International Journal of Information Technology (IJIT), Volume – 1, Issue – 1, August 2012
ISSN 2279 – 008X

Hybrid Access Control Model in Semantic Web

Sonu Verma[1], Suresh Kumar[1], Manjeet Singh[2]

[1]Department of CSE,
Ambedkar Institute of Advance Communication Technologies & Research,
GGSIP University, New Delhi, India

[2]Department of CE,
YMCA University of Science & Technology Faridabad, India
sonu.verma@rediffmail.com, sureshpoonia@yahoo.com, mstomar2000@yahoo.com

Abstract: - *As the demand for data and information management increases, there is also a critical need for maintaining the security of the databases, applications and information systems. Data and information have to be protected from unauthorized access as well as from malicious corruption. With the advent of the web it is even more important to protect data and information as numerous individuals have access to this data and information. Therefore we need effective mechanism to secure the information from unauthorized access in semantic web. Role Base Access Control Model (RBAC) is a traditional access control model & has some pros and cons. To overcome the limitations of RBAC, Attribute Base Access Control Model (ABAC) has been introduced. ABAC proved beneficial and secure over RBAC but it has higher complexity. Both models are not a fine grained access control model. To overcome the limitations of both these models, we have proposed a Hybrid Access Control Model (HACM). This is a model which can overcome most of the limitations of RBAC and ABAC. It can be proved a fine grained access control model. It is a combination of RBAC and ABAC systems. In this paper first we have an introduction of RBAC and ABAC to know the problems with both of them. Then we proposed architecture of HACM followed by its basic approach. We will have an overview of the functioning of HACM step by step which will be expressed by XACML architecture. After that we will have the comparison of RBAC and ABAC with HACM with some points of issues*.

Keywords: - Hybrid Access Control Model, Authorization, Authentication, Dynamicity, Granularity.

1. INTRODUCTION

As the use of internet is growing day by day, more and more critical jobs/processes are run over the web, for example e-government, e-commerce and e-business applications. These are the processes which have to be protected from unauthorized access in an appropriate manner. These commercial information or services should be accessible only by paying customers or to authorize users only. Models like Role Base Access Control (RBAC) were used in such environment to protect the confidential information. But this approach has some disadvantages in large open system where number of users is very high and most of the users will not be known before. RBAC also has been criticized for the difficulty to assigning the initial roles and also for inflexibility in an open environment. RBAC does not support for dynamicity such as time of the day which is needed to be considered at permission allotment time. In large organizations a "role explosion" can result in thousands of separate roles for different collections of permissions [1]. As the RBAC model is not flexible to deal with these requirements in an open or distributed environment. Attribute Base Access Control (ABAC) model has the higher flexibility but there is a deficiency with ABAC that it has the higher complexity in the specification and maintenance of the policies. The access decisions in ABAC are based on attributes that the user proved to have such as clearance level or citizenship [11].

The RBAC model has been considered to be inefficient due to differentiating roles in different context, sometimes proved to be difficult. This resulted in large quantities of role definitions in some cases producing more roles than users. On the other hand RBAC remains somewhat coarse grained while modern requirement are increasingly fine grained. ABAC eliminates the user-role assignment and focuses on the attributes of a user required to grant access. ABAC is believed to be easier to administer than RBAC.

