# Risk Based Security Analysis of Permissions in RBAC

Nimal Nissanke and Etienne J. Khayat

Centre for Applied Formal Methods, London South Bank University
103 Borough Road, London SE1 0AA, UK

**Abstract.** Because of its vulnerability to errors and, hence, unauthorised access, assignment of access rights is a critically important aspect of RBAC. Despite major advances in addressing this clearly using formal models, there is still a need for a more robust formulation, especially incorporating strict guidelines on assignment of access rights and how to perform such tasks as delegation of access rights. In this respect, this paper proposes a precise mathematical framework, capable of considering important factors such as the relative security risks posed by different access operations when performed by different users. This is based on a novel concept of a security risk ordering relation on such tasks, to be established by a detailed independent risk assessment process. In the case of lack of information on security risks, the approach makes conservative assumptions, thus forcing the security analyst to re-assess such situations if he disagrees with this default interpretation. The risk ordering relation is central to a security-orientated definition of role hierarchies and a security-risk minimising strategy to role delegation.

## 1 Introduction

Role Based Access Control (RBAC) is a widely used access control mechanism whereby access rights to users (subjects) are granted on the basis of their roles in an institution rather than as individuals. Allocation of access rights, whether it takes place as a result of system administrator's duties, such as permission assignment to roles, or discretionary actions exercised by subjects higher up in role hierarchy, such as delegation, is a process vulnerable to errors and existence of unforeseen loopholes that could compromise the system security.

The growing interest in RBAC is evident from the large number of works devoted to it. Notable among them are the works [5, 6, 9, 10] characterising a hierarchy of RBAC models with increasing sophistication, dealing with role hierarchies, potential conflict of interests between roles, etc. A major outcome of these developments is the recognition by the research community of the need for a standard [1] aimed at a unified model for RBAC. Related is also our own work [7], providing a formal state-based model for the core RBAC [1]. Important issues related to delegation are elaborated in a number of works; with [2] dealing with a basic but sufficiently detailed model of delegation, [8, 11, 12] showing a practical scenario of the implementation of delegation and [13] giving detailed mathematical models for various types of delegation.

Despite the above advances, there seems to be little definitive rules or guidelines that govern the assignment of access rights in RBAC and clarify the means by which

Nissanke N. and J. Khayat E. (2004).
Risk Based Security Analysis of Permissions in RBAC.
In *Proceedings of the 2nd International Workshop on Security in Information Systems*, pages 331-340
DOI: 10.5220/0002687403310340

the goals of the security mechanisms are to be achieved. System invariants for access rights allocation that should always be respected are not sufficiently well defined in the literature, including those cited above. Allocation of access rights is often based on informal rules based on past experience or inherited institutional practices. In order to overcome the above deficiencies, this paper introduces a novel approach ensuring a consistent and systematic interpretation of security requirements and a compatible and effective way to enforce the security arrangements.

Our approach is based on the concept of a *risk ordering relation* [4] expressing the relative risk posed by a subject of a particular role performing a particular task, compared to the same posed by a similar subject-task combination. It is a mathematical concept designed both to introduce rigor into security modelling and to eliminate ambiguities, omissions and inconsistencies in the risk assessment process. Risk ordering itself is established through an appropriate independent security risk analysis of the organisation. From the perspective of security risk analysis, the approach offers two major benefits: firstly, it makes explicit the form of the information required from such an analysis and, secondly, it prompts the security expert to question his security assessment, thereby improving the quality and comprehensiveness of the process, as well as the end–product of the security risk assessment. Questioning of the security assessment is achieved by a default conservative interpretation of risks levels whenever there is a lack of information on security risks. According to this, any task with a possible inadequate consideration of risk is placed conservatively in a lower *security risk band* by default, alerting the security analyst to reconsider its risk nature if such an interpretation is undesirable. Turning to the modelling of RBAC, our work proposes certain security principles for permission assignment to roles and for subject-invoked role delegation.

The paper has the following structure. Section 2 introduces the required basic concepts of RBAC and the relevant mathematical definitions used later. Section 3 presents the concept of *security risk ordering relation*, expressing the risks posed by different combinations of roles and tasks (permissions) relative to one another. Section 4 states the proposed principles of allocation of access rights in RBAC in a precise manner. Section 5 presents a case study drawn from the domain of health care illustrating the application of the latter principles. Section 6 concludes the paper with a summary of achievements.

