Partial Access Control Permissions and Rights

Leigh H. de la Motte and Jacky Hartnett
School of Computing
University of Tasmania
Locked Bag 1359, Launceston 7250, Tasmania, Australia
lhdela@utas.edu.au

Abstract

In order to satisfy the Principle of Least Privilege[1] in large enterprises which employ Role Based Access Control systems a large number of roles must be defined. Role management can become a demanding and complex task in such situations. This paper introduces the concepts of *Partial Access Control Permissions* (Partial Permissions) and *Partial Access Control Rights* (Partial Rights) which enable the number of roles to be reduced and role management burdens to be eased.

Partial permissions are linked permissions which are applied simultaneously to two or more roles. The rights defined in a partial permission only become active when an access request triggers a sufficient number of linked partial permissions. Partial permissions enable permissions to be given to any combination of roles. For example, if a hospital patient is attended by clinicians with a "treating team" role and the hospital has a "doctor" role, a partial permission applied to the two roles is only triggered during an access request from a doctor who is on the treating team.

Similarly, a *Full Right* is triggered when a complete set of Partial Rights are activated. Partial rights provide a means for incorporating consent and authorisation into the access control system, as well as facilitating the application of general access control rules to groups of associated roles.

Keywords: Access Control, Role Management, Access Rights.

1 Introduction

This paper describes the concepts of Partial Permissions and Partial Rights. The use of these concepts can simplify the administration of current access control systems while at the same time increasing the level of control that can be achieved.

In short, Partial Permissions reduce the number of roles/groups that must be defined in a domain. They do this by enabling permissions to be assigned to role/group intersections and subtractions in addition to single roles/groups.

Current access rights, such as READ, WRITE and MODIFY are binary in nature in that a user either has the right or they don't have the right. Partial Rights enable access rights to be given to users subject to specified situations. This allows much greater flexibility in privilege allocation as well as allowing concepts such as consent and authorisation to be modelled directly as rights. By modelling these concepts as rights a mechanism is provided for their incorporation into a system as a part of normal practice rather than as system add-ons.

The long term goal of the research focuses on the development of an access control model based on set theory, called Set Based Access Control, which defines subjects, rights, and objects groups as sets. Partial Permissions and Partial Rights are the central mechanisms which facilitate the incorporation of Set Theory into an access control model. However, as the use of Partial Permissions and Partial Rights may have broader application than the research will consider, it is appropriate that the description of these concepts be dealt with in this concise paper.

1.1 Concept of Roles and Benefits

Role Based Access Control (RBAC), introduced by Ferraiolo & Kuhn (1992) is a well established access control model. By allowing administrators to group privileges together in role abstractions and to allocate roles to users, the administrative burdens imposed by previous Mandatory Access Control (MAC) models is reduced. The concept of assigning users to organisational roles is one that is intuitive, which also increases the administrative advantages.

Work in the area of trust management (Li & Mitchell 2003) has shown that the possession of an organisation role can be used to facilitate authorisations for accesses to systems of other organisations which recognise the role. Roles therefore can be used to enable remote authorisations, potentially up to a global scale.

1.2 Role Management and Granularity

While the possibilities of utilising roles on a large scale are attractive, applying RBAC to large organisations has already proved to be difficult. Kern & Walhorn (2005)

[1] Definitions of the main terms used in this paper can be found in Section 3. The terms *permission* and *privilege* are used interchangeably and the term *right* could be interpreted as meaning an *access right* or an *access type*.

give some insights into role management in a number of large organisations which vary from having 150000 users with 50 roles at one extreme to 11500 users with 2800 roles at the other extreme. Role management in such organisations is a complex task. If roles are to be used on a more global scale, the task becomes even more complex.

To alleviate the administrative management burden in large organisations a number of role management mechanisms have been proposed. Ferraiolo et al. (2003) describe a number of techniques including the Enterprise RBAC model. Kern & Walhorn (2005) use rules to automate role allocation. This approach stems from the Rule Based RBAC (RB-RBAC) model proposed by Al-Kahtani & Sandhu (2002).

Current versions of both Windows and MacIntosh operating systems both utilise hierarchical group structures. Groups in these systems are very similar to, if not identical to, roles.

Many access control models, including the Windows model, manage access control by the delegation of administrative privileges. While these privileges may be given directly to individuals or to the roles/groups that they belong to, there is a tendency for the allocation of the privileges to become hard to manage and track.

Another level of complexity is instituted in team-based access control models (Thomas 1997) (Georgiadis et al. 2001) when additional team-specific roles are defined. Alotaiby & Chen (2004) developed a team based model which utilised organisational roles. They advocate that this approach is better able to model real world organisational needs.

