Circuits and Systems, 2016, 7, 1113-1119
Published Online May 2016 in SciRes. http://www.scirp.org/journal/cs
http://dx.doi.org/10.4236/cs.2016.77095

RoBAC—A New Way of Access Control for Cloud

G. Krishnamoorthy[1], N. UmaMaheswari[2], R. Venkatesh[3]

[1]Department of Information Technology, R.V.S College of Engineering, Dindigul, Tamil Nadu, India
[2]Department of Computer Science and Engineering, P.S.N.A. College of Engineering & Technology, Dindigul, Tamil Nadu, India
[3]Department of Information Technology, P.S.N.A. College of Engineering & Technology, Dindigul, Tamil Nadu, India
Email: pnggkrishnamoorthy@gmail.com, numamahi@gmail.com, rlvenkatesh@gmail.com

Received 29 March 2016; accepted 24 May 2016; published 27 May 2016

Abstract

Access control has made a long way from 1960s. With the advent changes of technologies pertaining to location transparency in storage of data, there arises different access control scenarios. Cloud storage, the predominant storage that is being in use currently, also paves way to various access control problems. Though there are various access control mechanisms such as RBAC, ABAC, they are designed on the user's perspective such as the role held by the user or other attributes assigned to the user. A new access control mechanism called object relationship based access control (RoBAC) has been developed based on the relations held among the users. The policy decision of access control is based on the relationship among the classes followed in the Java programming. Results have shown that this model best suits various scenarios in the cloud environment, and it also shows that the time for making decision either to allow or to deny is reduced compared to the existing system.

Keywords

Cloud, Access Control, Class Relations, Roles

1. Introduction

The way in which the cloud services are provided to the end user changes the storage method of the organizations and individuals, with the increased trend of organizations moving towards the cloud infrastructure under different cloud models such as public, private and hybrid cloud; we concentrate on access control model in private cloud. We consider a private cloud which is deployed in a company's data centre and shared by the workers

How to cite this paper: Krishnamoorthy, G., UmaMaheswari, N. and Venkatesh, R. (2016) RoBAC—A New Way of Access Control for Cloud. *Circuits and Systems*, **7**, 1113-1119. http://dx.doi.org/10.4236/cs.2016.77095

of the organization. The assumption that we have made here is a single tenant assigned for the whole organization; this assumption is taken into consideration since cloud has a multitenant architecture. We formulate an access control mechanism and see how well it adapts to the specified cloud environment. An access control method is one which accepts or denies an access to a particular resource [1] [2]. All the access control policies that are in practice are based on the various policy specification languages. Extensible access control language (XACML) [3] based on xml is one such language that has been used widely; another such policy specification language is ponder [4] which is based on the object oriented technology. These kinds of languages need a new learning curve and the access control policies are implemented separately; the policy decisions of access privileges are designed based on the various relations that exist between classes in the java programming language, *i.e.* the concept of a particular programming language is being used as policy specification language. With the advent usage of automatic code generation and web services based development, our work will be of a starting point to this type of policy specification.

2. Related Work

Access control mechanism had its beginning in the form of Access matrix, which is followed by lot other mechanisms, the major methods that were and in practice are as follows, discretionary access control [2] is the one in which the owner of the data have all privileges over it and also he is capable of transferring the privileges he posses to others, in mandatory access control [5] access controls are managed by the administrators, objects and subjects are assigned with are security levels and based on it access is grant or denied. The next major Access control mechanism that comes in to use is RBAC, role based access control which is followed by Attribute Based Access Control (ABAC), RBAC [6] [7] is the method in which the access rights are provided on the basis of the roles held by the user. There are many variations of RBAC such as Access controls based on time [8], based on location [9], in order to handle roles across different organization, Role and Organization Based Access Control (ROABC) is developed [10] [11] deals with access control model based on the relationships. User to user relation and relation among users in social networks is handled in [12]. Similarly there are various ABAC models [13], all flavors of both RBAC and ABAC depend on policy based access control, policies specify the rules based on which the access decision is made. The standard policy specification language that is in use XACML, we are providing an alternate method of this policy generation. The following sections explain the proposed work and result obtained.

3. Proposed Work

The proposed work is explained with an example scenario, this would facilitate to understand the concept much better.

