Realizing MAC in RBAC/MAS Reference Model

*K*ommunist Sisai Weldemariam

*A*utomated Reasoning System Division
Center for Scientific and Technological Research (ITC-IRST)
(sisai)@itc.it

ABSTRACT

The environment of multi-agent systems (MAS) provides execution platform for agents needs to make use of security policies and security models, before MAS can be a viable design alternative for large scale industrial applications. One such security model is mandatory access control (MAC). To this end, a role-based access control (RBAC) has been incorporated in MAS environment and therefore a new reference model called RBAC/MAS was introduced. The extension provides a mechanism to enforce organization rules and policies that an agent should respect as it enters and resides in the MAS environment, facilitated by the notion of Agent Coordination Context (ACC). This paper, in particular, proposes an integration of a MAC model on top of an organization infrastructure for open MASs built upon RBAC/MAS reference model. This kind of integration helps for clear realization of mandatory access control in RBAC/MAS in order to reexamine the way to enable and control actions of the agent toward the other agents and resources.

1. Introduction

The implication of the current multi-agent systems (MAS) researches seemed to be a promising paradigm shift toward a noticeable framework for a broad range of industrial applications. To this end, security should be a first class citizen that should be included before MASs can be a feasible design and development alternative. In most cases, however, the inclusion of this concern in (agent-based) software design and development has been considered to be an addendum process.

A mechanism of access control to objects based on the sensitivity of the information contained within the objects and the formal authentication of subjects to access information of such sensitivity is an interesting technique supported by mandatory access control (MAC) model [1, 2]. In other words, in MAC, subjects must be given access rights in an explicit manner and meet the requirements (often specified in terms of subject and object attributes) for access before being exploited. In particular, this kind of approaches makes it possible for better realization of access control in role-based protection systems which allow to specify organization policy in terms of roles, role permissions and relationships between roles.

Adopting and exploring such effective techniques to new abstractions is currently recognized as efficient engineering observation. A role-based access control (RBAC) has been introduced in its classical form in [3] and moved to MAS perspective in [4] - adopting an RBAC-like approach to realize coordination and security concerns, it then further lifted-up in [5]. All those progresses effectively demonstrated that a fair suit of RBAC to the agent-oriented abstraction, as a mechanism to deal with organization security policies and security models (one such security model is MAC) within the MAS environment. Thus, therefore, a new reference model called RBAC/MAS was introduced. To this end, the notion of Agent Coordination Context (ACC) [6] is used as a conceptual border between the agent and the environment. It facilitates the means of binding sessions dynamically to agents as they enter into the MAS environment in order to locally enforce organization policies.

Driven from the above concerns, this paper proposes the incorporation of MAC into the MAS environment, which meant to show the realization of mandatory access control in role-based protection systems. The focus, in particular, is to incorporate MAC into RBAC/MAS for the definition of clearances and

classifications of relevant RBAC/MAS components - roles, role permissions, sessions and relationships between roles. Moreover, the approach can provide a confidential level on the attainment of organization security policy.

The remaining of the paper is structured as follows. In Section 2, mandatory access control and role-based access control are overviewed. The realization of mandatory access control in an organizational infrastructure for open MASs built upon RBAC/MAS model is discussed in Section 3. Section 4 discusses related work on attempting to incorporate security policies in MAS environment. The final observation and remarks are drawn in Section 5.

2. Mandatory Access Control and Role-based Access Control

In this section, we review the concepts on mandatory access control (MAC), role-based access control (RBAC) and its relevant components-roles, roles permissions, inter-role relationships and sessions.

2.1. Mandatory Access Control: An Overview

A mandatory access control (MAC) [1, 2] is a security model meant to limit access to objects derived from the classification of the information contained in the objects and the clearance of subjects (or users) to access information of such classification.

Basically, policy in MAC model is articulated based on security levels associated to each subject and object [1, 2, 7]: *Security classification* - often expressed as top secret, secret, confidential and unclassified, and *Security clearance* labels. While in the case of each subject (user) security clearance is assigned to, security classification label is associated to each object instead. More formally, each subject and object in the system must be labeled by security levels.

