MRBAC: Hierarchical Role Management and Security Access Control for Distributed Multimedia Systems

Na Zhao[1], Min Chen[2], Shu-Ching Chen[1], Mei-Ling Shyu[3]
[1]*Distributed Multimedia Information System Laboratory*
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
[2]*Department of Computer Science*
University of Montana, Missoula, MT 59812, USA
[3]*Department of Electrical & Computer Engineering*
University of Miami, Coral Gables, FL 33124, USA
[1]*{nzhao002, chens}@cs.fiu.edu,* [2]*min.chen@mso.umt.edu,* [3]*shyu@miami.edu*

Abstract

In this paper, a Role-based Access Control (RBAC) model is applied and extended to a multimedia version called Multi-Role Based Access Control (MRBAC), which can fully support the comprehensive and multi-level security control requirements of the distributed multimedia applications. The object-oriented concept is adopted in MRBAC to perform the hybrid role hierarchy management and security roles and rules administration. In summary, MRBAC can: 1) support the multi-level security protection for multimedia data; 2) provide access control by checking both the time constrains and IP addresses; and 3) decentralize the administration functions to make the access control management more efficient.

1. Introduction

With the proliferation of multimedia data and applications, security assurance of the multimedia database systems becomes a critical issue. For instance, many application domains such as medical and military may contain sensitive information which should not be or could only partially be accessed by general users. Therefore, it is essential to support security management of multimedia systems and design security models accordingly.

Many research studies have been conducted to address the system security issues using approaches like role management and access control. For instance, Role-Based Access Control (RBAC) has been widely used for user authorization control [9] in database management, workflow management [2], web environments [1], etc. However, most of the existing RBAC models mainly focus on document protection without considering spatial and temporal characteristics of multimedia data. In another words, the control is posed on the file-level without taking care of the internal multimedia contents. However, in real multimedia applications, it is quite common to block certain visual/audio objects or segments in a multimedia file. Some typical examples would be to prevent victim's face (i.e., object) from showing in the news broadcast or to skip violent scenes (i.e., segment) for young audience. Therefore, it is essential to provide a model which can fully support the comprehensive and multi-level security control requirements in distributed multimedia applications.

To address this need, in this paper, a Multi-Role Based Access Control (MRBAC) model is proposed, which is extended from the traditional RBAC model. In this model, the object-oriented concept is adopted to perform the hybrid role hierarchy management and security roles/rules administration.

This paper is organized as below. Related work is presented in Section 2. Section 3 introduces the background of security management requirements of distributed multimedia systems, and our proposed security framework. In Section 4, MRBAC is proposed by extending the traditional RBAC models. This proposed security model is evaluated in Section 5 by using two example scenarios. Finally, the conclusions are summarized in Section 6.

2. Related Work

Role-Based Access Control (RBAC) is a security solution to restricting system access to authorized users. The fundamental feature of RBAC is to support the administration of large numbers of privileges on system objects, and reduce the effort to define and manage complex security policies. With RBAC, roles are created for various characters based on their job functions. For a specific role, the permissions to perform certain operations are assigned to it and the member of this role can acquire these permissions to perform the particular system operations. Since the permissions are not assigned to users directly, the management of individual users becomes easier. It is simply a matter of assigning appropriate roles to the users.

Sandhu et al. [13] summarized and categorized the traditional RBAC models into four families: RBAC0 – base model; RBAC1 – hierarchical model; RBAC2 – constraint model; and RBAC3 – combined model. Traditional RBAC models have many restrictions on the access control modeling. By evaluating the traditional RBAC approaches, it has been found that several issues still remain open. First, temporal constraints may not be considered when setting the roles. Second, the locations of users are not restricted. Third, most security applications can only handle access control on multimedia files without taking care of multimedia contents. Fourth, it lacks a hierarchical architecture for the roles and therefore the role management will become complicated when the number of users increases manifold. Therefore, numerous extended RBAC models have emerged to handle those unresolved security issues.