3.1. Example Scenario

Consider an educational organization where there are various departments, each department has a specific number of students say 60. Consider the educational institution has a method for mentoring the students called faculty Advisor scheme, for each set of 20 students a Faculty Advisor will be allocated, to better understand the scenario the following hierarchy would help. The institution has a head, named principal, whose hierarchy is followed by the head of the departments which in turn followed by faculty (staff). Each faculty is responsible of 20 students.

The details of the particular set of students can only be accessed by their corresponding faculty Advisor and remember these students belong to a particular department.

Here the access control policy followed is only the faculty advisor of a particular student can access his/her details.

Solution in RBAC and ABAC

In case of the role based access control this can be achieved through assigning a role called faculty advisor to the particular user, but this would lead to the problem that any user with the role faculty advisor can access this the user may belong to a different department, a better method would be attribute based access control, in which an attribute is added to the user role, for example an attribute named department would solve this issue. So in case of ABAC a role called faculty advisor is created and then an attribute department is created for that user role, the condition would be if the user is a faculty advisor and has a intended department name in the department attribute, he would be allowed to access the data.

The thing that has to be considered here is a role called staff has to be created and a role called faculty advisor should also be created since the entire faculty in the department cannot act as advisors, some have to play the dual role, the faculty advisor is inherited from staff and a set of students should be associated with a particular faculty advisor.

Analysis shows that the solution of the above specified scenario can be achieved with the class relations such as association, composition, generalization, dependency in java. Based on this a new policy mechanism is created. The users and the resources are mapped to the classes and access control decisions are based on the relationship between the classes, a simple example for the above specified problem is as follows

If the user wishes to access a particular student record, the following conditions should be met

- The user should be a staff.
- The staff should be a faculty advisor.
- The faculty advisor should belong to the department to which the student belongs to.
- The faculty advisor should be associated with that student.

Relations of the above specified conditions

1) The user should be an instance of faculty advisor class-realization.

2) This faculty advisor should be inherited from the staff class which in turn should be associated with the department class to which the student belongs—generalization and association.

3) The faculty advisor should be associated with that student—faculty advisor object instance should have a association relation with the student instance—association.

Sample code for depicting the association relation between the student and the faculty advisor is as follows

```
Class student
{
....
}
Class Facultyadvisor
{
        Student s[20];
        .........
}
```

The above relation specifies that the faculty advisor and student are in association relation. **Figure 1** depicts this relation.

3.2. Implementation Details

Based on the above said requirements an access control model is created based on the class relations in Java. This model would be more suitable for scenarios where a specified relation such as association is needed, as in our example, 20 students are associated with a particular faculty advisor. Another example would be a particular set of patients associated with a particular doctor in a multi specialty hospital. The following sections explain the various stages in the process.

3.2.1. Role Object Creation

A class hierarchy is first created for the organization or concern to which the access control security must be implemented; class hierarchy is based on the roles and relations between the roles. Every role that has been defined is generated as a class and the users assigned with the roles are generated as the class instances and stored with the concept of object serialization. The contents (for example, files) that are to be secured are also considered as objects.

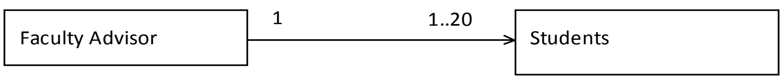

Figure 1. Example association relation.

3.2.2. Defining Rules

When the data objects are created it is embedded with the details of the object instances that can access it is also saved in a separate xml file.

Sample Access rule for accessing the file of student1

```
<fileobject>student1</fileobject>
<accessclass> class name </accessclass>
<relntoaccessclass>
                <relation>Association</relation>
                <relation>Generalization</relation>
                    .........
                    .........
</relntoaccessclass>
```

So when a user intends to access a particular resource the xml file would be checked for the conditions and based on this access control decision is made either to allow or deny.

3.2.3. Decision Making

Access requestor is a one who wishes to access the data. The requestor will send a request to access the intended data. The request consists of <requestor name, Data to be accessed>. This request will be in its native form, the work of the request handler is to convert the native format to an object *i.e.* the object type is identified; and this type is send to the decision maker.