In the model, the access request of the subject on the object is granted based on the relation between clearance and classification labels enforced by the following four mandatory rules [7, 8].

Simple Security Property : *"read down" -"no read up"* - a subject can read an object only if its clearance is greater than or equal to the classification of that object. For example, unclassified user cannot read/access any information at confidential levels;

Liberal *-property : *"write-up - no write down"* - a subject can write an object only if its clearance is less than or equal to the classification of that object;

Strict *-property : *"write equal"*- a subject can write an object only if its clearance is equal to the classification of that object;

Simple Integrity Property : *"write down - no write up"* - a subject can write an object only if its clearance is greater than or equal to the classification of that object.

It is important to understand the distinction between the second and third rules. The second rule, for instance, specifies a low subject to write a high object, which means that high organization data may be maliciously or accidentally modified or corrupted or destroyed by low subjects.

However, in an organization built on an open environment such as the case of multi-agent systems, we don't want to allow this kind of situation. For example, it might be the case that a single modification can affect the way agents interact/coordinate with the environment and the other agents and resources. To make strict the way data/information/resources are accessed and possibly to avoid malicious access, the third rule is employed.

2.2. Role-Based Access Control: Overview

The importance of role-based access control (RBAC) has been recognized in different literatures (well written in [1, 2, 3] and overviewed in [4, 9]), proving its capability to represent different kinds of security policies, as well as security models.

A role, in role-based model, is a first class citizen for facilitating access to organization resources - such as information, data, or artifacts in multi-agent systems context, and thus helps to enforce security in an organization system. It can be viewed as, for instance, a task, a function, an encapsulation of rights, duties, and commitments, as well as a specification of access rights at the disposal of a user authorized to the

role [8]. Consequently, a role-oriented security protection grants/provokes a flexible means of managing large numbers of access right on the organization infrastructures. Moreover, a role facilitates access to a given set of objects using the specified laws of access in the associated privilege set. Yet another important remark is that, RBAC is policy neutral [8], meaning that rather than embodying a particular security policy, RBAC is a mechanism for articulating policy. The policy enforced in a particular system is the total outcome of the precise configuration and interactions of various RBAC components as specified by the organization who owned the system. User-role authorization, for instance, is one of the various forms of authorization in role-based protection schemes.

The capability to change policy to meet the changing requirements of an organization is an important benefit of RBAC. This by itself sufficient reason being porting the concepts of RBAC to multi-agent systems (leason learned from [4]), since once agents enter into the environment they can dynamically request a new role other than they owned before possibly with different policy, facilitated by the agent coordination contexts.

A general RBAC reference model is given in [3] comprises the following core elements: user, role, and permission. The model, moreover, consists of additional concepts - session, operation and object. A comprehensive formal semantics and interpretation of the aforesaid components of RBAC, as well as the kind of relationships (user assignment and permission assignment) can be found in [3, 4, 8].

3. Mandatory Access Control in RBAC/MAS

This section describes the modeling aspect of MAC on top of an organization infrastructure built upon RBAC/MAC. Thus, all the description presented here is by considering the RBAC-like reference model using ACCs as depicted in [4].

3.1. Basic Components of RBAC/MAS

The basic elements of RBAC/MAS are agents, roles, ACCs, operations and objects (coordination artifacts). Moreover, the relationship assignments among agent and roles, agent and ACCs and roles to roles, are the essential components of the model. In RBAC/MAS, a role is interpreted as a semantic construct around which organization policy is specified, and then assigned dynamically to agent as it enters into the environment by the ACCs. To do so, two fundamental stages are supported by ACC: ACC negotiation and ACC use. While the former is meant to be negotiated by the agents with the MAS infrastructure to acquire a working session inside an organization, in the latter case the agent requests for an ACC by specifying which roles to activate [5, 4]

3.2. Why MAC in RBAC-like MAS

In particular, the notion of security label in MAC model is strongly related to the concept of a role in RBAC so do in RBAC/MAS. By its very nature, the first concept is specifically constructed to incorporate the policy of unidirectional information flow (such as in a lattice based MAC [8, 10]). This one-directional information flow can be applied for confidentiality, integrity, confidentiality and integrity together, or for group of policies. In particular, the same user cleared to, for example, Secret, can on different occasions authenticate (of course different levels of authentication are possible) to a system (for instance to Voting Machine) at Secret and Unclassified levels. In a sense the user decides what role (Secret or Unclassified) should be activated in a particular session, for instance to vote or configure the voting machine.