In [3], the Temporal Role-Based Access Control (TRBAC) model, which brings the basic temporal dependencies, was proposed. However, it still could not handle several useful temporal variables including the constraints on user-role and role-permission assignments. The Generalized Temporal Role-Based Access Control (GTRBAC) model [8] was proposed later to solve this problem. Recently, this model was extended to an XML-based version called X-GTRBAC [4], which incorporates the content- and context-aware dynamic access control requirements of an enterprise. However, these models only improved the control capability on temporal constraints. Moyer et al. proposed the Generalized Role-Based Access Control (GRBAC) model which leverages the traditional RBAC by incorporating subject roles, object roles, and environment roles [11]. However, they only introduced the temporal constraints in the environment roles, and it could only handle access control on multimedia files without taking care of multimedia contents. Another Generalized Object-Composition Petri-Net Model (GOCPN) was proposed in [7], which mainly focuses on the modeling of documents to allow secure accesses to a multimedia database management system. GOCPN utilizes a mandatory access control (MAC) approach which cannot fully perform complicated roles, role hierarchies, temporal constraints, and IP address restrictions. All these motivate us to develop the Multi-Role Based Access Control (MRBAC) model that not only enables sophisticated management on roles and their hierarchies, but also supports temporal constraints and IP address restrictions.

3. Overview of SMARXO

In our previous research [5], we have proposed a security framework called SMARXO. The SMARXO architecture consists of three phases to achieve security verification and access control for multimedia applications.

- Multimedia indexing phase

In order to describe the hierarchical structure of multimedia data and to perform multi-level security control, the multimedia source data need to be pre-processed and segmented. The recent advances in multimedia data analysis techniques have enabled image object identification, video shot segmentation, etc. The automatically generated results can be used for indexing. In addition, the users are also allowed to manually identify their interested visual objects or video segments. Such information is indexed using XML languages. Due to the advantages of XML, the tags can be defined flexibly depending on the detailed requirements, and thus the users can define a particular type of visual or audio object in the multimedia repository in their own preferences. Here, the formal definitions of an image object, a video shot, and a salient moving video object are presented.

Definition 1: An image object is defined as IO = <*io_id, io_name, i_id, io_x, io_y, io_width, io_height*>, which means the object ID, object description, the ID of the image containing this object, x and y coordinates of the top-left point, and the object width and height, respectively. Here a rectangular bounding box is adopted to represent an image object. In case of the object is not permitted for displaying, the system would use the information of this bounding box to process the image source data such that only the other surrounding area of this image is displayed.

Definition 2: A video shot is defined as VS = <*vs_id*, *vs_name*, *v_id*, *vs_startframe*, *vs_framenum*>, where it identifies the ID of the video shot, its description, the ID of the video containing this video shot, starting frame number, and the number of total frames of this shot.

Definition 3: A salient moving video object can be defined as VO = <*vo_id*, *vo_name*, *v_id*, *vo_startframe*, *vo_framenum*, *vo_positions*>, where it includes the ID and description of this video object, the ID of the video containing this object, starting frame number, the number of total frames where this video object shows, and the position information of this object in the consecutive frames. That is, *vo_positions* is defined by a sequence of detailed position information *vo_position* = <*frame_id*, *vo_x*, *vo_y*, *vo_width*, *vo_height*> of this certain object in the consecutive frames represented by their corresponding moving bounding boxes.

- Security modeling phase

In this phase, we extended the traditional RBAC to the multimedia version of RBAC, which can also be represented as Multi-Role Based Access Control (MRBAC). Four kinds of roles are defined to handle a request behavior in the multimedia applications.

1) User roles (also called subject roles) are defined to recognize the users' responsibility and privilege in the system. As the access control permissions are granted to the roles, the users with the same role are permitted to perform the same set of operations.
2) Object roles are utilized to describe the groups of multimedia objects or segments such that the access control can further reach multiple hierarchical levels of multimedia data.
3) Temporal roles are defined as the groups of time period. They are responsible of controlling the effective time of the access functionalities.
4) Spatial roles are described using the sets of IP addresses. They are designed to restrict unauthorized accesses from alien computers by checking their IP addresses.

In the traditional access control models [12], the "privilege" is normally defined as a pair as (*x*, *m*), where *x* is an object and *m* refers to the accessing model, e.g. read-permission, write-permission, etc. RBAC model makes it a three tuple (*Ru*, *x*, *m*), where *Ru* represents a user role. In this proposed research, the access control rule is defined as follows.

