5

EMEDAC: ROLE-BASED ACCESS CONTROL SUPPORTING DISCRETIONARY AND MANDATORY FEATURES

Ioannis Mavridis, George Pangalos and Marie Khair

Abstract In this paper, we present an enhanced use of RBAC features in articulating a security policy for access control in medical database systems. The main advantage of this implementation is that it supports both MAC and DAC features at the same time; a feature that has been proved to be necessary in healthcare environments. The eMEDAC security policy that results from the above implementation provides an enhanced redefinition of a number of mechanisms of the already known MEDAC security policy. The concept of hyper node hierarchies is proposed for deriving totally ordered security levels while preserving the role hierarchy levels required satisfying particular administration needs. Finally, a demonstration example is given based on the pilot implementation of the proposed security policy in a major Greek hospital. The advantages offered are related to the efficiency of access control, the flexibility and decentralisation of administration, and the storage savings.

Keywords: EMEDAC, role-based access control, security policy

1. INTRODUCTION

Until now, in the key area of access control three major categories of security policies have been proposed that are usually used in computer systems: the discretionary policies, the mandatory policies, and the role-based policies. A recent well known role-based approach is RBAC (Role Based Access Control), which has received considerable attention as a promising way to enhance traditional discretionary (DAC) and mandatory (MAC) access controls. According to [16], an important characteristic of RBAC is that by itself it is policy neutral. Furthermore, the security policy enforced in a particular system could be the net result of the appropriate configuration and interactions of various

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35508-5_22

V. Atluri et al. (eds.), *Research Advances in Database and Information Systems Security*

RBAC components (mechanisms). Another important benefit of RBAC is the ability to modify the security policy to meet the changing needs of an organization. However, most of the research work today, has being focused on the support of the DAC philosophy using the RBAC approach [4,5,14,21]. The scope of this paper is to present an exploitation of RBAC features in articulating a security policy for access control in medical database systems, where both mandatory and discretionary features are needed. To achieve this, the mechanisms of the already known MEDAC security policy have been redefined on the basis of RBAC components. The resulting security policy (eMEDAC) is an enhancement of MEDAC that offers significant advantages, especially in access control, administration and storage requirements, with some drawbacks of additional computational workload.

2. THE MEDAC SECURITY POLICY

Security policies are sets of principles and high level guidelines that concern the design and management of access control systems [2]. Mechanisms are low level software and hardware functions, which can be configured to implement a security policy [15]. In general, there are no policies that are better than others. This is because, not all systems have the same protection requirements. Policies suitable for a given system may not be suitable for another. The choice of security policy depends on the particular characteristics of the environment to be protected. However, in recent years there is an increasing consensus that there are legitimate policies that have aspects of both mandatory and discretionary security models. Role-based policies are an example of this fact. In health care environments there is a need for a security policy that is able to satisfy all the security components (availability, integrity and confidentiality). As has been demonstrated previously [7,12,13], both DAC and MAC when used separately have their limitations in achieving this. As a result, we have previously proposed a security policy called MEDAC (MEdical Database Access Control) based on both MAC and DAC approaches, that has been proved to be able to satisfy all the needed security requirements.

2.1 OVERVIEW OF THE ORIGINAL MEDAC SECURITY POLICY

The MEDAC security policy is based on the Bell-LaPadula model [2] and takes advantage by utilizing the characteristics of discretionary, mandatory and role-based approaches. A detailed presentation of MEDAC can be found in [7,11,12] and [13]. It is based on the following principles:

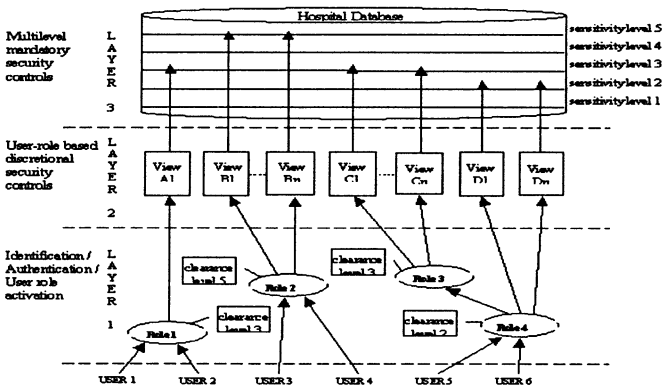

Figure 5.1 The MEDAC model.

