A New Semantic Role-based Access Control Model for Cloud Computing

Masoud Barati
Department of Computer Engineering, Islamic Azad University- Kangavar branch, Kangavar, Iran
emsbarati@yahoo.com

Mohammad Sajjad Khksar Fasaei
Department of Computer Engineering, Islamic Azad University- Songhor branch, Songhor, Iran
sajjadkhaksar@gmail.com

Soheil Lotfi
Department of Computer Engineering, Kermanshah Science and Research branch, Islamic Azad University, Kermanshah, Iran
soheillotfi1983@gmail.com

Azizallah Rahmati
Department of Computer Engineering, Islamic Azad University- Kangavar branch, Kangavar, Iran
m_aziz_rahmati@yahoo.com

***Abstract*- One of the main topics in Cloud computing is access control. Among the approaches of access control in this environment, semantic role-based access control is an interesting issue. In current methods of role-based access control used in Cloud, when a user has no permission for a specific function, its request may be aborted. In this paper, we want to propose a new semantic role-based access control model being compatible with cloud. In our model, a number of functions will be semantically suggested for a user with a certain role. These offered functions can be perfectly used by the user without rejection of its request. In fact, in our approach, by using of two agents called request agent and permission agent, the permissions will be issued based on the semantic similarity between the function asked by a user having a certain role and the predefined functions being in Cloud environment.**

Keywords-Cloud computing; Role-based access control; Ontology; Semantic similarity function; SPARQL query.

I. INTRODUCTION

Cloud computing provides computing services through the Internet. Cloud services let businesses and individuals to tap software and hardware, which are handled by third parties in remote places. For example, these services include file storages, webmail, social networks, and online business applications. With Cloud computing model, users are allowed to access the information and computer resources from everywhere in a network [1].

The service models of Cloud computing are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In the SaaS model, an application, along with any needed software, operating system, and network are provided. In PaaS, an operating system, hardware, and network are provided, and the customer installs or develops its own software and applications. The IaaS model offers only the hardware and network; the client installs or promotes its own operating systems, and software applications [2].

One of the main issues in Cloud is access control. Access control is divided into: Discretionary Access Control Model (DAC), Mandatory Access Control Model (MAC) and role-based access control model (RBAC) [3]-[6]. Among the existing methods of access control, RBAC is the way of permissions combination relying on permissions defined in the functional role. In addition, RBAC models are more flexible than their mandatory counterparts because users can be assigned several roles and a role can be associated with several users.

It is clear that in the Cloud system, autonomous domains [6] have a separate set of security policies. Hence, the access control Mechanism has to be flexible to support various kinds of policies and rules. With the progress of distributed systems, role based access control has become quite significant [7]-[10].

Among some new approaches in the case of access control in Cloud environment, Sun et al. [11] analyzed existing access control methods and presented a semantic-based access control model which considers semantic relations among different entities in Cloud. Besides, Jung and Chung [12] proposed an adaptive security model for Cloud computing environment. The model is based on the improved RBAC model and adapts the role switching model [6].

In this paper, by using semantic descriptions and ontology, we propose a new semantic model for RBAC used in a Cloud environment. In our model, each user requirements and roles are predefined semantically, and agents as brokers are able to give a permission to a user based on semantic similarity function. In fact, allocating a permission is flexible in our approach, since with determining a threshold, a role with the most similar functions set could be dedicated to user instead of simply exact ones. In fact, a user with a certain role may have no permission to an exact requested function, whereas in Cloud may be a number of functions having the most similarities with the function which user asked. Besides, it is possible that these similar functions could meet the user needs. So, in our method, these similar functions can be found and suggested to user by using semantic similarity function.

 ISBN: 978-1-61208-361-2

The structure of the rest of the paper is as follows: in Section II, we present the definitions and primary concepts. Then, in Section III, our semantic model for access control in a Cloud environment is proposed. Finally, in Section IV, the conclusion is highlighted.

II. DEFINITIONS AND PRIMARY CONCEPTS

A. RBAC

RBAC is a method to limit the system access for authorized users [5]. In this respect, access is the ability of a user to perform a specific task, such as delete, create, or update a record. Roles are defined according to job authority, and responsibility in an organization. In this organization, roles are defined for a variety of job functions. The permissions of performing specific functions are devoted to certain roles. The users of system are assigned particular roles, and through such roles assignments get the permissions of computer to perform a group of specific system functions. As users are not directly assigned permission, but only get them through their roles, management of individual user rights becomes a subject of simply assigning appropriate roles to the user's account. This straight forwarded ordinary operations, such as changing a user, or adding a user's institute [6].