Example request consists of <Bob, Student23>

The Request Handler identifies the various object instances corresponding to the access requestor and the student and sends it to the decision maker based on the rule from the rule storage for the corresponding student object an evaluation is done and the decision of whether to allow or deny the access request is made.

The following **Figure 2** depicts the architecture of the above explained concepts.

For the educational institution scenario we have taken, the **Figure 3** is the class diagram which depicts the entire role hierarchy as classes. These classes and the relations between them are used for making the decision.

For every access request the relation between the object instance and the data request is found, and if the rule permits such a relation to access the request the request is granted else it is denied.

4. Experimental Results

The experiment is done in a system with Intel i3 processor and 4 GB RAM and 32 bit operating system, Java is used for developing the scenario and since we have used java class relation as a rule creator for access control instead of a separate policy language it's easier to develop a rule engine to make decision. For a comparative study the same process is also implemented with sun XACML instances with the concepts of attribute based

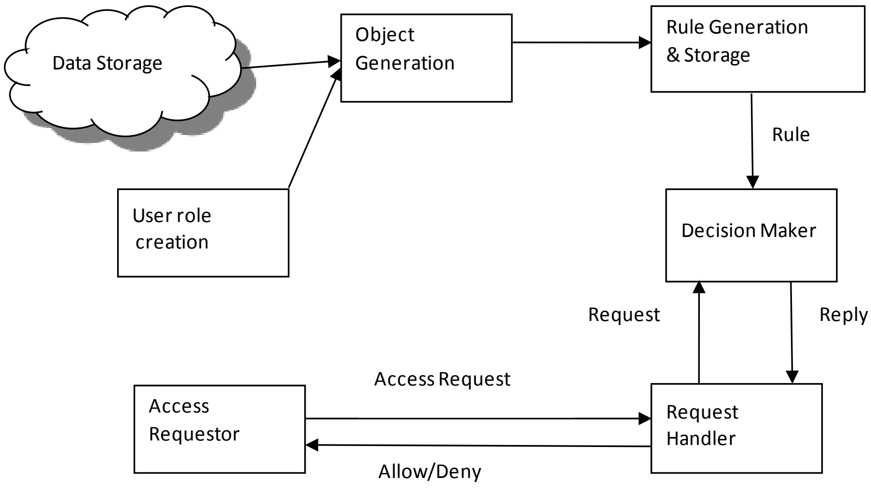

Figure 2. Proposed architecture.

