular file. This unique identity is same for file which has same properties such as storage size, file type and file contents. This scheme removes file redundancy in file storage server.

6. CONCLUSION AND FUTURE WORK

In this paper, we introduced a unique combination of Role Based Access Control and Attribute Base Access Control with Attribute Based File Encryption method. Our proposed protocol provides both access controls strength simultaneously in a cloud environment. All computations are done at server end which degrades the performance of cloud server. Reducing this server load can be an area of research. Our proposed work can be extended to instant file access for anonymous user who is not register on our server.

7. REFERENCES

[1] Barrie Sosinsky “Cloud Computing Bible”: 2011 by Wiley Publishing, Inc.

[2] C. Baun, M. Kunze, J. Nimis, and S. Tai “Cloud Computing (2nd edn.)” : Springer-Verlag Berlin Heidelberg 2011

[3] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou “Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing”: INFOCOM, 2010 Proceedings IEEE, 2010

[4] Habiba, Mansura, Md Islam, and A. B. M. Ali. "Access Control Management for Cloud." Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on. IEEE, 2013

[5] Indrajit Ray and Indrakshi Ray “Trust-Based Access Control for Secure Cloud Computing” : K.J. Han et al. (eds.), *High Performance Cloud Auditing and Applications*, DOI 10.1007/978-1-4614-3296-8 8, © Springer Science+Business Media New York 2014

[6] Zhiguo Wan, Jun’e Liu, and Robert H. Deng “HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing” : IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 7, NO.2, APRIL

[7] Lingyu Wang, Duminda Wijesekera, and Sushil Jajodia “A Logic-based Framework for a Attribute based Access Control” : Proceedings of the 2004 ACM workshop on Formal methods in security engineering. ACM 2004.

[8] Ravi Sandhu, David Ferraiolo and Richard Kuhn “The NIST Model For Role Based access Control: Toward a Unified Standard” : ACM workshop on Role-based access control. Vol. 2000. 2000.

[9] Wang, Cong, et al. "Toward secure and dependable storage services in cloud computing." Services Computing, IEEE Transactions on 5.2 (2012): 220-232.

[10] Ling Li Lin Xu Jing Li Changchun Zhang “Study on the Third-party Audit in Cloud Storage Service” International Conference on Cloud and Service Computing.

[11] J. Anderson, “Computer Security Technology Planning Study,” Airn Force Electronic systems Division, Report ESD-TR-73-51, 1972.