RBAC has three primary rules, namely, *Role assignment, Role authorization, and Permission authorization.* In the first rule, if an individual has been assigned a role, he can use a permission. In the second rule, a person's active role must be authorized for that person. Finally, in permission authorization rule, a man can get a permission only if the permission is authorized for the man's active role. This rule ensures that users could get only permission for which they are authorized [3].

B. Ontology and semantic similarity function

Ontology is a formal structure including information about semantic description of data and a group of concepts and the relations between them. It will be used to retrieve information about user requests. A formal definition of ontology [13] in a certain domain, as follow:

$$O=\{C, \leq c, R, \leq r, A\},$$

where C is a set of concepts, R as set of relations, $\leq$ c is an order on C, and $\leq$ r is a partial order on R. In this definition, *A* is considered as a set of axioms [13]-[15].

Semantic similarity function is used for computing similarity between two concepts. The similarity between two concepts illustrates the degree of likeness between them [16]. Similarity function is defined as: sim(x,y):c×c→[0,1]. The result of this function is a real number in the interval [0,1] that shows the rate of similarity between two concepts x, y. In this case, zero means no similarity and one indicates complete similarity between the two concepts [16]-[19]. We compute semantic similarity based on the method from [18]:

$$sim\,(x,y) = \rho \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1-\rho) \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|} \quad (1)$$

Here, ρ is a real number in the interval [0,1] and it is used to determine the degree of influence of generalizations depending on the hierarchical graph of ontology. Here, we can assume that $is\ \frac{1}{2}$, as there is no difference between sim (x,y) and sim(y,x) in our ontology graph. $\alpha(x)$ is the set of nodes which are upwardly reachable from node *x* in the ontology graph. Also, $\alpha(x) \cap \alpha(y)$ is the reachable nodes which are shared by node *x* and node *y* [20].

For instance, an example of ontology with hierarchical graph is depicted in Fig. 1. It has 7 concepts with *'is a'* relationships.

As indicated in Fig. 1, we define Thing as a root node, and which has sub-nodes including Account, Centralize and Decentralize. Account also includes sub-nodes Short Account, Current Account and Long-term Account.

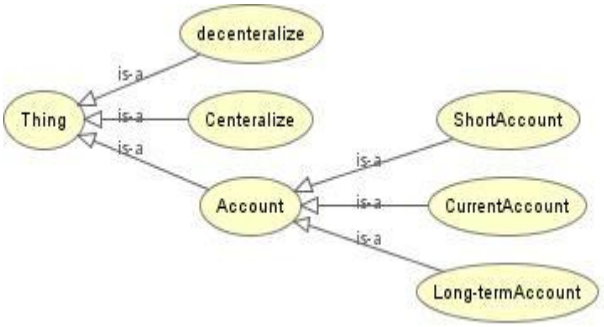

Figure 1. A simple ontology graph of bank account

In case of Eq. 1, the concepts Account and Centralize have 2 reachable upward nodes from themselves. Hence, α(Account)=2 and α(Centralize)=2.
Besides, the similarity of α(Account)∩ α(Long-term Account)=2 is more than α(Account)∩ α(Decentralize)=1.

C. SPARQL

SPARQL [21] is a query language that enables us to retrieve and manage the data saved in Resource Description Framework (RDF) format [22]. The forms of SPARQL queries include a set of triple patterns named a basic graph pattern. In the triple patterns of SPARQL, each of the subject, predicate and object may be a variable. Moreover, SPARQL provides aggregation, sub-queries, negation, and creating values by expressions, and constraining queries with source graph of RDF. The outputs of SPARQL queries could be outcome sets or RDF graphs.

 ISBN: 978-1-61208-361-2

In general, SPARQL graph patterns containing paths are converted to subject-object joins in the SQL [21], and those involving multiple attributes about the similar entity contain subject-subject joins in the SQL.

An example of SPARQL query which models the question of *"What are all the country capitals in America?"* is shown in Fig. 2.