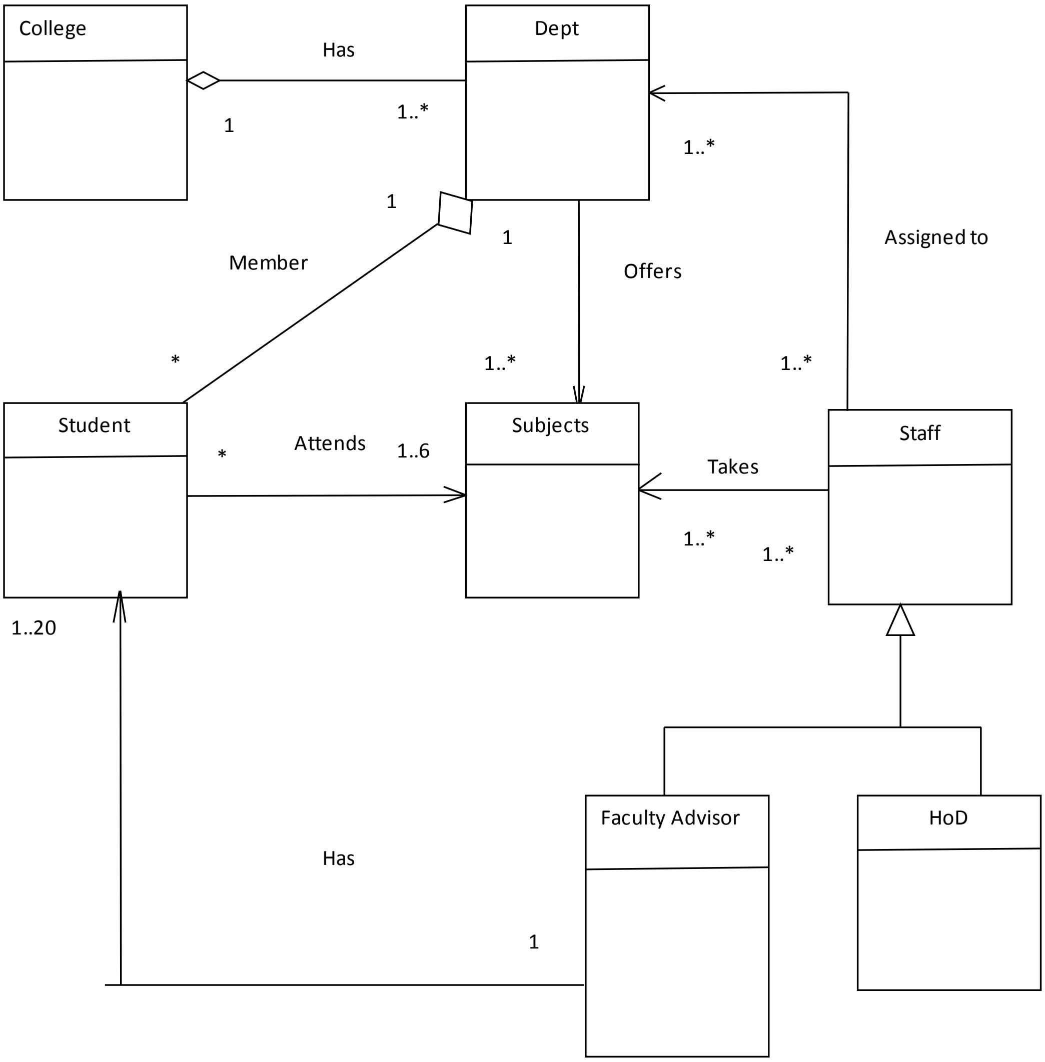

Figure 3. Class diagram of the example scenario.

access control, studies have shown that the time increases with the increase in number of attributes, considering our relation as attributes added to the roles and using XACML policy language, Figure 4 shows the comparative chart of our proposed method, it shows the response time of a request made by a client and the corresponding reply.

Since our access control model is based on the Relationship between the classes, a study of difference in the time of Authorization between two relations is done, as a sample the following graphs (Figure 5(a) and Figure 5(b)) represent the difference in time in terms of mean and standard deviation when an authorization is made based on association and authorization is made based on generalization

Though there is variation in time in which the access control decision is taken it is negligible, and since we have taken two relations (association and generalization) which is the dominant relation in the available scenario, we hope it is not necessary to analyze the time difference of other relations.

5. Conclusion and Future Work

A new access control mechanism is framed based on the existing relations available between classes in Java, This work will be a stepping stone towards creating a standard framework for access control; only preliminary access control mechanism is discussed in this work and our future work includes various other facilities such as generating dynamic rules and making dynamic request handling process. We also look forward to create a

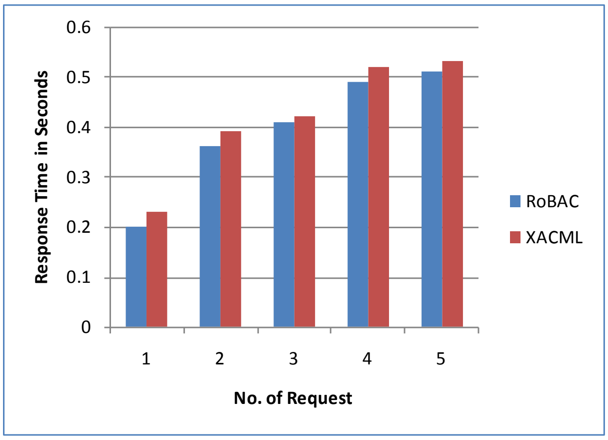

Figure 4. Time comparison between XACML and RoBAC.