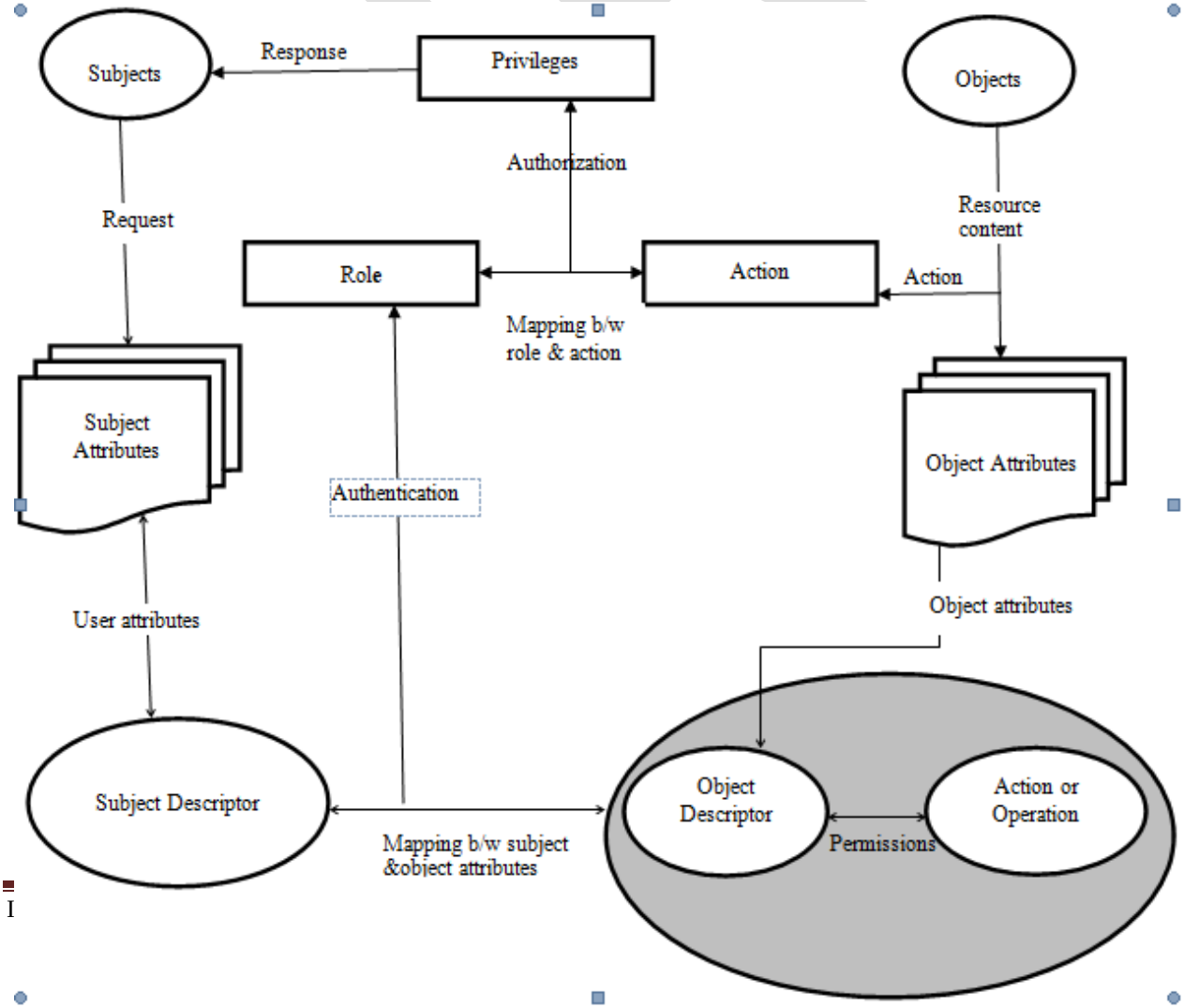
The number of rules is reduced due to the absence of user-role assignment tables. ABAC has become widely used especially for application level AC (Access Control) such as Web Applications [4], and Database Applications [10]. But problem is that the different applications each have their own sets of AC policies. These can conflict with each other or more importantly conflict with general company policies. Sometimes there is the burden of administering several overlapping policy sets. The administration of disparate application level rules also proved to be very difficult due to ABAC's capability to specify complex rules [3].

To overcome the limitation of both the models and to find a fine grained & more suitable access control model we proposed an integrated model which is produced by adding the features of RBAC and ABAC together called Hybrid Access Control (HAC) Model. For this HAC model the permission are not associated directly to the attributes or the roles [8]. In the HACM model we use the attributes for the authentication purposes and assign the roles to the users according to the attributes provided by the user such as dynamic role assignment. Then the authorization process is starts and permission to the access is granted to the user based on the roles which is assigned to the user through the dynamic attributes.

The paper is organized as follows: In section2 we will proposed an integrated architecture of RBAC & ABAC models. This section also explains the way in which Hybrid model will work. Section 2.1 explains the dynamic role assignment through the attributes and 2.2 will explain the granted permission to the roles. Section 3 will provide the XACML architecture of the proposed model and step by step working of the model. In section 4 there will be a comparison of RBAC, ABAC and HACM with some points of issues. Section 5 will conclude the paper and give a direction to the future work.