Definition 4: In MRBAC, the security access control rule is defined as a 5 tuple *p* = (*Ru*, *Ro*, <*Rt*>, <*Rs*>, *M*), where *Ru*, *Ro*, *Rt*, *Rs* denote user role, object role, temporal role, and spatial role, respectively. *M* denotes the access mode (or permission) for the objects represented by *Ro*.

Please note the following updates are made in this concept:

1) The object role *Ro* is used to replace the traditional object. Therefore, we can define security permission for a set of multimedia objects or other type of files.
2) The temporal role *Rt* and the IP address-based spatial role *Rs* are introduced such that the temporal constrains and access computers can be controlled. However, these two items can be omitted when defining an access control policy. The default situation is that the rule can be applied all the time by using any computer.
3) The definition of access control mode *M* is also extended to handle multimedia data. The extended access modes include: reading-allowed, reading-denied, reading-partially-allowed, writing-allowed, writing-denied, writing-partially-allowed, etc.

By combining all these four roles, the security access policies are defined as access control rules. All the security roles and rules can be stored in XML documents for efficient retrieval and display purpose. Please refer to [5] for further details.

- DBMS management phase

The pre-processed multimedia contents, extracted multimedia features, and XML-based multimedia indexing documents as well as security access roles and rules are stored in an object-relational database. Combining the advantages of ORDBMS and XML, this proposed SMARXO framework is manageable, efficient, scalable, adaptable, and can provide a more secure environment for the distributed multimedia systems.

4. Hybrid Role Hierarchy in MRBAC

In the RBAC systems, a user role can also be defined as the set of rights and duties which are assigned to the person who occupies that role. As aforementioned, the "role" concept is further generalized in our proposed security access control framework to describe the groups of accessing objects, time periods, as well as IP address groups. Each category of roles can be modeled and managed using a hierarchical manner, which will be further defined and formalized in this section.

Role hierarchy is a very important concept in RBAC as the systems need to make access permission decisions based on the position of a role in the whole hierarchy. Moffet et al. [10] summarized three kinds of hierarchies in an organization or a system:

- Generalization hierarchy: The "isa" relationship is incorporated in the generalization hierarchy. The generalization hierarchy means that each of these roles is more general than the previous one, and they constitute a partial order. The backward order of this hierarchy can actually represented as the inheritance relationship in the object-oriented concept.

- Aggregation hierarchy: Complex objects can be composed of aggregated parts. Therefore, this relationship can also be described as the "part of" relationship. Here, the aggregation hierarchy (also called the activity hierarchy) is partially ordered by subsets of activities. In the RBAC systems, it is highly possible that we define the role hierarchy based upon the activities that the roles are responsible for. In this aggregation hierarchy, the higher role is responsible for a super set of the activities of the lower level roles. The activity is either performed by the role directly, or it will be delegated to another lower level role.
- Supervision hierarchy: In a supervision role hierarchy, each node describes a named position, which may contain one or more roles.

Depending on the role hierarchies discussed above, especially the generalization hierarchy and aggregation hierarchy, we can further explore the hybrid role hierarchy management and cross-category role management in MRBAC.

In the Generalized Temporal RBAC (GTRBAC) model [9], the hybrid hierarchy was introduced to facilitate specifications of fine grained RBAC policies [6]. In a hybrid hierarchy, the following three hierarchical relations among roles can co-exist and their semantic meanings are explained in Table 1.

- *I*-hierarchy ($\geq i$): *Permission-inheritance-only*.
- *A*-hierarchy ($\geq a$): *Role-activation-only*.
- *IA*-hierarchy ($\geq$): Both *permission-inheritance* and *role-activation*.

Table 1. Semantic meaning of hybrid user role hierarchies

Symbol	Descriptions
$Ux \geq i\ Uy$	Permissions available through *Uy* are also available through *Ux*.
$Ux \geq a\ Uy$	Any user who can activate *Ux* can also activate *Uy*.
$Ux \geq Uy$	*Ux* inherits permissions of *Uy* and the users that can activate *Ux* can also activate *Uy*.