- Every authorized person has the right to access the system. For this reason, each user accessing directly the system must have an account allowing him to log into the system. Every account is associated to a predefined user role. This user role represents the user task in the application.

- Every user role has a clearance level and a category set. The category depends on the nature of the user role and the data sets it needs to access. The clearance level of the user role represents its trustworthiness.

- Every data set is assigned a sensitivity level and a category set. The sensitivity level reflects its sensitivity depending on its context, content, exterior factors or specific situations (e.g. related to sensitive cases). The category depends on the use and the nature of the data set.

The levels within a certain category are completely ordered (hierarchical), while the categories are partially ordered.

The MEDAC security policy is formed of three layers that every user requesting to access the database has to pass through. More specifically:

- First layer: In this layer, the identification and authentication of a particular user is performed by the security mechanisms of the host system. After his identification, the user activates a user role from the set of roles that has been assigned to him initially.

- Second layer: User roles in this layer are permitted to access the database via a discretional access control. Each user role has the authorization to see a certain view and to exercise just the authorized access rights. The

view has been predefined depending on the need-to-know requirements of the users performing this role.

- Third layer: Users roles are required to access the database satisfying the multilevel access control. Depending on the clearance level of the specific user role, it will be able to access just the part of the database which is defined according to the dominance relation (i.e., the sensitivity level of the information which can be read by the user role is less or equal to its clearance level).

MEDAC has been implemented already in the AHEPA general hospital of Thessaloniki. This experimental implementation has produced successful results [12,13].

2.2 SUPPORTED CHARACTERISTICS OF THE MAJOR SECURITY MODELS

'DAC' Characteristics. Discretionary controls are supported in the MEDAC policy by views that are used to define the sets of permitted types of access that can be executed by specific user roles to data sets. Schematically, the DAC aspects of the MEDAC policy are handled in the 2nd layer of the MEDAC model.

'MAC' Characteristics. In the MEDAC policy the multilevel model of MAC is supported. An important characteristic of the MEDAC policy is that mandatory security controls are used additionally to the discretional security ones. Mandatory access control is based on the sensitivity of the personal and medical data of patients in order to ensure their protection in a strict way [9,13]. Furthermore, the use of multilevel security has made it possible to provide generalized explanations (cover stories) for unavoidable observable information that would otherwise lead to partial or complete inference of sensitive information [10]. Schematically, the MAC aspects of the MEDAC policy are handled in the 3rd layer of the MEDAC model.

'RBAC' Characteristics. MEDAC also partly supports the RBAC approach, mainly by means of user roles and the user role hierarchy. The use of the concept of user roles has made it possible for every user to have access just to that part of the application he needs to use in order to perform his specific task [10]. Schematically, the RBAC aspects of the MEDAC policy are handled in the 1st layer of the MEDAC model. However, although MEDAC takes into account the RBAC concept, it does not utilize the RBAC features fully, since it has been developed independently, before RBAC was widely accepted.

3. THE ENHANCED MEDAC SECURITY POLICY (EMEDAC)

3.1 THE RBAC COMPONENTS USED

Role-based policies allow the specification of authorizations to be granted to users on objects like in the discretionary approach, together with the possibility of specifying restrictions on the assignment or on the use of such authorizations. They also provide a classification of users according to the activities they execute. Analogously, a classification is recommended for objects according to their type or to their application area [15].

RBAC is an emerging role-based policy [6,17,18]. Role hierarchies are used in RBAC for structuring roles to reflect an organization's lines of authority and responsibility. More powerful (senior) roles are placed toward the top and less powerful (junior) roles toward the bottom of the diagrams representing them [19]. To limit the scope of inheritance in hierarchies, a special kind of roles is used, named private roles [18]. Permissions are approvals of particular modes of access to objects in the system. Permissions are always positive and confer on their holder the ability to perform an action in the system [19]. Constraints are another important aspect of RBAC and are sometimes argued to be the principal motivation for RBAC. A common example is that of mutually exclusive roles. Constraints are a powerful mechanism for enforcing higher level organizational policy. Once certain roles are declared mutually exclusive, there's less concern about assigning individual users to roles [19].