2. PROPOSED ARCHITECTURE

In This section we introduce a hybrid model of RBAC and ABAC.



I

This is the first phase of the proposed hybrid architecture in which the user request for an access along with his subject attributes. A role is assigned to the user based on the attributes provided by the user.

2.2 ACCESS GRANTING

This is the second phase of the proposed hybrid architecture. In this phase the user is granted or denied for the requested access based on the role & the permissions adjacent to the roles of the user.

3. BASIC APPROACH

For the basic approach of the proposed model there are two points of the policy specification. A point has the policy about the authentication and the other point will provide authorization policies for the user. For this purpose we extend the architecture defined in the XACML specification [6] with semantic web techniques. We extend the architecture by adding the permission management point (PMP) to the architecture. The PMP will decide the gran/deny for the access. PMP map the role and the action request and will decide. Our extension is highlighted by bold labels and gray shading in figure2.

An access control decision is performed according to the following steps:

1.) The Policy Administration Point (PAP) provides the policies to the Policy Decision Point (PDP) created by the security administration.
2.) The user (Access Requester) send s resource request to the Policy Enforcement Point (PEP).
3.) PEP sends this request to the context handler. The request contains the user, resource and the environment attributes.
4.) The context handler creates XACML request for the same and sends it to the PDP.
5.) In case the PDP needs additional subject, resource and environment attributes, they are requested form the context handler.
6.) The context handler requests those attributes from a Policy Information Point (PIP).
7.) The PIP collects the requested attributes of possible from the subject, resource and environment.
8.) The PIP delivers the attributes back to the context handler.
9.) If some attributes requested by the PDP are still missing, the attributes are sent to the inference engine.
    a. The inference engine combines the attributes with ontology delivered by the Ontology Administration Point (OAP).
    b. The inference engine delivers the complete set of attributes to the context handler [6].
10.) The context handler provides these attributes to the PDP.
11.) PDP assign the role and authentication to the user according to the attributes which are provided by the context handler.
12.) Context handler sends the user request to the Privilege Management Point (PMP).
13.) PMP sends a role query to the context handler.
14.) Context handler sends the role responsibility to the PMP (the role which is assigned by the PDP).
15.) PMP attaches the resource context to the request.
16.) The PMP evaluates the policy based on roles and action permission with them and send the response context back to the context handler.
17.) The context handler translates the response context back to the native format of PEP and forwards it to the PEP.
18.) The PEP satisfies the possible obligation e.g. if access is granted the PEP allows for it otherwise access is denied.