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
     abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
     abc:isInContinent abc:America .
}
```

Figure 2. An example of SPARQL query

A variable is indicated by a "?" prefix, bindings for ?capital and, the ?country will be returned.

III. THE PROPOSED MODEL

A three layers model is presented for our semantic access control model in Cloud computing. As illustrated in Fig. 3, the layers are known as *User Layer*, *Broker Layer*, and *Knowledge Layer*.

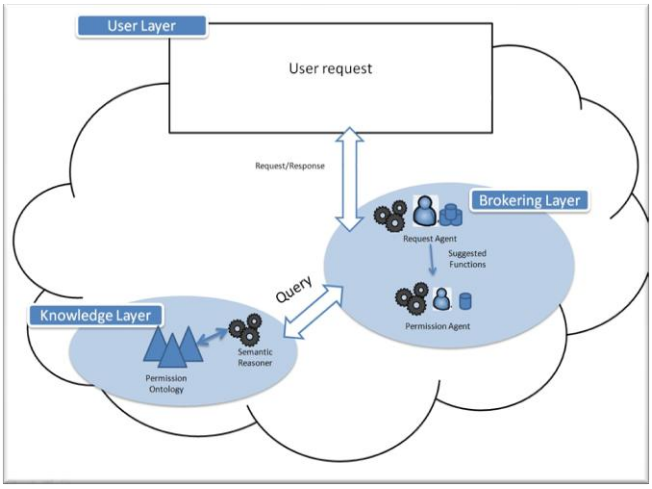

Figure 3. Our proposed semantic model

User Layer

Each user having his role may ask a permission from network to accomplish a function. In this layer, the user request is received and then translated into a format of (*Role, Function*). Following that, the request will be delivered to *request agent*, in the next layer.

Broker Layer

This layer is responsible of getting user requests in the right formats, and issuing permissions for them. In fact, there are two agents called *request agent* and *permission agent* in this layer. The duty of *request agent* is getting the binary set of (Role, Function) from User Layer, and suggesting a sort of functions which are selected based on the semantic similarity function.

To do so, by regarding the ontology graph formed for functions in Knowledge layer, and also by using the semantic similarity function, a matrix of similarities among functions is made by *request agent*. a semantic similarity matrix SIM($n \times n$) can be constructed, as follows:

$$\text{SIM (n} \times \text{n)} = \begin{pmatrix} sim\,(f1, f1) & \cdots & sim\,(f1, fn) \\ \vdots & \ddots & \vdots \\ sim\,(fn, f1) & \cdots & sim\,(fn, fn) \end{pmatrix}$$

Then, this agent based on a predefined threshold (i.e., 0.9), may offer and find more functions having the most similarities with the asked function of user. Following that, it may deliver a number of binary sets of (Role, Function) to permission agent in this layer.

Once Permission agent gets the suggested binary sets of (Role, Function) from request agent, it runs a SPARQL query in the predefined ontology graph in Knowledge layer to search the relationship between the user role and suggested functions. So, the permission of offered functions can be issued, if there are direct relationships between role and functions. Finally, this agent gives the right permissions to the user layer.

Knowledge Layer

In this layer, there is an ontology graph with three primary concepts of permission, roles, and functions. The direct relationships between a role and a function in this graph indicates a permission between that role and function. A general schema of this graph is illustrated in Fig. 4.

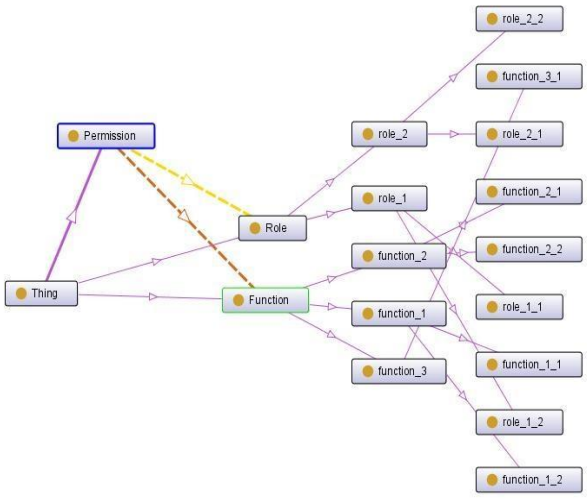

Figure 4. A scheme of ontology graph in knowledge base layer

For example, we assume that a manager in an office wants to calculate his small budgetary computations with an account application A. So, he could join cloud environment with his role, and with his requested function (application A). Then, the user request is semantically translated with the agents in broker layer. In this case, regarding the knowledge layer, should permission agent find the exact application A, then it will issue the permission for the user, otherwise, it tries to find the most similar function for user (e.g., account application B). What is more, this suggested application can properly do the user function.

IV. CONCLUSION AND FUTURE WORK

In this paper, we introduced a new semantic access control model for Cloud computing based on RBAC. In our presented model, the permissions can be assigned to users based on semantic similarity function. In fact, to give a permission, the most similar functions of a role is selected by the agents in broker layer instead of exact ones. So, in our approach, may be found and suggested more than one function for a certain role. Moreover, our model is scalable and it is able to use into different large scale environment.

In future work, we would focus on how we can offer a semantic discovery algorithm to find suggested functions, and we will compare the algorithm with some existing algorithms related our work.