According to [18], RBAC provides a means for articulating policy rather than embodying a particular security policy. Hence, RBAC could be used for the redefinition of mechanisms of the original MEDAC security policy, by extending its role-based concepts and replacing stored mandatory security labels with derived ones. In this way the enhanced MEDAC (eMEDAC) security policy should utilize fully the RBAC features to support the changing administration needs of healthcare organizations.

3.2 OVERVIEW OF THE EMEDAC MODEL

The eMEDAC model uses RBAC features that have been tailored to meet the specific needs of a healthcare information system. In such a context, the access-matrix could be implemented with the RBAC permission mechanism to support the DAC features needed. In order to satisfy the MAC requirements for accessing sensitive patient data, the role hierarchy and constraint mechanisms are used in the following way.

Because there is a strong similarity between the concept of a security label (consisting of a security level and a set of categories) and a role [18],

security levels could be implemented by means of the position (depth) of a role (node) in the hierarchy and categories could be derived from the ancestor nodes reached when moving towards the top of the hierarchy. As a result of this, security labels are not stored but are directly derived from the hierarchy.

However, there is a need, e.g. for the medical and the ward activities of a Health Information System (HIS), to protect in a strict way the privacy of patient personal and medical data [9]. This requirement results in a definite number of required (mandatory) security levels. Therefore, each role in the hierarchy is assigned to a specific level number that cannot exceed a predefined limit. This is a constraint that dominates the construction of hierarchies and is useful mainly in a decentralized access control system.

In order to solve the problem of assigning to a given role a level number that is not necessarily equal to the level number of its ancestor minus one (e.g. the role is of level 2 and its ancestor is of level 4) we use dummy nodes between the two hyper nodes in the hierarchy.

Every hierarchy has at least one level of initial nodes that are assigned to the highest (or lowest) security level. Every other node is a descendant of its ancestor node and is placed in a different level. As a result of this, the category set of a node could be the set of all its ancestors (in case they are not dummy nodes) that are nearest to the top of the hierarchy (first ancestors).

Until now we assumed that each node is assigned to a different level than its ancestor node. As a result, going down a level in the hierarchy the level number is changed by one. However, this fact introduces restrictions when for example the security lattice has depth five and the role hierarchy has depth ten. To solve this problem we propose a special connection between nodes of the same level, named link.

To support the above features the concept of the Hyper Node Hierarchy (HNH) mechanism has been introduced.

3.3 THE HYPER NODE HIERARCHY MECHANISM

A Hyper Node Hierarchy (HNH) is a set of nodes and connections. Each (regular or dummy) node is connected to another node by a branch or a link. A branch is the connection of a node to its ancestor node in the above level of the same HNH. Links are connections that are used between nodes of the same level. A link is used in order to specialize the inner structure (in the form of a sub-hierarchy) of a hyper node, by introducing its descendant starting from the same level (otherwise branches could be used).

The HNH mechanism provides totally ordered hierarchies and supports multiple inheritance by using multiple occurrences of the same node. A formal definition of the HNH mechanism is given below.

Definition 1. The HNH mechanism:

- N, a set of nodes
- C, a set of connections
- $HN \subseteq \times C$, each hyper node HN is a double $\{N, C\}$,
- $HNH \subseteq HN \times HN$, is a totally ordered hyper node hierarchy.
- HN and DN, disjoint sets of (regular) hyper nodes and dummy nodes respectively,
- BC and LC, disjoint sets of branches and links respectively,
- a node N_i has a level (depth in the hierarchy) of number i,
- $BC : N_i \rightarrow N'_{i\pm1}$, branch is a function mapping a node to its ancestor node at the above level,
- $LC : N_i \rightarrow N'_i$, link is a function mapping each node to its ancestor (hyper) node at the same level,

The security labels (consisting of a security level and a category set) are implemented in a HNH as defined below.

Definition 2. Implementation of the security level and the category set in a HNH:

- a hyper node HN_i has a security level of number i,
- the category set of a hyper node HN_i is consisted of all its possible first ancestors.