## 2 Basic Concepts and Mathematical Preliminaries

The purpose of RBAC is to determine at run–time whether to allow, or deny, a user (*subject*) accessing a required resource (*object*) based on access rights granted to the roles that subjects perform in the organisation. This section introduces the basic RBAC concepts relevant to these issues and an appropriate notation for the discussion; see also [7]. Our formulation is based on the following basic types of entities: *SUBJECT* denoting the set of all possible users (subjects) of the computer system (including any non-human agents), *OBJECT* the entities (objects) being accessed by the subjects, *ROLE* the roles in the capacity of which the subjects derive the access rights to the objects concerned, and *OPERATION* the set of operations that may be performed on the objects. Disregarding here the applicability of operations to specific objects, the set of all possible

tasks are denoted by *TASK*, defined as

$$TASK = OPERATION \times OBJECT \tag{1}$$

Associated with the above are the following functions [Note: $\mathbb{P}$ denotes the power set of its operand set (on the right)]:

$$SubjectRoles : SUBJECT \rightarrow \mathbb{P}\, ROLE \tag{2}$$

$SubjectRoles(s)$ giving the set of roles associated with each subject $s$, and

$$Permissions : ROLE \rightarrow \mathbb{P}\, TASK \tag{3}$$

$Permissions(r)$ giving the set of tasks authorised for each role $r$. When dealing with individual permissions, it is convenient to have the elements of the following set

$$PERM \subseteq ROLE \times TASK \tag{4}$$

the elements of which denote roles and the applicable tasks.

In relation to delegation of access rights or tasks in hierarchical RBAC, we introduce two types of delegation: lateral delegation (a role delegating its duties to another role lying at the *same* level of the hierarchy) and downward delegation (a role delegating its duties to a junior role). A record of such delegated roles to each particular role may be maintained by the following functions:

$$lat\_del\_roles, down\_del\_roles, delegated\_roles : ROLE \rightarrow \mathbb{P}\, ROLE \tag{5}$$

$lat\_del\_roles(r)$, $down\_del\_roles(r)$ and $delegated\_roles(r)$ giving, respectively, the roles delegated to role $r$ laterally, downward and in total. Note that for each of the above, $r$ cannot be delegated to itself. Together the above satisfy

$$\forall\, r \in ROLE \bullet delegated\_roles(r) = lat\_del\_roles(r) \cup down\_del\_roles(r) \tag{6}$$

Turning attention to conflicts of interests (COI), there are two kinds of separation of duties that need to be taken into account in determining the permitted delegations of roles and tasks, namely: a) Static Separation of Duty (SSD), which concerns the prevention of any conflict of interests arising from the mere assignment of such roles to the same subject, and, b) Dynamic Separation of Duty (DSD), which concerns the concurrent exercise of such roles by any subject at the same time and not whether they can be assigned to the same subject. With the above in mind, let us introduce three symmetric and irreflexive binary relations *SSD*, *COI* and *COI* on *ROLE*, such that $COI = SSD \cup DSD$.

## 3 Security Risk Ordering

In general, risk expresses a combined measure of the likelihood of a hazardous, or a harmful, event occurring and the ensuing consequences should it ever take place. In computer security, such events include intrusion, tampering with data, eavesdropping, etc., violating system security properties such as *confidentiality*, *integrity* and *availability*. Security threats not intensifying, the risk of such events taking place usually reduces

with increasing protection. Risk assessment is an exercise in its own right and is beyond the scope of this paper. What is important here is, however, the outcome of the risk assessment process and, in particular, the relative risks posed by various security threats relative to one another.

Risk ordering relation, introduced here, relies on a comparison of risks arrived at by an appropriate independent risk assessment process. It is denoted by $\sqsubseteq$ and has the form

$$\sqsubseteq: PERM \leftrightarrow PERM \tag{7}$$

Its meaning is such that, given two permissions $p_1$ and $p_2$, where $p_1, p_2 \in PERM$, $p_1 \sqsubseteq p_2$ signifies that $p_2$ is more, or equally, secure compared to $p_1$ or, alternatively, $p_1$ carries a higher, or an equal, security risk compared to $p_2$. $\sqsubseteq$ is reflexive and transitive, but not necessarily symmetric or antisymmetric. We decompose $\sqsubseteq$ into two relations:

- $\preccurlyeq$: a partial order relation over the elements of *PERM*, which orders their risk levels. If $p_1 \preccurlyeq p_2$, then $p_1$ carries a higher security risk than $p_2$, unless $p_1$ and $p_2$ denote the same permission.
- $\approx$: an equivalence relation between the elements of *PERM*. If $p_1 \approx p_2$, then $p_1$ and $p_2$ are identical in terms of security risk.