A drawback of using roles is that the desire to ease administrative burdens can lead to the granting of roles which are applied too generally to users. For example, in a hospital system, doctors may be given access to the records of all patients when they are allocated the role of doctor. This single role makes management easier, but the Principle of Least Privilege, which states that access privileges should only be given if they are needed, is violated. In order to obtain access control granularity which satisfies the Principle of Least Privilege RBAC requires more specific roles. These specific roles tend to be more transitory and dynamic in nature which increases the administrative burden significantly.

1.3 Set Theory

Set Theory is an area of mathematics which has been around for well over a century. While a set is simply a collection of things of a particular kind, a set can also be defined by a rule or predicate which all elements must satisfy.

This paper proposes that roles/groups should be treated as sets of users who satisfy a certain rule. In the same way, system objects are also described with sets. Rules of Set Theory can then be used to describe interactions between associated roles and associated object groups.

Set theory has been used as the basis for a number of computing mechanisms. For example, Dovier et al (2000) incorporate set theory in their studies into handling constraints in logic programming. In the area of access control, Chen & Sandhu (1996) use set theory notation to describe their RBAC model. Li & Mitchell (2002) discuss the usefulness of using the intersection of roles. They also define new operators to facilitate authorisations from multiple individuals in different roles. Mechanisms like this provide a means to easily solve problems such as separation of duties where different users with defined roles are required to authorise actions.

If sets A and B are considered, there are three binary set operations that are of interest (see Figure 1). They are the *union* of A and B (A ∪ B), which contains all the elements present in both A and B; the *intersection* of A and B (A ∩ B), which contains all the elements of A that are also present in B, and the *relative compliment* of B in A (A – B), which contains all the elements of A which are not present in B.

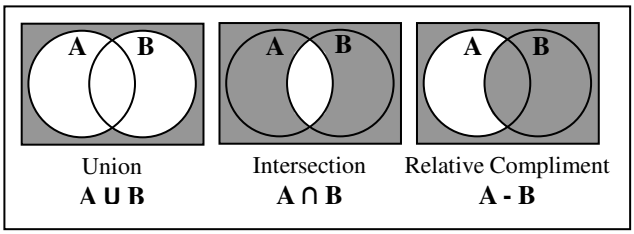

Figure 1: Binary Set Operations

1.4 Efficient Storage and Processing

Many attributes, such as name or address, are relatively unique and not relevant to access control decisions. These attributes can be described as *independent attributes*. There are other attributes, however, such as role or location, which are generally collective in nature and are relevant to access control decisions. These attributes can be described as *dependent attributes*.

It is common to store attributes associated with users in some form of user profile. This type of storage could be termed *user-based storage*. Another approach is to list users who have a particular attribute in some abstraction of the attribute. This type of storage could be termed *attribute-based storage*. For example, on the one hand, all users who are doctors could have an attribute in their user profile which indicates that they hold the role of doctor (user-based storage) or a list of all doctors could be stored in the system (attribute-based storage).

The advantage of user-based storage is that it is easy to retrieve all the attributes of individual users. The advantage of attribute-based storage is that all users who have a particular attribute can easily be found.

In order to efficiently determine which dependent attributes, such as roles or group memberships, a user holds, it is advisable to employ user-base storage. In contrast, to make use of set operations it is more efficient to use attribute-based storage for dependent attributes. A set can be used to group users who hold particular dependent attributes. For example, sets can be used to

hold users who hold a particular role. They can also be used to hold other information, such as the users who work in a particular location or the users are responsible for a particular client.

Where memory is cheap and speed of processing is paramount it is advantageous to incorporate a *dual-storage* method which uses both user-based and attribute-based storage. This enables efficient establishment of role/group membership and the use of role/group intersections and *subtractions* (relative compliments). The end result of employing a dual storage approach is that it allows privileges to be processed in an efficient manner when Partial Permissions and Partial Rights are utilised.

1.5 Outline of contents

The next section describes aspects of set theory that are relevant to the access control mechanisms proposed in this paper. Section 3 contains definitions that are used in the following two sections which explain Partial Permissions and Partial rights. These sections describe the details of the access control mechanisms.

Examples relating to a hospital access control system are employed to relate the use of these mechanisms to a real situation. The dynamic, volatile and complex nature of hospital access control systems makes them a good test domain for the application of access control models.

Section 6 of the paper discusses the uses of the proposed mechanisms. Finally, Section 7 draws conclusions and outlines further work.

2 Set Interactions

This paper proposes that roles or groups can be modelled as sets and that set operations can be used to both increase the granularity of privilege assignment and to reduce the number of roles that need to be defined.

2.1 Access Requests

To determine whether or not an access should be allowed, an access control system must check to see if the user making the request is entitled to the type of access required (defined by an access right) with regard to the object to which access is sought. In other words, the system checks a (subject, right, object) *access triple* to determine if the subject holds the required right on the object.

To maximise the use of set operations they can be applied to all three components of access triples. That is, subjects can be grouped into sets, rights can be described by sets, and objects can be grouped into sets. While it is ideal to apply set operations to all three components, it is also possible to apply them just to one or two of the components. This may be necessary if an implementation must be designed to fit into an existing framework such as RBAC.