The HNH concept is used to construct user role and data set hierarchies.

3.4 THE USER ROLE HIERARCHY

More specifically, the User Role Hierarchy (URH) consists of a HNH having at least one level of hyper nodes with connection "All Users". Then, these roles may be specialized into one or more levels of (senior) user roles.

The number of levels in a URH is predefined depending on the granularity of the control needed (for example in military environments it is considered to be four, respectively for Unclassified, Confidential, Secret, and Top Secret). User roles placed on the top of the URH are of lowest level (e.g. 1).

By its turn, the use of links introduces new sub-hierarchies where the same procedure can be repeated until the security administrator is satisfied that the

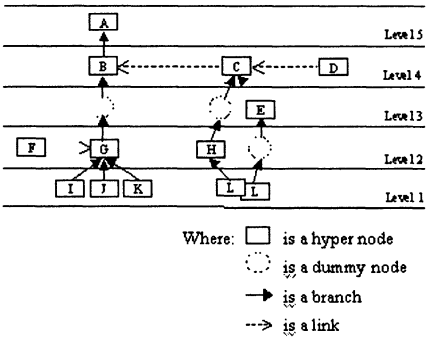

Figure 5.2 The Hyper Node Hierarchy (HNH).

specialization achieved is enough for the specific application. The number of new sub-hierarchies resulting from the use of links of hyper nodes is not predefined, allowing the security administrator the ability to analyze any hyper node, reaching to any depth of analysis in the tree, depending on the special needs of the application. For example in a specialized hospital (e.g. cancer hospital) the set of user roles needed is more sophisticated than in a general hospital. However, each sub-hierarchy resulting from a hyper node (to analyze further its inner structure) has a predefined number of levels that is derived from the level of the source hyper node. This sub-hierarchy cannot include levels lower than the level of the source hyper node (e.g. it includes only levels 3 to 5 for a source hyper node level of 3).

Concerning the derivation of the clearance level and the category set of a given user role, we always initialize the level to be one (lowest level) and the category to be the empty set. Then, in the hierarchy we find the first entry (occurrence) of the role and we move towards the top of the hierarchy by following its connection to its ancestor node. In case it is a branch, we increment the clearance level. In case the ancestor node is a regular node we assign it to the user role category. When the connection is 'All Users' we stop moving and we add the last category found to the category set. The same procedure is repeated for every subsequent occurrence of the user role.

The derivation of the security (clearance) level of a user role UR is described in the following algorithm 1.

Algorithm 1. Derivation of the security level.

security_level : = *lowest_level*

until *connection*(UR) = 'All Users' **do**

find UR

```
    if type_of_connection(UR) is branch
      then security_level : = security_level + 1
    UR : = connection(UR)
  od
```

The following algorithm 2 describes the derivation of the category set of a user role UR.

Algorithm 2. Derivation of the category set.

```
find UR
category_set : = fcs(UR)
function fcs(A)
  fcs : = {}
  if type_of_node(A) is hyper then fcs : = node(A)
  if connection(A) ¡¿ 'All Users' then
    Anc : = connection(A)
    find Anc
    if fcs(Anc) ¡¿ {} then fcs : = fcs(Anc)
    findnext A
    while A exists do
      fcs := fcs ∪ fcs(A)
      findnext A
    od
  fi
fend
```

The use of the URH is different from the use of a classical RBAC hierarchy. In the second layer of the eMEDAC model, the URH is used to inherit permissions. However, in the third layer the URH is used for deriving security labels consisting of clearance levels and category sets. As a result, the two basic principles of the MAC model (18], which are the simple-property (or read-up protection) and the *-property (or write-down protection), are satisfied in a subsequent step by the secure DBMS implementing the following dominance relationship. A security label $S1 = (L1, C1)$ dominates a security label $S2 = (L2, C2)$ if and only if $L1 \geq L2$ and $C1 \supseteq C2$, where L is the security level and C is the category set. [2].

In order to demonstrate the above concepts we describe below a representative subset of our pilot implementation. The general model of a HIS archi-

tecture that has been defined by the ISHTAR group [1], has been used as the basis of the implementation since it corresponds to the structure of the AHEPA General Hospital that is used as our test-bed side. The four main categories of activities defined in this model have been used also as the user role and data set categories : ward, medical, administration and logistics. According to the organizational schema of the hospital, the URH shown in the following figure has been constructed [12], [13].