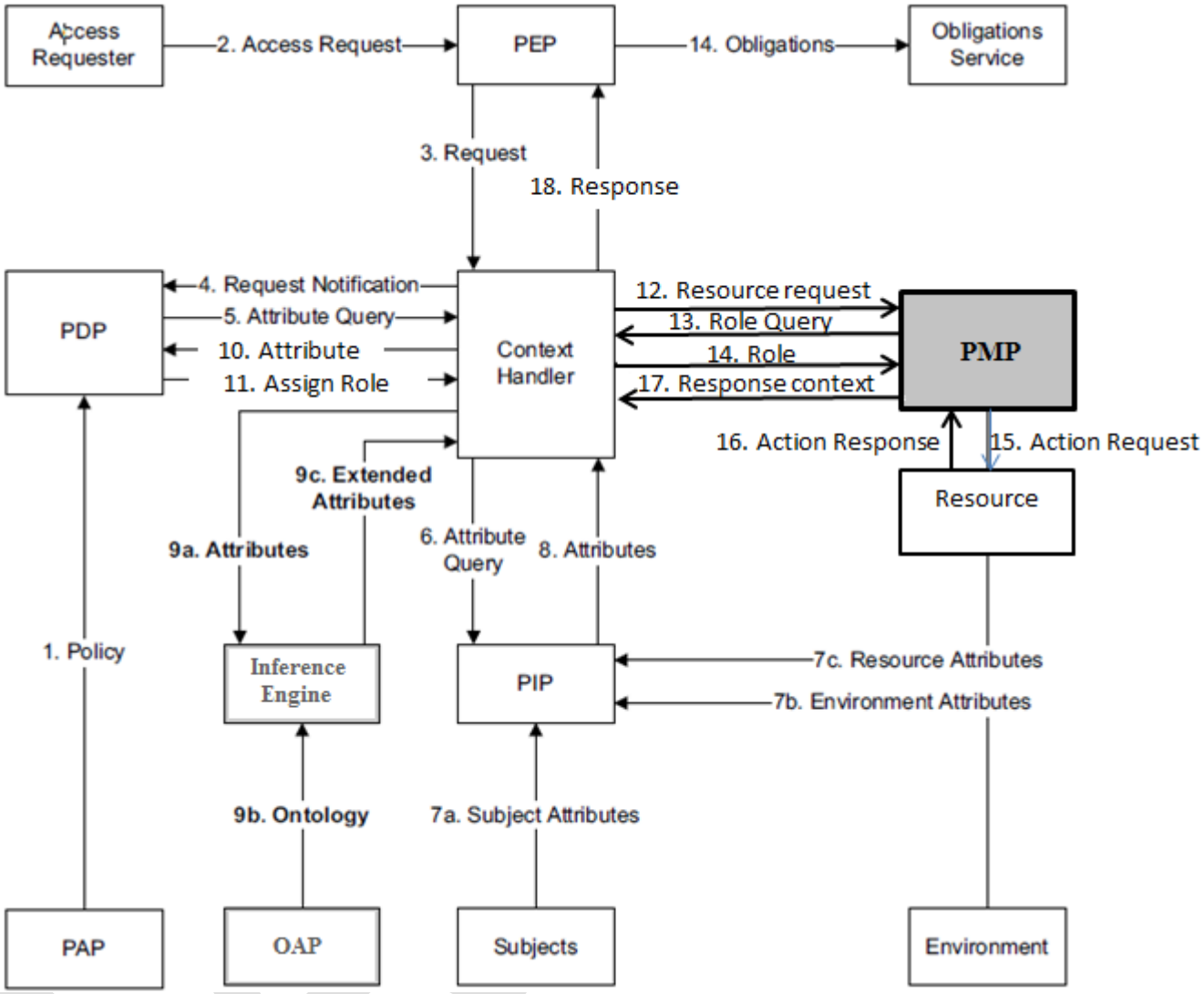


Figure 2: Extended XACML Architecture

4. COMPARISON ISSUES OF THE ABAC, RBAC AND HACM

There are some points of comparison issues between ABAC, RBAC & HACM. These are mentioned in the following comparison table.

Comparison issues explanation:

4.1 DYNAMICITY

A RBAC model has not this feature because the roles and permission with them are assigned to users statically in user's domain and in an dynamic semantic web environment it is very difficult to assign a role to users which are not known in advance [2]. Whether ABAC is supportive in dynamic environment because the user attributes are poses at the time of request and the authorization decisions is made according to the attributes. But HACM has this feature because it also uses the attributes like ABAC at the same time of request and a role is assigned to the user dynamically based on the user provided attributes.

4.2 GLOBAL AGREEMENT

RBAC not support the global agreement because the permission to the user roles are assigned in the local domain whether ABAC supports global agreement due to domain to domain interaction and sharable user attribute database. HACM inherits this feature from ABAC and also support the global agreement feature.

Issue	RBAC	ABAC	Hybrid
Dynamicity	No	Yes	Yes
Global Agreement	No	Yes	Yes
Flexibility	No	Yes	Yes
Simplicity	Yes	No	Yes
Authorization	Locally	Globally	Both
Granularity	Low	Moderate	High
Manageability	Good	Better	Much
Trust	Locally	Globally	Both
Confusing deputy	No	Yes	No
Changing	Complex	Simple	Simple
Policies	Simple	Complex	Simple
Error Prone	No	Yes	No
Fine Grained	No	No	Yes

Table1. Comparison issues between ABAC, RBAC & HACM

4.3 FLEXIBILITY

RBAC model is not flexible in the open and distributed environment due to its static nature. But ABAC is much more flexible due to its dynamic nature in an open and *distributed* system. In HACM, this feature is also inherits from ABAC. As the ABAC model is more flexible in an open and distributed environment. HACM is also more flexible in an open and distributed environment.