These three hybrid role hierarchy definitions can be applied to manage the user roles in our proposed SMARXO framework. The administration privilege is partially decentralized and delegated to the lower hierarchy roles. However, this policy needs improvements to satisfy the requirements in a complicated multimedia system. There are multiple issues need to be addressed:

1) This policy cannot be fully applied to other types of roles in MRBAC. For the object roles, temporal roles, and spatial roles, the role hierarchy is rather simple without the requirements of administration decentralization. Assume there is a temporal role hierarchy as Holiday→Thanksgiving. If the user does not have accessibility in Holiday then he/she cannot access the system on Thanksgiving. That is, the lower-hierarchy role normally belongs to or is a subset of the higher hierarchy roles (similar to the aggregation hierarchy concept).
2) The cross manipulation relationships between user roles and the other types of roles are not modeled.
3) The partially accessibility of multimedia data is not covered. Take an image as an example. If a permission rule is granted to user as partially readable, the whole image can be shown to the user except one or more image objects in this image.

Table 2. Semantic meaning of temporal and spatial role hierarchies

Symbol	Descriptions
$Tx \geq t\ Ty$	Access rules available through temporal role *Tx* are also available through temporal role *Ty*.
$Sx \geq s\ Sy$	Access rules available through spatial role *Sx* are also available through spatial role *Sy*.

Therefore, in this research, we expand the role hierarchy concept for other categories of roles. Basically, the temporal role and spatial role hierarchies are inherited and therefore simple to represent. The denotations and their meanings are shown in Table 2.

The situation becomes more complicated when dealing with multimedia object roles. As we discussed, a "partially-accessible" permission mode is designed particularly for multimedia data. For this kind of permission mode, the security checks are performed on the "Object Entity Set" (OES) of the request object *o* that includes both the object itself and all the entities *s* (segments or sub-objects) belonging to *o*.

Definition 5: Object Entity Set: $OES(o) = \{o\} \cup \{s : s \in o\}$. The object role hierarchies are defined in Table 3.

Assume the function "*p_check*(*o*)" can check if the user can access object *o* in the specified time from some specified computer. The partially accessible condition can be represented as below:

$$(p\_check(o) = TRUE) \wedge (\exists s \in o[p\_check(s) = FALSE])$$

where *s* can be any sub-object or segment which belongs to *o*. In this situation, the user can access the processed multimedia data *o** where the prohibited sub-objects are removed from *o*.

Given the role hierarchies defined above, Table 4 shows the function representations for various conditions for user permission control, where *u* is a user, *Ru* is a user role, *Rt* is a temporal role, *Rs* is a spatial role, *Ro* is an object role, and *p* is a permission. Each of these functions is defined using the overloading functionality in the object-oriented concept. Different parameters can be considered upon different environmental conditions. Taken the most comprehensive situation as examples, the following implications hold:

- *permission_assigned*(*p*, *Ru*, *Ro*, *Rt*, *Rs*)
→*can_be_acquired*(*p*, *Ru*, *Ro*, *Rt, Rs*)
- *user_assigned*(*u*, *Ru*, *Rt*, *Rs*)
→*can_activate_role*(*u*, *Ru*, *Rt*, *Rs*)
- *can_activate_role*(*u*, *Ru*, *Rt*, *Rs*)
∧ *can_be_acquired*(*p*, *Ru*, *Ro*, *Rt, Rs*)
→*can_acquire_permission*(*u*, *p*, *Ro*, *Rt*, *Rs*)

Table 3. Semantic meanings of object role hierarchies.

Symbol	Descriptions
Ox ≥*o* *Oy*	Permission is fully accessible or fully deny: Access rules available through object role *Ox* are also available through object role *Oy*.
Ox ≥*o'* *Oy*	Permission is partially accessible: For the object set *o'* which is not accessible in *Ox*: If $o' \cap Oy = \phi$, *Oy* is accessible; If $o' \cap Oy = Oy$, *Oy* is not accessible; If $o' \cap Oy \neq \phi$ and $o' \cap Oy \neq Oy$, *Oy* is partially accessible.

Table 4. Status related to role activation, permission assignment, and acquirement.