As a consequence of this decomposition, $\sqsubseteq$ is the union of $\preccurlyeq$ and $\approx$ . In other words, for permissions $p_1$ and $p_2$, $p_1 \sqsubseteq p_2$ if and only if $p_1 \preccurlyeq p_2$ or $p_1 \approx p_2$.

The relation $\sqsubseteq$ is best depicted in the form of a graph, as in Figure 1(a), showing the ordering of the permissions. Since the risk analysis is performed by human security analysts, the relation $\sqsubseteq$ may contain gaps, inaccuracies and inconsistencies. Therefore, following [4], we use the concept of *risk band* to alert the risk analyst to such deficiencies. The idea is to interpret any lack of information conservatively in favour of greater provision of security. In effect, risk bands extend the graph of $\sqsubseteq$ with numerically indexed risk bands such that permissions carrying relatively greater security risks are placed in higher risk bands, while the more secure permissions in lower risk bands; see Figure 1(b). In the event of insufficient information as to where a particular permission is to be placed, it is interpreted as an indication that the permission concerned is to be placed in the highest possible risk band, subject to any constraints imposed by other pairs in the relation $\sqsubseteq$ . Any disagreement with this default interpretation obliges the security risk analyst to clarify the relative risk levels of the permissions concerned more accurately, thus helping to refine the risk ordering relation and, thereby, making it more complete, accurate and consistent with the required security requirements. The graph of the relation $\sqsubseteq$ , extended with risk bands, is referred to as the *risk graph*; see Figure 1(b). The arcs in the graph are assumed to run upward and the reflexivity of the permissions in the relation $\sqsubseteq$ are not shown in the graph to reduce clutter. The risk bands are numbered from 1 to some *n*, higher indices signifying greater risk. Risk graph, corresponding to a specific relation $\sqsubseteq$ , is to be determined according to the following rules:

- Permissions with the highest security risk, or the least secure ones, (i.e. those in the *n*th risk band) are exactly:
  a) The permissions that are lowest in the partial order relation $\preccurlyeq$, but not related by $\approx$ to any other permission in $\preccurlyeq$ .

b) Any other permissions related by $\approx$ to the ones just mentioned in (a) above.

– If there exist two distinct permissions $p_1$ and $p_2$ such that a) $p_1 \preccurlyeq p_2$, b) $p_1$ is the only immediate predecessor so related to $p_2$, and c) $p_1$ is in risk band $i$, then $p_2$ is in risk band $(i-1)$. If $p_2$ has several immediate predecessor permissions, then its risk band index would be one less than the lowest risk band index of those predecessor permissions.
– If there exist two permissions $p_1$ and $p_2$ such that $p_1 \approx p_2$, then $p_1$ and $p_2$ are in the same risk band.

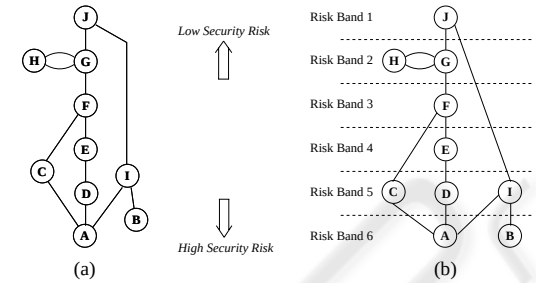


**Fig. 1.** (a) Risk ordering relation (b) Risk graph.

Associated with the risk graph is a risk distance between two permissions of the form: $RD(p_1, p_2) = RB(p_1) - RB(p_2)$, where $RB(p)$ gives the risk band index of a given permission $p \in PERM$, taking the sign into consideration. From the security risk perspective, two permissions $p_1$ and $p_2$ are said to be *risk-comparable* if and only if they are equivalent through $p_1 \approx p_2$ or are in different risk bands (i.e., $RB(p_1) \neq RB(p_2)$). If they are in the same risk bands (i.e., $RB(p_1) = RB(p_2)$), but are not equivalent (i.e., $p_1 \not\approx p_2$), then they are said to be *risk-non-comparable*.