REFERENCES

[1] L. Wang, J. Tao, and M. Kunze, "Scientific Cloud Computing: Early Definition and Experience," in: Proceedings of the 2008 International Conference on High Performance Computing and Communications (HPCC 2008), 2008, pp. 825–830.

[2] L.M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," In: ACM SIGCOMM. Computer communication review 2009. New York: ACM Press,2009, pp. 50–55.

[3] D.F. Ferraiolo and D.R. Kuhun, "Role Based Access Control," Proceeding of 15th National Computer Security Conference, Baltimore MD, 1992, pp. 554-563.

[4] B. cha, J. Seo, and J. Kim, "Design of Attribute Based Access Control in cloud computing," Proceeding of International conference on IT convergence and Security, Springer. 2011, pp. 41-50.

[5] R. Sandhu, "Role-based access control," In M. Zerkowitz, editor, Advances in Computers, vol. 48. Academic Press, 1998.

[6] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," 1996. IEEE Computer, 29(2), 1996, pp. 38–47.

[7] L. Obrst, D. McCandless, and D. Ferrell, "Fast Semantic Attribute-Role-Based Access Control (ARBAC) in a Collaborative Environment," The 7th IEEE International Workshop on Trusted Collaboration (TrustCol 2012), October 14–17, 2012, Pittsburgh, PA, 2012.

[8] S. Ullah, Z. Xuefeng, and Z. Feng, "TCloud: A Dynamic Framework and Policies for Access Control across Multiple Domains in Cloud Computing," CoRR abs/1305.2865, vol. 62, no. 2, January 2013.

[9] M. Amirreza and J. Joshi, "OSNAC: An Ontology-Based Access Control Model for Social Networking Systems, Social Computing (SocialCom)," 2010 IEEE Second International Conference on Social Computing, 20-22 Aug. 2010, Minneapolis, MN, 2010, pp. 751 – 759.

[10] C. Ngo, P. Membrey, Y. Demchenko, and C. Laat, "Policy and Context Management in Dynamically Provisioned Access Control Service for Virtualized Cloud Infrastructures," Seventh International Conference on Availability, Reliability and Security (ARES), 2012.

[11] L. Sun, J. Yong, and G. Wu, "Semantic access control for cloud computing based on e-Healthcare," Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, China, 2012, pp. 512-518.

[12] Y. Jung and M. Chung, "Adaptive Security Management Model in the Cloud Computing Environment," In: 2010 the 12th International Conference on Advanced Communication Technology (ICACT), vol. 2, 2010, pp. 1664–1669.

[13] D. Fensel, "Ontologies: A silver bullet for knowledge management and electronic commerce," Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2003.

[14] H. Stuckenschmidt, "Ontology-based information sharing in weakly structured environments," Ph.D. thesis, AI Department, Vrije University, Amsterdam, 2002.

[15] T.R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," KSL-93-04, Knowledge Systems Laboratory, Stanford University, 1993.

[16] T. Andreasen, H. Bulskov, and R. Knappe, "From ontology over similarity to query evaluation," in: R. Bernardi and M. Moortgat (Eds.): 2nd CoLogNET-ElsNET Symposium - Questions and Answers: Theoretical and Applied Perspectives, Amsterdam, Holland , 2003, pp. 39–50.

[17] O. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity and natural language," Journal of Artificial Intelligence Research, vol.11, 1999, pp. 95–130.

[18] R. Richardson, A. Smeaton, and J. Murphy, "Using WordNet as a knowledge base for measuring semantic similarity between words," Tech. Report Working paper CA-1294, School of Computer Applications, Dublin City University, Dublin, Ireland, 1994.

[19] M.A. Rodriguez and M.J. Egenhofer, "Determining semantic similarity among entity classes from different ontologies," IEEE Transactions on Knowledge and Data Engineering, vol. 15, 2003, pp. 442–456.

[20] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in WordNet," Tech. Report, University College Dublin, Ireland, 2004.

[21] SPARQL Query Language for RDF. W3C Working Draft 4 October 2006. http://www.w3.org/TR/rdf-sparql-query/, 2006.

[22] P. Muster, "Quantitative and Qualitative Evaluation of a SPARQL Front-End for MonetDB," in Department of Informatics, University of Zurich: Zurich, 2007.

 ISBN: 978-1-61208-361-2