4.4 SIMPLICITY

RBAC is simple and easy to use access control model in which permissions are denoted to the roles statically according to the static /predefined policies. On the other hand ABAC is very complex due to its flexibility, heterogeneity of user attributes, global agreement and sharing. For these features the policy specification and maintenance makes ABAC very complex. HACM is more simple than the ABAC and RBAC models. HACM model has the simplicity because the complexity of the policy specification and maintenance is divided into two parts. One in authentication and the other is authorization. It manages the policies for these two at the different management point. This is comparatively simple & easy for policy specification & maintenance for the security administrator.

4.5 AUTHORIZATION DECISION

In RBAC authorization decision is in advance at the time of permission role assignment in local domain. But in ABAC authorization decision are made globally according to the user credential provided dynamically. The authorization decisions are more secure in ABAC as Access decisions directly based on subject- and object attributes: In this scenario subject- and object-specific attributes are directly used to render access decisions. In particular, we specify policy rules that define which attributes and attribute values are needed by a subject to access an object (of course, different applications domains may require different attribute sets). For example, a policy rule may specify that certain subject and object attribute values must be equal to grant a specific access request [7]. This authorization decision is also forwarded at the time of domain to domain interaction. In HACM the authorization decisions are dynamic because it authorizes the user based on her role which is assigned to him according the attributes provided by the user dynamically.

4.6 GRANULARITY [2]

RBAC has low granularity due to its local domain and user has least privileges. On the other hand ABAC has the high degree of granularity due to centralize user attributes database. ABAC has more privileges. In an service oriented environment or open distributed environment services will be invoked by a large number of temporary subjects, and at the same time authentication and authorization need to cross several security domains frequently, raising new demands for access control [8]. Our proposed model HACM model fulfills these requirements as it makes access control decision depending on authorization and authentication process at the same time. It combines the features of both the models RBAC & ABAC. On the other hand HACM provides the privileges based on the attributes provided by the requester. So more the attributes, more the privileges & higher the granularity.

4.7 MANAGEABILITY [2]

Manageability in RBAC is simple because the subject has the permission to the action according to his role and the policies are made for the roles actions and permissions. ABAC has two advantages in terms of manageability, first to derive access control information from same database, second like RBAC simplifies assignment and revocation of permissions [9]. Sometimes management of the user's attributes is become problematic due to the heterogeneity of the user attributes. There is another problem to protect the user attributes in centralize database in an open and distributed environment. HACM divides the whole procedure into two parts authentication & authorization procedure, at different management point, which make it easy to manage.

4.8 TRUST

Trust is an important issue in access control mechanism. In RBAC, trust is highly obtained in local domain. In ABAC trust establishment is more difficult due to the global agreement of the attributes in sharable environment. Trust can be compromised in various situations, i.e. in mobile situation that are common in context aware or pervasive computing where the identity of the user is not known before. For example we may wish to grant a role to a visitor or to a first time client without permanently registering the client's data [8]. But in a HACM trust is more prominent due to its two way checking procedure and decision is based on this two way procedure.

4.9 CONFUSING DEPUTY [2]

In RBAC there is no issue for the confusing deputy because the security administrator assigned the roles to the user and allots the privilege to them according to their role. In ABAC, sometimes there are confusing attributes which may create confusion to the administrator or service providers. This occurs due to the heterogeneity of the user attributes. In HACM there is no confusion because the attributes which can be confused the deputy are only used for authentication. Authorization decisions are based on the roles & permissions with them at the PMP.

4.10 CHANGING PRIVILEGES

To change the privilege of a user is very difficult in RBAC system because to change the privilege of a user it is compulsory to change the role of the user. For this there is a need to change the policy specification. On the other hand in ABAC system to change the privilege of a user there is no need to change the user identity or the policy specification as the policies are so defined that a user has the permission according to his/her attributes values. In HACM it is very simple to change the privileges of the user because if there is a change in authentication or authorization process, it only needs to make changes in specific point as both procedures are at different point & policy specification and maintenance is also at different point for different procedures.