The different variations of MAC can be simulated in RBAC/MAS. It turns out that we can realize this by systematically altering the role hierarchy assigned to each agent and defining appropriate constraints in session. This suggests that role hierarchy and constraints (on agent-role assignments or on the roles that can be activated within or across user's session) are fundamental for defining organization policy in RBAC/MAS architecture in order to arrange organization rules as agent enters into the MAS environment. To this end, ACCs represents the notion of session to facilitate both design and runtime activities between agents and the organization environment. Most of the MAC enforcement are in terms of read and write operation based on the security classification and clearance labels. Indeed, however, these are not the only operations involved in MASs environment where a number of com-

plex asynchronous computations take place.

3.3. Realizing and Modeling MAC in RBAC/MAS

Here, we are interested in to enforce the MAC model in RBAC/MAS protection systems. This can be done both at ACC negotiation and ACC use stages. Specifically, agents must be given access rights (such as based on organization policy specified around the model elements) in an explicit manner and meet the requirements for enabling and controlling actions before being allowed access to the environment infrastructures - the so called (coordination) artifacts, to fulfill their intended role(s). Note that, in MAC, information flow among levels is necessary acyclic - one direction information flow. This policy indicates, in the context of RBAC/MAS, which agent has access to which resource to play its role. This kind of access control model can increase the level of security policy defined by the organization, because it is based on a policy that does not allow any operation not explicitly authorized by the policy maker - could be another autonomous agent. Moreover, subjects (could be agents or roles) are given clearances while resources (artifacts or objects) are dynamically assigned sensitivity labels at ACC negotiation.

The ACCs must facilitate to emulate mandatory access control. To this end, we must ensure that the organization mandatory access control function relies solely on the fact that agent has authorization to a role, that the role contains an associated organization policy (based on role policy description) specifying the laws that rule agent access to resources according to their roles, that the agent access is via this legal rule, and that the access does not violate the specified flow policy. The description of role policies defining the patterns of actions and interactions permitted to a role can be specified using logic programming (such as Prolog) in the extended version of TuCSoN for RBAC [4]. Consequently, legal access of the organization resource should be enforced in order to ensure that the resource is accessed in no laws other than those specified in the role as well as any other constraints specified on such access. Furthermore, a flow policy must be observed since it is the criterion that determines security concerns.

3.4. MAC constraints in ACC

As lesson leaned from [1], i.e., in role-based systems information can be partitioned via roles, the organization infrastructures (coordination artifacts or resources or information) can be structured or partitioned and availed them via the policy defined around the roles. Consequently, the resources available via one such policy compose a context related to the role. Each of these contexts is available to agents via the two stages activities provided by ACC. The term context is used here to distinguish the system-defined security levels from that of policy defined around the role. To be more precise a role's resource context is that of organizational infrastructure (such as information, or resources, or artifacts) solely obtainable via the role following successful negotiation with ACC.

The notion of ACC is exploited to embody an agent working session within the TuCSoN organization, enforcing the forms of access control specified by the policies associated with agent roles(s) [4]. In particular, this must be carried out during ACC negotiation governed by agent-role and inter-role relationships. Accordingly, this facilitates access to objects or resources via the role thus we can use this fact to specify MAC constraint. To this end, we employ two key concepts on agents and organization infrastructures: agent authorization to a role and resource accessibility via a role.

Moreover, we can enforce mandatory authorization constraint. In a sense that agent can solely access a resource via an authorized role in the sort defined for the associated policy in the role. The role authorization can be specified in the form of role's access list which makes easier the fortitude of agent's authorization to a given role. Agent authorization to a role in ACC confers the agent can access or enable resources available via the role via legal role policy description modes. The authorization scheme must ensure that it does not violate the specified organization security policy.