## 4 Principles of Allocation and Delegation of Permissions

This section formulates several principles to be followed when allocating access rights. These concern the cases of permission assignment to roles and delegation of access rights.

### 4.1 Relations on Roles

The hierarchical model of RBAC [1], also known as $\text{RBAC}_1$ [10], places the roles in a hierarchy in accordance with the functional requirements of the organisation and other considerations such as the skills, the competence, the past experience, etc., required as part of the job descriptions. However, this is based mathematically on a simple set–theoretic characterisation of roles as a partial order $\leqslant$, namely, for any two roles $r_1$ and $r_2$ as

$$r_1 \leqslant r_2 \Rightarrow \mathit{Permissions}(r_1) \subseteq \mathit{Permissions}(r_2) \tag{8}$$

It is important to note that (8) characterises only a hierarchical relationship between roles with inheritance of permissions of juniors by seniors. In our view, however, there are other notions of seniority relations of relevance to security. Of particular interest here is a relation that characterises roles performing different kinds of activities but being equivalent. This is because, for example, being equivalent in status would allow the delegation of roles that deal with authorisations, etc. With this in mind, this work uses three relations on roles, two of them being

- A partial order relation, $\leqslant$ , as defined in (8), dealing with hierarchical inheritance of permissions of junior roles by their seniors.
- An equivalence relation, $\simeq$ , dealing with equivalence of roles belonging to different categories of roles in terms of their status.

leaving the third relation for Section 4.2. In relation to $\leqslant$ and $\simeq$ , as an example, consider the members of a hospital in two different role categories: *medical* and *nursing*. According to $\leqslant$ , roles in the *medical* category may be ordered hierarchically as: *resident* $\leqslant$ *surgeon* $\leqslant$ *consultant*, whereas those in the *nursing* category as *nurse* $\leqslant$ *senior_nurse* $\leqslant$ *chief_nurse*. Furthermore, using the equivalence relation $\simeq$ , it is possible to relate the chief nurse and the surgeon as *chief_nurse* $\simeq$ *surgeon* in order to convey that they have the same seniority status and, therefore, they are eligible to delegate, for instance, certain authorisation tasks between them.

### 4.2 Principle I: Permission Assignment to Roles

As noted above, roles in RBAC are assigned permissions by associating them with the tasks that they are authorised to perform. In most cases, this association is based solely on the functional requirements of the organisation. Prior to such assignment of permissions to roles, however, a security risk assessment needs to be performed in order to verify if the functional requirements would induce any unintended security threat to the organisation's assets. This is where the security risk ordering relation, introduced in Section 3, proves to be useful. With further implications in terms of risk bands, the risk graph of $\sqsubseteq$ represents a detailed ordering of security risks posed by different permitted role–task combinations.

The third hierarchical relation on roles, introduced in this work, takes into account the risks described above. It is a hierarchical partial order and is denoted by $\lll$ . It extends the relation $\leqslant$ in (8) by incorporating $\sqsubseteq$ and, following [4], is defined as

$$r_1 \lll r_2 \Leftrightarrow (r_1 < r_2) \wedge (\forall\, t \in \mathit{TASK} \bullet t \notin \mathit{Permissions}(r_1) \wedge t \in \mathit{Permissions}(r_2) \Rightarrow (r_1, t) \preccurlyeq (r_2, t)) \tag{9}$$

According to this principle, role $r_2$ is senior to $r_1$, i.e., $r_1 \lll r_2$, if and only if the role $r_2$ is senior to the $r_1$ in the sense of $\leqslant$ in (8), i.e., $r_1 < r_2$, and all permissions, which are not included in the junior role $r_1$ but are in $r_2$, are handled more securely by $r_2$ than by $r_1$ with respect to the relevant risk graph. The intention is to ensure that senior roles, while inheriting permissions of the respective junior roles, are entrusted with certain permissions requiring greater degree of security. This is a justification for a security–orientated notion of a hierarchical seniority. However, this does not necessarily mean

that the senior role can handle all its permissions more securely than the junior role. In fact, it may be the case that the junior role is intended to handle its own tasks, perhaps with the exception of its own inherited ones, more securely than the senior role because of, for example, the specialist expertise required by the tasks concerned.