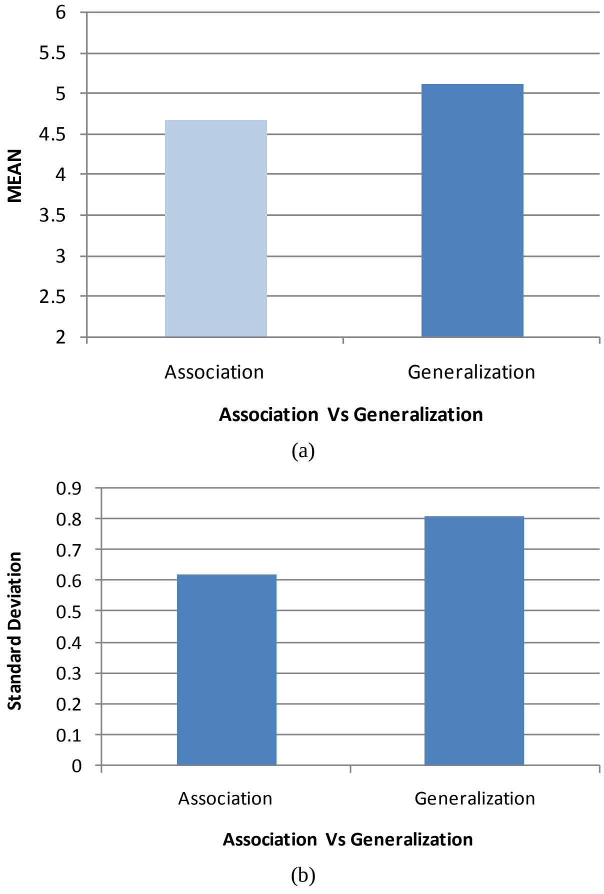

Figure 5. (a) Mean time comparison between association and generalization; (b) Standard deviation comparison between association and generalization.

standard for this kind of access control.

References

[1] Samarati, P. and Vimercati, S.D.C.D. (2001) Access Control: Policies, Models, and Mechanisms. Springer-Verlag, London, 137-196. http://dx.doi.org/10.1007/3-540-45608-2_3

[2] Sandhu, R. and Samarati, P. (1994) Access Control: Principle and Practice. *IEEE Communications Magazine*, **32**, 40-48. http://dx.doi.org/10.1109/35.312842

[3] OASIS Standard (2015). https://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[4] Damianou, N., Dulay, N., Lupu, E. and Sloman, M. (1995) The Ponder Policy Specification Language. *Workshop on Policies for Distributed Systems and Networks*, Bristol, 29-31 January 2001, 18-39.

[5] Sandhu, R.S. (1993) Lattice Based Access Control Models. *IEEE Computer*, **26**, 9-19. http://dx.doi.org/10.1109/2.241422

[6] Ferraiolo, D.F., Sandhu, R., Gavrila, S., *et al.* (2001) Proposed NIST Standard for Role Based Access Control. *ACM Transactions on Information and System Security*, **4**, 224-274. http://dx.doi.org/10.1145/501978.501980

[7] Sandhu, R.S., *et al*. (1996) Role-Based Access Control Models. *IEEE Computer*, **29**, 38-47. http://dx.doi.org/10.1109/2.485845

[8] Hansen, F. and Oleshchuk, V. (2003) SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems. *Proceedings of the* 7*th* *Nordic Workshop on Secure IT System*, Karlstad, 129-141.

[9] Joshi, J., Bertino, E., *et al*. (2005) A Generalised Temporal Role-Based Access Control. *IEEE Transactions on Knowledge and Data Engineering*, **17**, 4-23. http://dx.doi.org/10.1109/TKDE.2005.1

[10] Zhang, Z., Zhang, X. and Sandhu, R. (2006) ROABC: Scalable Role and Organization Base Access Control Models. 2006 *International Conference on Collaborative Computing*: *Networking*, *Applications and Worksharing*, November 2006.

[11] Fong, P.W.L. (2011) Relationship-Based Access Control: Protection Model and Policy Language. *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, San Antonio, 21-23 February 2011, 191-202. http://dx.doi.org/10.1145/1943513.1943539

[12] Cheng, Y., Park, J. and Sandhu, R. (2012) Relationship-Based Access Control for Online Social Networks: Beyond User-to-User Relationships. 2012 *International Conference on Privacy*, *Security*, *Risk and Trust* (*PASSAT*), Amsterdam, 3-5 September 2012, 646-655. http://dx.doi.org/10.1109/SocialCom-PASSAT.2012.57

[13] Jin, X. (2014) Attribute-Based Access Control Models and Implementation in Cloud Infrastructure as Service. Dissertation, The University of Texas at San Antonio, San Antonio.