To strengthen the mandatory authorization constrains motioned above, we must induce more constraints following the four mandatory rules discussed in Section 2.1. To do so, here we consider Liberal *-Property rule. Subjects with security labels higher have more authority with respect to read operations

but less power with respect to write operations. An alternative approach for representing the security labels is by using lattice and computation of the clearance and classification levels facilitated by defining appropriate partial order function (see for example in [11]). The role hierarchy, as mentioned before, is a useful metaphor for our formalization to configure the mandatory rules. In role hierarchies subjects (sessions) with roles higher in the hierarchy always have more authority than those with roles lower in the hierarchy.

Accordingly, we need to exploit appropriate constraints to reflect the labels on subjects in MAC to complete our formalization. Users in MAC have a sole security clearance. This is enforced by requiring that each agent in RBAC/MAS is associated with a number of roles that the agent context possibly after successfully ACC negotiation. Moreover, a MAC user can login at any label dominated by the user's clearance. In RBAC/MAS, this requirement can possibly be captured by requiring that each agent control room is associated with exactly two matching roles. Intuitively, MAC is enforced in terms of read and write operations. Consequently, in RBAC/MAC, this reflects the fact that the policies specified to the agents are that of read and write on the individual artifact of the organization. In this way, it's possible to configure the other rules given in Section 2.1 in RBAC/MAS architecture.

3.5. Example

For instance, consider electronic voting system (e-voting) where a number of agent (possibly autonomous) context enforcing a serious of organization laws or policies (in this case the organization is the government - for example Provincia Autonoma di Trento). There are number of agents (voter, election officials, poll place worker, election president, etc) each of them having different clearance labels varying according to the legal laws specified by the government body, each agent assigned to different role with the minimum privilege assigned to each role expected to be played by the associated agent at design time. Voter (one of the agents) has the right to vote after his/her right to vote is authorized against the national or regional laws. Once the voter authenticates him/herself, a working voting session containing a blank ballot box is provided. The voter then requests an ACC (the voting terminal) specifying which roles to activate. If the voter request is compatible with organization rules (election laws), a new ACC is created, and released to the voter for actively playing inside the organization - the actual casting process. The voter then can use the ACC to interact with the organization environment, by exploiting the actions/perceptions enabled by the ACC, for instance, asking the printer (one of the possible organization resource around which MAC policy specified and enforced during runtime) to print and show to the voter what he/she actually voted.

4. Related Work

Michael and Ahn in [12] proposed integration of a role-based access control into unified modeling language (UML) aimed to close the gap between security model and system development by viewing RBAC from static view, functional view and dynamic views. A similar work has also been proposed in [9] meant to integrate MAC model into UML, specifically with use case, class, and sequence diagrams to support security assurance checking. Moreover, the latter approach focused on defining the clearances and classifications to support relevant UML elements. By its very nature, differently from these works, our applicative domain is agent-oriented paradigm.

An approach has been proposed to configure RBAC to enforce MAC and discretionary access control (DAC) policies in [8]. Here, the main focuses were to simulate and enforce the common forms of lattice based access control (LBAC or MAC) and DAC model upon the well known RBAC (sometimes referred as RBAC96) model. The approach essentially considered users and permissions to articulate any access control model. LBAC was simulated by role hierarchy whereas DAC policies were enforced by considering administrative role hierarchy. Differently from this work, our approach specifically proposes an integration of MAC in organization built upon role-based multi-agent systems.

A work that deals with integration of MAC in RBAC model - therefore being particularly close in topic to ours - is the approach by Matunda and Sylvia [1]. This approach suggests modeling and realization of mandatory access control into role-based protection system. However, our approach a bit advanced. In a sense that

our approach specifically proposes an integration of MAC on top of organization built upon RBAC/MAS reference model.