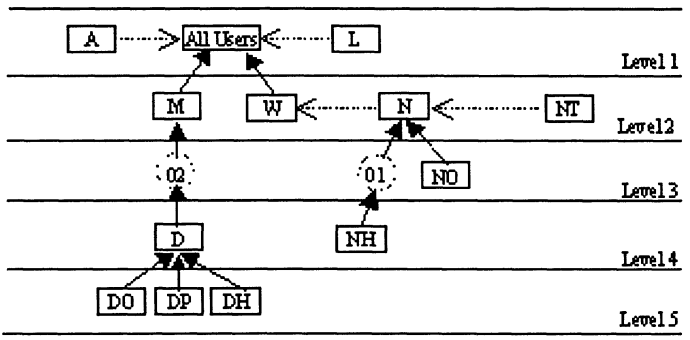

Figure 5.3 The User Role Hierarchy (URH) example.

To derive the security label (consisting of a clearance level and a set of categories) of the user role NH we initialize the level to be one. We first find in the data structure the entry of node NH. Then we follow its connection that is a branch to the dummy node 01 and the level is incremented by one. Again, following the branch of dummy node 01 to the user role N, the level is incremented by one. Then we follow the link of N to W. Because the connection of W is 'All Users' we stop moving, the node W is included in the category set and the level is incremented by one having now the final value four.

3.5 THE DATA SET HIERARCHY

The sensitivity of data sets is expressed in a similar Data Set Hierarchy (DSH). The same as above procedures are applied, except that data sets placed on the top of the DSH are of highest level (e.g. 5). Each sub-hierarchy that is generated for analyzing further the inner structure of a data set has a predefined number of levels and cannot include levels higher than the level of the source data set (e.g. it includes only levels 1 to 3 for a data set level of 3). Cover stories could be implemented by using the concept of private roles [18], but for data.

Concerning the derivation of the sensitivity level and the category set of a given data set, we always initialize the level to be five (highest level) and the

category to be the empty set. Then, in the hierarchy we find the first entry (occurrence) of the data set and we move towards the top of the hierarchy by following its connection to its ancestor node. In case it is a branch, we decrement by one the sensitivity level. In case the ancestor node is a regular node we assign it to the data set category. When the connection is 'All Data' we stop moving and we add the last category found to the category set. The same procedure is repeated for every subsequent occurrence of the data set.

The derivation of the security (sensitivity) level of a data set DS is described in the following algorithm 3.

Algorithm 3. Derivation of the security level in DSH.

security_level : = *highest_level*
until *connection*(DS) = 'All Data' **do**
  find DS
  if *type_of_connection*(DS) **is** *branch*
    then *security_level* : = *security_level* - 1
  DS : = *connection*(DS)
od

The algorithm for the derivation of the category set of a data set DS is identical to the algorithm 2 (defined for user roles).

The data sets are allocated at each node depending on the need to know principle of user roles and the specific security constraints that are implemented by the secure design methodology. Taking under consideration the need to know requirements of user roles and the specific security constraints that have been implemented by the secure design methodology [12,13], the DSH shown in the following figure has been identified for the pilot implementation.

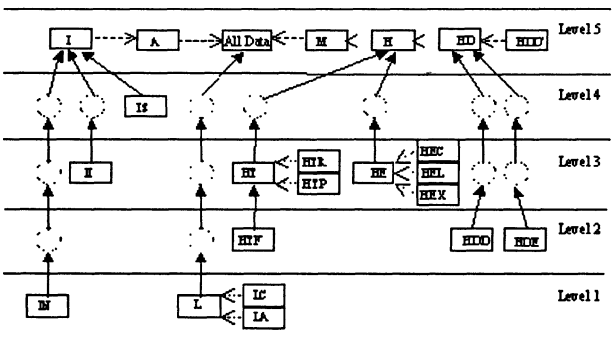

Figure 5.4 The Data Set Hierarchy (DSH) example.