Notations	Descriptions
role_enabled(*Ru*)	*Ru* is enabled
role_enabled(*Ru*, *Rt*)	*Ru* is enabled at time period in *Rt*
role_enabled(*Ru*, *Rs*)	*Ru* is enabled by using the computers included in *Rs*
role_enabled(*Ru*, *Rt*, *Rs*)	*Ru* is enabled at time period in *Rt* by using the computers included in *Rs*
user_assigned(*u*, *Ru*)	*u* is assigned to *Ru*
user_assigned(*u*, *Ru*, *Rt*)	*u* is assigned to *Ru* at time in *Rt*
user_assigned(*u*, *Ru*, *Rs*)	*u* is assigned to *Ru* by using the computers included in *Rs*
user_assigned(*u*, *Ru*, *Rt*, *Rs*)	*u* is assigned to *Ru* at time in *Rt* by using the computers included in *Rs*
permission_assigned(*p*, *Ru, Ro*)	*p* is assigned to *Ru* on *Ro*
permission_assigned(*p*, *Ru*, *Ro*, *Rt*)	*p* is assigned to *Ru* at time in *Rt* on *Ro*
permission_assigned(*p*, *Ru*, *Ro*, *Rs*)	*p* is assigned to *Ru* by using the computers included in *Rs* on *Ro*
permission_assigned(*p*, *Ru*, *Ro*, *Rt*, *Rs*)	*p* is assigned to *Ru* at time in *Rt* by using the computers included in *Rs* on *Ro*
can_activate_role(*u*, *Ru*)	*u* can activate *Ru*
can_activate_role(*u*, *Ru*, *Rt*)	*u* can activate *Ru* at time in *Rt*
can_activate_role(*u*, *Ru*, *Rs*)	*u* can activate *Ru* by using the computers included in *Rs*
can_activate_role(*u*, *Ru*, *Rt*, *Rs*)	*u* can activate *Ru* at time in *Rt* by using the computers included in *Rs*
can_acquire_permission(*u*, *p*, *Ro*)	*u* can acquire *p* on *Ro*
can_acquire_permission(*u*, *p*, *Ro*, *Rt*)	*u* can acquire *p* at time in *Rt* on *Ro*
can_acquire_permission(*u*, *p*, *Ro*, *Rs*)	*u* can acquire *p* by using the computers included in *Rs* on *Ro*
can_acquire_permission(*u*, *p*, *Ro*, *Rt*, *Rs*)	*u* can acquire *p* at time in *Rt* by using the computers included in *Rs* on *Ro*
can_be_acquired(*p*, *Ru*, *Ro*)	*p* on *Ro* can be acquired through *Ru*
can_be_acquired(*p*, *Ru*, *Ro*, *Rt*)	*p* on *Ro* can be acquired through *Ru* at time in *Rt*
can_be_acquired(*p*, *Ru*, *Ro*, *Rs*)	*p* on *Ro* can be acquired through *Ru* by using the computers included in *Rs*
can_be_acquired(*p*, *Ru*, *Ro*, *Rt, Rs*)	*p* on *Ro* can be acquired through *Ru* at time in *Rt* by using the computers included in *Rs*

In general, the role hierarchies allow the administrator to change the hierarchical relationships between a set of roles. The administration privilege can also be decentralized and delegated to lower hierarchy roles. As shown in Definitions 6 to 10, the proposed MRBAC role hierarchy theory also considers the temporal roles, spatial roles, and object roles.

Definition 6: I-Hierarchy for the user roles in MRBAC. Let *Ux* and *Uy* be user roles such that ($Ux \geq i\ Uy$), that is, *Ux* has a permission inheritance-only relation over *Uy* on object *Ro* at time *Rt* using the computer represented by spatial role *Rs*. Then the following implication holds:

$$\forall p, (Ux \geq i\ Uy) \wedge can\_be\_acquired(p, Uy, Ro, Rt, Rs)$$
$$\rightarrow can\_be\_acquired(p, Ux, Ro, Rt, Rs)$$

Definition 7: A-Hierarchy for the user roles in MRBAC. Let *Ux* and *Uy* be user roles such that ($Ux \geq a\ Uy$), that is, *Ux* has a role activation-only relation over *Uy* at time *Rt* using the computer represented by spatial role *Rs*. Then the following implication holds:

$$\forall p, (Ux \geq a\ Uy) \wedge can\_activate\_role(u, Uy, Rt, Rs)$$
$$\rightarrow can\_activate\_role(u, Ux, Rt, Rs)$$