5. Conclusion and Future Work

This paper proposed an integration of a mandatory access control on RBAC-like MAS to strengthen the security over organization infrastructures. The most important feature of mandatory access controls involve limiting agent's full control over the access to environment infrastructures (such as resources or artifacts) that meant to be shared with other agents. Therefore the integration of MAC in RBAC/MAC can be seen as dual effect. On one hand, we can reexamine the way to enable and control actions of the agents toward the organizational infrastructure by realizing the mandatory access control policies. The role hierarchy, on the other hand, can be used as metaphor to control the follow of information across the organization systems respecting the laws specified around the roles.

Even though the approach is quite interesting and unique in its kind, it absolutely lacks formalization and practicality - working example. Sometime discussions have been made in very abstract sense hence making it difficult to understand.

However, the work can be easily refined and extended in many directions. For instance, revision of the current RBAC-like MAS environment to include MAC model. Devising a methodology for organization infrastructure partitioning based on the role hierarchy metaphor to which the agent requests the ACC would also be interesting. Appling the approach to working domain, for instance, e-voting would be a nice organization system where many coordinable artifacts with different security levels utterly need mandatory access control policies.

References

[1] Matunda Nyanchama and Sylvia L. Osborn. Modeling Mandatory Access Control in Role-Based Security Systems. In *IFIP Workshop on Database Security*, pages 129–144, 1995. Available at `http://citeseer.ist.psu.edu/nyanchama95modeling.html`.

[2] Sylvia Osborn. Mandatory Access Control and Role-Based Access Control Revisited. In *RBAC '97: Proceedings of the second ACM workshop on Role-based access control*, pages 31–40, New York, NY, USA, 1997. ACM Press.

[3] Ravi S. Sandhu and Edward J. Coyne and Hal L. Feinstein and Charles E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996. Available at `http://citeseer.ist.psu.edu/sandhu96rolebased.html`.

[4] Andrea Omicini and Alessandro Ricci and Mirko Viroli. RBAC for Organisation and Security in an Agent Coordination Infrastructure. *Electr. Notes Theor. Comput. Sci.*, 128(5):65–85, 2005. `http://dx.doi.org/10.1016/j.entcs.2004.11.045` and DBLP `http://dblp.uni-trier.de`.

[5] Mirko Viroli, Alessandro Ricci, and Andrea Omicini. An organisation Infrastructure for Multi-Agent Systems based on Agent Coordination Contexts. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 1189–1190, New York, NY, USA, 2005. ACM Press.

[6] Alessandor Ricci and Mirko Viroli and Andrea Omicini. Agent Coordination Context: From Theory to Practice. In *In proceedings of the 17th European Meeting on Cybernetics and System Research (EMCSR 2004), Vienna, Austria*, volume 2, pages 618–623, 2004.

[7] Charles E. Phillips Jr., Steven A. Demurjian, and T. C. Ting. Security Assurance for an RBAC/MAC Security Model. In *IAW*, pages 260–267, 2003. DBLP, `http://dblp.uni-trier.de`.

[8] Sylvia Osborn, Ravi Sandhu, and Qamar Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Trans. Inf. Syst. Secur.*, 3(2):85–106, 2000. `http://doi.acm.org/10.1145/354876.354878`.

[9] Thuong Doan and Steven Demurjian and T. C. Ting and Andreas Ketterl. MAC and UML for Secure Software design. In *FMSE '04: Proceedings of the 2004 ACM workshop on Formal methods in security engineering*, pages 75–85, New York, NY, USA, 2004. ACM Press. http://doi.acm.org/10.1145/1029133.1029144.

[10] Yixin Jiang and Chuang Lin and Hao Yin and Zhangxi Tan. Security Analysis of Maindatory Access Control Model. *2004 IEEE International Conference on Systems, Man and Cybernetics*, 6:5013– 5018, 2004.

[11] Ravi S. Sandhu. Lattice-Based Access Control Models. *IEEE Computer*, 26(11):9–19, 1993. citeseer.ist.psu.edu/article/sandhu93latticebased.html.

[12] Michael E. Shin and Gail-Joon Ahn. UML-Based Representation of Role-Based Access Control. In *WETICE '00: Proceedings of the 9th IEEE International Workshops on Enabling Technologies*, pages 195–200, Washington, DC, USA, 2000. IEEE Computer Society.