4. EXTENSIONS FOR DISTRIBUTED MEDICAL DATABASES

Until now, the discussion addressed mainly the needs of centralized database systems. However, the presented eMEDAC security policy does not take under consideration any additional requirements for distributed database systems where centralized administration of access rights is a very difficult task. In such a case, access control systems might allow administrative authority for a subset of objects to be delegated by the global security administrator to other local (regional) security administrators. Control over the regional administrators could be also centrally administered. The construction of local HNH (sub-hierarchies for local user roles and data sets) could be accomplished by regional administrators without having the possibility to assign security levels higher than the corresponding global user roles and data sets that have already defined by the global security administrator. This process of delegation could be repeated within each region to set up subregions and so on [15].

However, because of the specific needs of the distributed systems design and operation, the introduction of new proposals or the extension of the already existing ones in eMEDAC is required. In our opinion, the main issue is the location of users and data sets. We have already studied the allocation of data sets in the context of a secure distributed database design methodology [8]. However, we believe that it is worth to study the impact of different user locations in the determining process of access control for at least two situations. First, knowing the location, which may be a site or an administrative domain, from where the user is accessing remotely the database system, and having previously defined the trustworthiness of this location, the decision of what set of roles he can activate depends, besides his task requirements, on the trustworthiness of his current location. Second, knowing the administrative domain wherein the user is acting, the decision of what kind of access is allowed to him depends not only on his current set of activated roles but also on its particular user location.

In such a context, we consider the use of global and local access control mechanisms (user roles, data sets, hierarchies and permissions) in distributed medical database systems. As has been proposed in [20], global roles could be authorized with global permissions to access an intranet's resources on the basis of different local roles in different servers (sites). However, this assumption introduces lack of flexibility (e.g. the granularity used when defining the permission set) and alike behavior of the access control system on local and remote access requests. We prefer to define separately each one subject (global or local user role) and its permission set to access remotely any defined object of the system from any defined location. Our perspective is to reduce autho-

rized privileges of a given user role while its location is going farther on either on organizational or security level.

As with user roles and data sets, a user location hierarchy is resulting. The User Location Hierarchy (ULH) is a means for representing the organizational structure of the healthcare establishments involved in the application. Furthermore, in a ULH can be included national or international administrative domains, as well as a number or all the possible workstations from where a given user can log in the distributed medical database system. As a result of this the total amount of specifications for defining all the needed data can easily become huge and very difficult to manage. This huge amount of data can be reduced however dramatically by exploiting the inheritance in the already presented three types of hierarchies.

In each administrative domain the following actions should be accomplished for the definition of those hierarchies:

- Inherit catalogues and hierarchies of user roles, data sets and user locations from the upper (global) administration domain,
- Refine each hierarchy to meet the special local needs of the specific administrative domain.

Local security administrators have to decide about the authorization of subjects of other administrative domains on objects that belong to their domains, in order to reject the penetration that other administrators can do in the access control policy of each administrative domain [3]. This can be accomplished by eliminating the user role permissions set on the basis of the assigned set of user locations. To achieve this a third dimension, concerning the user location, could be added to the classical access matrix.

In our oncoming DIMEDAC policy, which is an extension of the eMEDAC security policy for distributed database systems, we have attempted to cover in a satisfactory way those needs. However, this is beyond the scope of this paper.

5. CONCLUSION AND FUTURE WORK

The eMEDAC security policy uses RBAC components to provide an enhanced redefinition of a number of mechanisms of the already known MEDAC security policy. The main innovation of this paper when compared with our previous MEDAC papers is related to the use of RBAC mechanisms in all the three layers of the original MEDAC security policy.

As explained in the paper, the use of RBAC mechanisms in the third layer raises two several problems: the construction of totally ordered security levels and the definition of different number of security and hierarchy (depth) lev-

els. As a solution, we proposed the concept of Hyper Node Hierarchy that is a unique mechanism for discretionary and mandatory access control that provides: permission inheritance, URH and DSH hierarchies of any depth, derived security levels and categories, access control data (DSH) stored separately from the application data holders, etc.

Because MAC, DAC and RBAC are supported in eMEDAC at the same time, access control becomes more efficient and hence particularly suited to security critical environments, like the health care ones, where all three aspects of security are important.