Definition 8: IA-Hierarchy for the user roles in MRBAC. Let *Ux* and *Uy* be user roles such that ($Ux \geq Uy$), that is, *Ux* has a general inheritance relation over *Uy* at time *Rt* using the computer represented by spatial role *Rs*. Then the following implication holds:

$$\forall p, \forall u, ((Ux \geq Uy)$$
$$\wedge\ can\_be\_acquired(p, Uy, Ro, Rt, Rs)$$
$$\wedge\ can\_activate\_role(u, Ux, Rt, Rs))$$
$$\rightarrow \quad (can\_be\_acquired(p, Ux, Ro, Rt, Rs)$$
$$\wedge\ can\_activate\_role(u, Uy, Rt, Rs))$$

Definition 9: Temporal role hierarchy in MRBAC. Let *Tx* and *Ty* be temporal roles such that ($Tx \geq t\ Ty$), that is, *Tx* has an inheritance relation over *Ty*. Then the following implication holds:

$$\forall p, (Tx \geq t\ Ty) \wedge can\_be\_acquired(p, Ru, Ro, Ty, Rs)$$
$$\rightarrow can\_be\_acquired(p, Ru, Ro, Tx, Rs)$$

Definition 10: Spatial role hierarchy in MRBAC. Let *Sx* and *Sy* be spatial roles such that ($Sx \geq s\ Sy$), that is, *Sx* has an inheritance relation over *Sy*. Then the following implication holds:

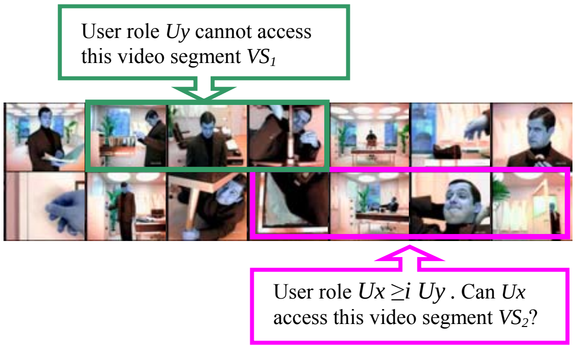
$$\forall p, (Sx \geq s\ Sy) \wedge can\_be\_acquired(p, Ru, Ro, Rt, Sy)$$
$$\rightarrow can\_be\_acquired(p, Ru, Ro, Rt, Sx)$$

5. Security Model Evaluation

In this section, two examples in a distributed multimedia system are presented to show that our proposed MRBAC model is more practical and effective in security modeling of the complicated scenarios especially in multimedia applications. It is noticed that the other RBAC models cannot function well in these situations.

Scenario 1**:** In a distributed multimedia system in a hospital, there is a security rule shows that the certain medical images are only accessible by user role of Doctor through the computers inside the hospital local network. For example, the IP address role contains "131.94.*.*". Therefore, is it possible to access these images using a doctor's home computer with IP address "131.95.12.32"?

The traditional RBAC models do not check the accessing computers. However, it is a useful function to confirm the secure access point by using IP addresses, which is done in our proposed MRBAC model. In addition, our model also can support temporal constraint checking.


User role *Uy* cannot access this video segment VS_1

User role $Ux \geq i\ Uy$. Can *Ux* access this video segment VS_2?

Figure 1. Access control example on video segments

Scenario 2**:** In a distributed multimedia system, the video file *V* can be further segmented into 14 video shots (as shown in Figure 1). User role *Uy* can access this video except the video set VS_1. There is another user role *Ux* and another video segment VS_2. Given $Ux \geq i\ Uy$, what is the accessibility of *Ux* on VS_2?

In this scenario, the traditional RBAC model can hardly describe this situation because: 1) there is no multi-level access control mechanism defined to process multimedia data; and 2) the object role hierarchy is not defined in the traditional hybrid role hierarchy. On the other hand, in our proposed MRBAC model, the scenario can be defined as follows:

Assume for user role *Uy*, $V \geq o'\ VS_1$, which means the permission mode for *Uy* on *V* is partially accessible as the object set VS_1 is not accessible by *Uy*. For $V \geq o'\ VS_2$, since $VS_1 \cap VS_2 = \phi$, VS_2 is accessible by *Uy*. Given $Ux \geq i\ Uy$, VS_2 is therefore accessible by *Ux*.