### 4.3 Principle II: Delegation of Tasks

Our approach to delegation of access rights is based on certain rules that take security risks into consideration. The lack of such explicitly stated rules in other works may be due to the informality of the way delegation is handled normally or the excessive number of possibilities in delegation encountered in practical situations. Note that delegation applies only to *level 1 delegation* [3], that is, to roles initially assigned by the system administrator and not to those gained by previous delegations from other roles.

**Principle II(a): Lateral Delegation of Tasks.** The lateral delegation here concerns the delegation of roles at the same level of seniority as understood by the relation $\simeq$, introduced in Section 4.1. This may be expressed as

$$\forall r_1, r_2 \in ROLE \bullet r_1 \neq r_2 \wedge r_2 \in lat\_del\_roles(r_1) \Rightarrow r_1 \simeq r_2 \tag{10}$$

**Principle II(b): Downward Delegation of Tasks.** This principle deals with the delegation of its access rights by one role to another in a strictly lower level in the hierarchy $\ll$; see Section 4.2. Let us deal here only with the total delegation, i.e., the delegation of all access rights of the delegator role [3]. In order for such a delegation to be permitted, the two conditions (11) and (13) are to be satisfied.

Firstly, the delegating and delegatee roles must be hierarchically related, as in

$$\forall r_1, r_2 \in ROLE \bullet r_1 \neq r_2 \wedge r_2 \in down\_del\_roles(r_1) \Rightarrow r_1 \ll r_2 \tag{11}$$

Secondly, security risk considerations need to be taken into account. To minimise security risks, the access rights are better be delegated to the role(s) that would present the least risk when they perform the delegated tasks. This can be established using the risk graph, introduced in Section 3. Considering each of the tasks to be delegated under the delegating role, it is possible to calculate the worst (lowest, taking the sign into account) risk distance from the delegating role to each candidate delegatee role. Thus, for each potential pair of delegating-delegatee roles there is to be a lowest risk distance. The delegatee role giving the largest of these risk distances (taking the sign into account) would be the one to be favoured for delegation. With this in mind, let us first define the worst risk distance between the permissions of one role $r_1$ relative to the same permissions under another role $r_2$

$$\begin{gathered}\forall r_1, r_2 \bullet r_2 \ll r_1 \Rightarrow worst\_risk\_dist(r_1, r_2) = \\ min\{RD((r_1, t), (r_2, t)) \mid t \in permissions(r_1) \wedge t \notin permissions(r_2)\}\end{gathered} \tag{12}$$

where *min S* gives the minimum value in the set $S$ (of integers). The role(s), which is the least risky for delegating $r_1$'s permissions, is a role $r_2$ having the largest among the worst risk distances calculated as described above. In other words, for delegating $r_1$ the least risky delegatee role is $r_2$, provided that

$$\forall r_3 \bullet r_3 \ll r_1 \Rightarrow worst\_risk\_dist(r_1, r_2) \geqslant worst\_risk\_dist(r_1, r_3) \tag{13}$$

**Principle II(c): Avoidance of Conflicts of Interest.** Furthermore, neither of the above forms of delegation should result in any static conflict of interest with other delegated roles and the target (delegatee) role ($r$ below) itself. That is, delegation must respect the static separation of duty. This principle may be expressed as

$$\forall r \in ROLE \bullet \exists roles \in \mathbb{P}\, ROLE \bullet roles = delegated\_roles(r) \cup \{r\} \Rightarrow roles \times roles \cap SSD = \emptyset \tag{14}$$

## 5 Case Study: A Health Care Information System

This section illustrates the proposed approach using a hypothetical, but realistic, simple access control system applicable to a hospital environment, but in relation to: a) the construction of a role hierarchy (Principle I), and b) downward delegation (Principle IIb), both based on security considerations. A description of the functional requirements of the access control system are summarised, along with the notation, in Table 1.