4.11 POLICY SPECIFICATION AND MAINTENANCE

In a RBAC system policy specification is not much complex in comparison to the ABAC system in which it is very complex. The reason of the complexity are the flexibility, dynamicity, sharing in open and distributed environment, heterogeneity of user attributes, mismatching of attributes, confusing attributes etc. attribute database, object database protection is another reason for this complexity. Sharing the attributes in open system can create the problem of integrity or privacy of the user information and policy defined for this purpose are much more complex. Policy specification and maintenance is comparatively simple in HACM because the policies for authentication and authorization are defined at different points, which need to specify or maintain separately.

4.12 ERROR PRONE

RBAC is not error prone because all the policies about the roles and adjacent permissions are defined statically. ABAC may be error prone due to its dynamic nature in open & distributed environment. But HACM is not error prone because it uses attributes along with roles for access control decision and the decisions are made at the different point for the different procedure.

4.13 FINE GRAINED ACCESS CONTROL

RBAC & ABAC are not fine grained access control model because both the models make authentication and authorization decision at the same point which may be error prone. But in HACM there is no change for error because it uses two way procedures for access control decision. It authenticates the user at one point based on the user provided attributes. On another point it authorizes the user based on the authentication decision of the first point. Both the procedure executed at the different point and so it can be proved a fine grained access control model.

5. CONCLUSION & FUTURE WORK

In this paper we have proposed an integrated model for access control in semantic web. RBAC and ABAC have been integrated for the proposed model. HACM model overcome the limitation or weaknesses of the RBAC and ABAC over each other. We also have comprised the three models (RBAC, ABAC & HACM) with some emerging points of issues. For future work we will implement this proposed architecture to find out a working model of it and to find out a fine grained access control model in semantic web.

6. REFERENCES

[1] D. Richard Kuhn, Edward J. Coyne, Timothy R. Weil, "Adding Attributes to Role-Based Access Control", IEEE Computer Society, JUNE 2010.

[2] Alan H. Karp, Harry Haury, Michael H. Davis, "From ABAC to ZBAC: The Evolution of Access Control Models", HP Laboratories-2009-30.

[3] Bernard Stepien, Stan Matwin, Amy Felty, "Advantages of a Non-Technical XACML Notation in Role-Based Models", 2011 Ninth Annual International Conference on Privacy, Security and Trust.

[4] G. Bayer, D. Sengupta, T. Wang, A. Vogt, "PrplAc: Attribute-based Access Control in PRPL for Fine Grained Information Sharing using Semantic Web", Technical Report, http://senguptas.org/Documents/cs343-PrplAc.pdf.

[5] Sonia Jahid, Imranul Hoque, Hamed Okhravi, Carl A. Gunter, "Enhancing Database Access Control with XACML Policy", poster at 16th ACM Conference on Computer and Communications Security, http://www.cs.illinois.edu/homes/sjahid2/pub/myabdac-ccs09-abstract- JahidHOG.pdf, 2009

[6] Torsten Priebe, Wolfgang Dobmeier, Christian Schläger, Nora Kamprath, "*Supporting Attribute-based Access Control in Authorization and Authentication Infrastructures with Ontologies*", First International Conference on Availability, Reliability and Security (ARES 2006), Vienna, Austria, April 2006.

[7] Gerald Stermsek, Mark Stremmbeck, Gustaf Neumann, "Using Subject- and Object-specific Attrubutes for Access Control in Web-based Knowledge Management System"

[8] Guoping Zhang, Jing Liu, "A Model of Workflow-oriented Attributed Based Access Control", I.J. Computer Network and Information Security, 2011, 1, 47-53.

[9] Rakesh Bobba, Omid Fetemieh, Fariba Khan, Carl A. Gunter, Himanshu Khurana, "Using Attributes-Based Access Control to enable Attributes Based Messaging", Annual Computer Security Applications Conference (ACSAC'06).

[10] H. Shen, F. Hong, "An Attribute-Based Access Control Model for Web Services", in Proc. of the 7th International Conference on Parallel and Distributed Computing, Applications and Technologies, pages 74-79, 2006.

[11] N K Prasanna Anjaneyulu Anna, Shaik Nazeer, "Semantic Web Security and Privacy", Journal of Theoretical and Applied Information Technology, Copyright- 2005-2010.