In general, these two scenarios show the advantages of our proposed MRBAC model, along with the extended hybrid role hierarchy theory. Furthermore, MRBAC can actually handle more complicated security requirements in a distributed multimedia system, for example, the access control on a moving video object.

6. Conclusions

In this paper, the unique security control requirements caused by the spatial and temporal characteristics of multimedia data in distributed multimedia systems are discussed. Accordingly, a Multi-Role Based Access Control (MRBAC), extended from tradition RBAC, is proposed. This model performs hybrid role hierarchy management and security roles/rules administration. Therefore, it can support multi-level security protection for multimedia data and provide access control by checking both the time constrains and the IP addresses. In addition, it is efficient as the administration functions are decentralized. Two example scenarios are used to evaluate our proposed MRBAC model and to demonstrate its effectiveness and practicality in security modeling for multimedia systems.

7. Acknowledgements

Shu-Ching Chen's research was supported in part by NSF HRD-0317692 and Florida Hurricane Alliance Research Program sponsored by the National Oceanic and Atmospheric Administration. Na Zhao's research was supported in part by Florida International University Dissertation Year Fellowship.

8. References

[1] J. Barkley, A. Cincotta, D. Ferraiolo, S. Gavrula, and D. R. Kuhn, "Role Based Access Control for the World Wide Web," In *Proceedings of 20th National Information System Security Conference*, NIST/NSA, 1997, pp. 331-340.

[2] E. Bertino and E. Ferrari, "The Specification and Enforcement of Authorization Constraints in Workflow Management Systems," *ACM Transactions on Information and System Security*, Feb. 1999, Vol. 2, No. 1, pp. 65–104.

[3] E. Bertino, P. A. Bonatti, and E. Ferrari, "TRBAC: A Temporal Role-Based Access Control Model," *ACM Transactions on Information and System Security (TISSEC)*, Vol. 4, No. 3, August 2001, pp. 191–233.

[4] R. Bhatti, J. B. D. Joshi, E. Bertino, and A. Ghafoor, "X-GTRBAC: An XML-based Policy Specification Framework and Architecture for Enterprise-Wide Access Control," *ACM Transactions on Information and System Security*, 2005, Vol. 8, No. 2, pp. 187–227.

[5] S.-C. Chen, M.-L. Shyu, and N. Zhao, "SMARXO: Towards Secured Multimedia Applications by Adopting RBAC, XML and Object-Relational Database," In *Proceeding of the 12th Annual ACM International Conference on Multimedia (ACM-MM),* October 10-16, 2004, New York, USA, pp. 432–435.

[6] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-based Access Control," *ACM Transactions on Information and Systems Security*, August 2001, Vol. 4, No. 3, pp. 224–274.

[7] J. B. D. Joshi, K. Li, H. Fahmi, B. Shafiq, and A. Ghafoor, "A Model for Secure Multimedia Document Database System in a Distributed Environment," *IEEE Transactions on Multimedia,* June 2002, Vol. 4, No. 2, pp. 215–234.

[8] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "Generalized Temporal Role Based Access Control Model," *IEEE Transactions on Knowledge and Data Engineering*, Jan. 2005, Vol. 7, Issue 1, pp. 4–23.

[9] J. B. D. Joshi, E. Bertino, and A. Ghafoor, "Formal Foundations for Hybrid Role Hierarchy," *ACM Transactions in Information and Systems Security*, in print for Nov. 2007.

[10] J. D. Moffett and E. C. Lupu, "The Uses of Role Hierarchies in Access Control", In *Proceedings of the Fourth ACM workshop on Role-Based Access Control*, 1999, pp. 153–160.

[11] M. J. Moyer and M. Ahamad, "Generalized Role-Based Access Control," In *Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS 2001)*, April 2001, pp. 391–398.

[12] M. Nyanchama and S. Osborn, "The Role Graph Model and Conflict of Interest," *ACM Transactions on Information and System Security*, 1999, Vol. 2, No. 1, pp. 3–33.

[13] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models", *IEEE Computer*, IEEE Press, 1996, Vol. 29, No. 2, pp. 38–47.