Administration also becomes easier because the eMEDAC policy, besides the utilization of RBAC administration advantages, every HNH can be centrally defined and replicated to each site. Then local administrators can add more refinement levels for satisfying their local needs, without having the ability to override a predefined limit of security level.

A possible disadvantage is a non-significant, to our opinion, loss in performance (or higher computational workload) which stems from the need for deriving instead of using already stored security labels. However, there is expected to be a significant saving of storage space because, instead of storing security labels for every object (depending on the selected granularity, e.g. for field granularity the storage cost becomes considerable high), only the necessary nodes of HNH are stored.

In our future work we are going to investigate and compare the computational and storage cost of performing access control by deriving security labels instead of using stored ones. We are also working to study the extensions of the eMEDAC that are necessary in distributed database systems. Furthermore, the case of distributed systems with mobile parts is expected to introduce a number of new factors affecting the overall access control system.

References

[1] Blobel B., Bleumer G., Muller A., Flikkenschild E. and Ottes F. (1996). Current security issues faced by health care establishments. *HC1028 Implementing Secure Healthcare Telematics Applications in Europe (ISHTAR).*

[2] Castano S., Fugini M., Martella G. and Samarati P. (1994). *Database security*, Addison Wesley.

[3] Ceri S. and Pelagatti G. (1985). *Distributed Databases: Principles and Systems*, McGraw-Hill.

[4] Essmayr W., Kapsammer. E., Wagner R.R. and Tjoa A.M. (1998). Using role templates for handling recurring role structures. *Proceedings of*

the Twelfth International IFIP WG11.3 Working Conference on Database Security.

[5] Ferraiolo D. and Kuhn R. (1992). Role-based access control. *Proceedings of Fifteenth National Computer Security Conference.*

[6] Ferraiolo D., Cugini J. and Kuhn R. (1995). Role-based access control (RBAC): Features and motivations. *Proceedings of the Annual Computer Security Applications Conference.*

[7] Khair M. (1996). Design and Implementation of Secure Database Systems with Application on Heatlhcare Information Systems. Dissertation.

[8] Khair M., Mavridis I. and Pangalos G. (1998). Design of secure distributed medical database system. *Proceedings of Database and EXpert systems Applications, DEXA'98.*

[9] Pangalos G. (1995). Medical Database Systems Security. EEC/AIM, SEISMED (A2033) Project Report, No. AIM/SEISMED/SP-07/20-04-95/3.

[10] Pangalos G. (1996). Secure medical databases. *Proceedings of the IMIA security conference.*

[11] Pangalos G. and Khair M. (1996). Design of a secure medical database systems. *Proceedings of IFIP/SEC'96, the Twelfth International Information Security Conference.*

[12] Pangalos, G., Gritzalis D., Khair, M. and Bozios, L. (1995). Improving the security of medical database systems. *Information Security - the Next Decade* (eds. J. Eloff and S. Von Solms), Chapman & Hall.

[13] Pangalos, G., Khair, M. and Bozios, L. (1995). An integrated secure design of a medical database system. *MEDINFO'95, The Eighth World Congress on Medical Informatics.*

[14] Poole J., Barkley J., Brady K., Cincotta A. and Salamon W. NISTIR 5820 Distributed Communication Methods and Role-Based Access Control for Use in Health Care Applications. http://www.itl.nist.gov/div897/staff/poole/documents/nistir5820.htm

[15] Sandhu R. and Samarati P. (1997). Authentication, access control, and intrusion detection. *The Computer Science and Engineering Handbook.*

[16] Sandhu R. (1996). Access control: The neglected frontier. *Proceedings of First Australian Conference on Information Security and Privacy.*

[17] Sandhu R. (1996). Rationale for the RBAC96 family of access control models. *Proceedings of the First ACM Workshop on RBAC.*

[18] Sandhu R. (1998). Role-based access control. *Advances in Computers*, **46**, Academic Press.

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

[20] Tari Z., and Chan S.-W. (1997). A role-based access control for intranet security. *IEEE Internet Computing*, pp. 24 - 34.

[21] Vandenwauver M., Govaerts R. and Vandewalle J. (1997). Role based access control in distributed systems. *Communications and Multimedia Security*, **3**, pp. 169-177.