**Table 1.** The tasks defined in the hospital's information system

| Task Name | Representation | Brief Description | Authorised Roles† |
|---|---|---|---|
| $t_1$ | (lead,op) | leading an operation | consultant (c) |
| $t_2$ | (asst,op) | assist in performing an operation | consultant (c)<br>surgeon (s) |
| $t_3$ | (prep,pat) | pre-operation care for a patient | nurse (n) |
| $t_4$ | (mont,pat) | post operation monitoring of patient | nurse (n) |
| $t_5$ | (adm-med,pat) | administering medication to patient | nurse (n) |
| $t_6$ | (adm-aneas,pat) | administering anaesthetics to patient | anaesthetist (a). |

† Note: Shown in brackets is the notation to be used later.

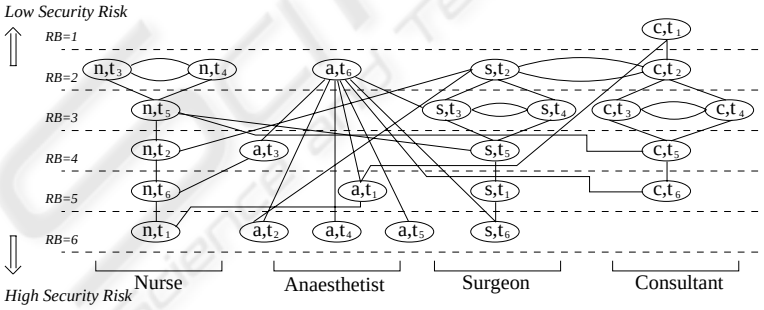


**Fig. 2.** Risk graph of the permissions in the hospital's information system.

A security risk assessment, involving the permissions and the roles concerned, has resulted in a security risk ordering relation shown in Figure 2. Though its primary purpose is to present relative security risk levels between various pairs of permissions, belonging also to different roles, it also indicates the risk graphs of individual roles. Arcs on the graphs, assumed to run upwards, show the risk–comparability between permissions.

Let us first consider three possible role hierarchies in relation to both the Principle I and the functional requirements. These are shown in Figure 3 along with the

permissions associated with each role. The three role hierarchies can be checked for conformity with (9) against the security risk graph shown in Figure 2. By the manner of their construction, all three hierarchies satisfy the relation $\leqslant$ in (8) – the first conjunct of (9). Hierarchy 1 satisfies also the second conjunct. In this case, note that $Permissions(s) \subset Permissions(c)$, $t_1 \in Permissions(c)$ but $t_1 \notin Permissions(s)$ and, according to the risk graph, $(s, t_1) \preccurlyeq (c, t_1)$. Analogous arguments apply to pairs of roles *s* and *n*, and *a* and *n*. It may be noted that Hierarchy 1 also satisfies the functional requirements. Following a similar analysis, we note that Hierarchy 3 conforms with Principle I, but violates the functional requirements. In Hierarchy 2, however, in relation to the pair $n \leqslant s$ (by transitivity of $\leqslant$), $t_6 \in Permissions(s)$ and $t_6 \notin Permissions(n)$ but $(s, t_6) \preccurlyeq (n, t_6)$, which violates Principle I. Thus, we conclude that only Hierarchy 1 satisfies both the functional requirements and the security considerations expressed in Principle I, thus justifying its applicability to RBAC as proposed here.

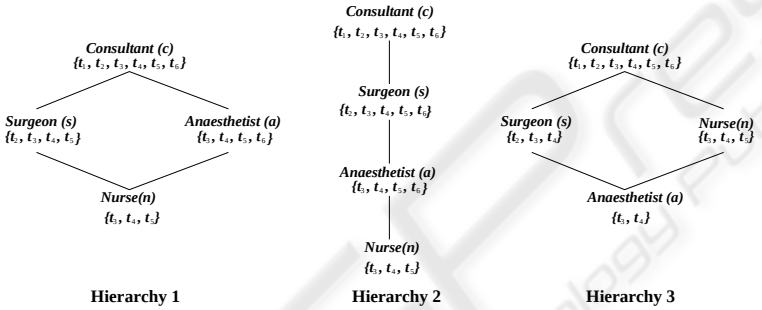


**Fig. 3.** Three possible role hierarchies

Turning attention to delegation, let us consider a situation where a subject exercising the role *consultant* wishes to delegate his role (i.e. the totality of the tasks) to a junior role in Hierarchy 1. In determining the most secure role(s) to whom the delegation should take place, the risk distances between the permissions of *consultant* and the same performed by the other roles need to be calculated. Note, however, that some of *consultant*'s tasks are shared also by the junior roles. Therefore, risk distances are needed only in relation to the non–shared tasks. The tasks concerned are: $t_1$ and $t_6$ in the case of delegation to *surgeon*; $t_1$ and $t_2$, in the case of *anaesthetist*; and $t_1$, $t_2$ and $t_6$ in the case of *nurse*. Therefore, according to (12), the worst risk distances are, respectively, -4, -4 and -5, leading to the least risky roles *anaesthetist* and *surgeon* for delegating *consultant*'s role.

## 6 Conclusion

This paper presents a rigorous formal approach for dealing with some of the key issues in RBAC, in particular, delegation and allocation of access rights. Assignment of access rights is a critical and an error-prone process. Therefore, precise, clear and well-studied guidelines are essential for combating security breaches resulting from unauthorised access rights. An important contribution of the proposed approach, in this respect, is the formulation of several principles for defining role hierarchies and handling role

delegation based on a novel idea of a security risk ordering relation. The approach also incorporates precise ways to consider other factors, such as functional requirements and conflicts of interest, etc., essential for assuring the system integrity. The risk ordering relation relies on a detailed assessment of the risks faced by the system. In the event of lack of sufficient information, the approach enforces certain default interpretations of risk in a conservative manner, so that any disagreement leads to a refinement of the security risk analysis. A case study drawn from health care domain illustrates the approach and demonstrates its effectiveness.

## References


1. American National Standard for Information Technology. *Role Based Access Control.* Draft BSR INCITS 359, April 2003.
2. Barka E. and Sandhu R. *A Role-Based Delegation Model and Some Extensions.* Proceedings of the 23rd NIST-NCSC National Information Systems Security Conference, pp: 101–114, Baltimore, USA, October, 2000.
3. Barka E. and Sandhu R. *Framework for Role-Based Delegation Models.* Proceedings of the 16th IEEE Annual Computer Security Applications Conference, pp: 168–175, New Orleans, Louisiana, USA, December, 2000.
4. Dammag H. and Nissanke N. *A Mathematical Framework for Safecharts.* Proceedings of the 5th International Conference of Formal Engineering Methods, pp: 620–640, Singapore, Singapore, November, 2003.
5. Ferraiolo D. Cugini J., and Kuhn R. *Role-Based Access Control (RBAC): Features and Motivations.* Proceedings of the 11th Annual Computer Security Applications Conference, pp: 241–248, New Orleans, LA, USA, December, 1995.
6. Ferraiolo D., Sandhu R., Gavrila S., Kuhn R. and Chandramouli R. "Proposed NIST Standard for Role-Based Access Control". *ACM Transactions on Information and System Security (TISSEC)*, Vol. 4, No. 3, August 2001, pp: 224–474.
7. Khayat E. and Abdallah A. *A Formal Model for Flat Role-Based Access Control.* Proceedings of the ACS/IEEE Conference on Computer Systems Applications, Tunis, Tunisia, July, 2003.
8. Na S. and Cheon S. *Role Delegation in Role-Based Access Control.* Proceedings of the 5th ACM workshop on Role-Based Access Control, pp: 39–44, Berlin, Germany, June, 2000.
9. Sandhu R., Coyne E., Feinstein H. and Youman C. "Role-Based Access Control Models". *IEEE Computer*, Vol. 29, No. 2, November 1996, pp: 38–47.
10. Sandhu R., Ferraiolo D. and Kuhn R. *The NIST Model for Role-Based Access Control: Towards A Unified Standard.* Proceedings of 5th ACM Workshop on Role-Based Access Control, pp: 47–64, Berlin, Germany, July, 2000.
11. Zhang L., Ahn. G.J. and Chu B.T. "A Rule-Based Framework for Role-Based Delegation and Revocation". *ACM Transactions on Information and System Security*, Vol. 6, No. 3, August 2003, pp: 404–441.
12. Zhang L., Ahn. G.J. and Chu B.T. *A Role-Based Delegation Framework for Healthcare Information Systems.* Proceedings of the 7th ACM symposium on Access Control Models and Technologies, pp: 125–134, Monterey, California, USA, June, 2003.
13. Zhang X., Oh S. and Sandhu R. *PBDM: A Flexible Delegation Model in RBAC.* Proceedings of the 8th ACM symposium on Access Control Models and Technologies, pp: 149–157, Como, Italy, June, 2003.