2.2 Subject Sets

Subjects can be grouped into sets and set operations can be used to allow privileges to be given to set unions, intersections and relative compliments. In this sense the sets represent roles. Figures 2-4 show examples of how set operations can be applied to subjects.

Inheritance of roles is modelled by defining subsets. For example, in Figure 3 a "Surgeon" set could be a subset of the "Doctor" set, allowing "Surgeons on the Treating Team" to be represented. As Surgeons are also Doctors, they would also be represented in "Doctors on the Treating Team". Non-Surgical Doctors could be represented by set subtraction.

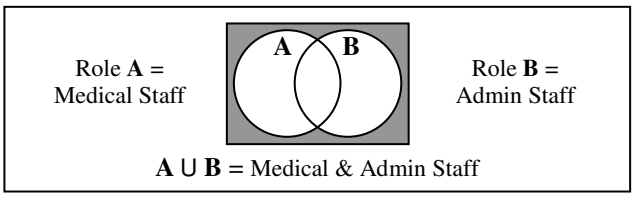

Figure 2: Subject/Role Union

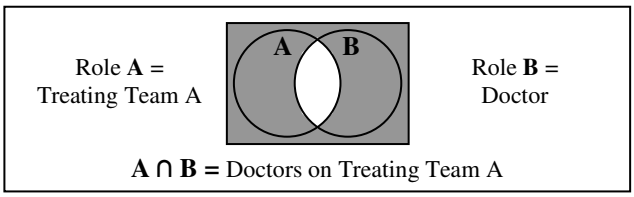

Figure 3: Subject/Role Intersection

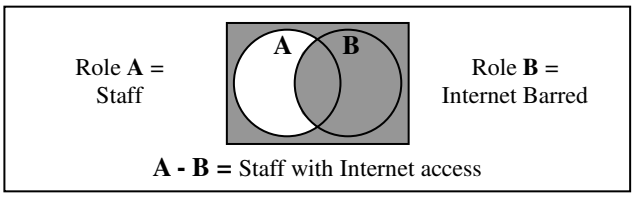

Figure 4: Subject/Role Subtraction

Figure 4 shows an example where a set is used to stop a certain type of access – internet access in this case. Placing staff in this category could be a punishment for system misuse or conversely it may be required that certain roles are not allowed to have internet access to certain types of records.

2.3 Right Sets

In a system there are a finite number of rights which may appear in access triples. For each of these rights a set of rights which satisfy the access requirements can be defined.

If, for example, possession of the rights MODIFY (M) and APPEND (A) equate to the possession of the right WRITE (W) then the following *rights set* can be defined for WRITE:

$$\text{WRITE} = \{M \cap A, W\}$$

This means that when processing a request for WRITE access to an object, the system can grant access if the user possesses either both the *M* and *A* rights or the single *W*

right. Similarly, the rights set for each of the other possible rights can be defined.

2.4 Object Sets

Objects can be grouped into sets and set operations can be used to allow privileges to be given to set unions, intersections and relative compliments. Objects sets represent object groupings such as directories or folders, but an object may belong to multiple object sets.

Figures 5-7 show examples of how set operations can be applied to objects.

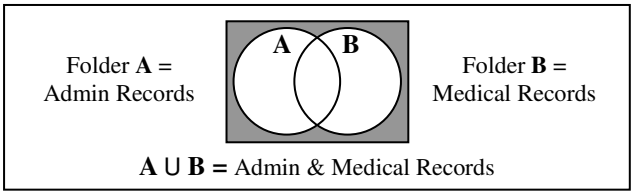

Figure 5: Object Group Union

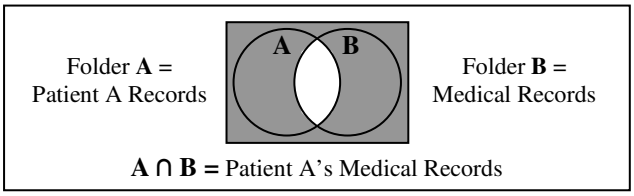

Figure 6: Object Group Intersection

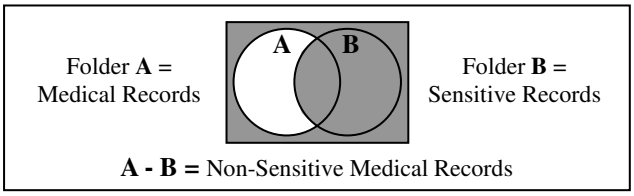

Figure 7: Object Group Subtraction

As is normally the case, folders can be subsets of other folders. For example, in Figure 6 "Pharmaceutical Records" would be a subset of "Medical Records".

Subsets can also be used to define different levels of record sensitivity or security classification. For example, in Figure 7 a number of subsets of sensitive records could be defined such as "Very Sensitive" or "Personal". It follows that Set-Based Access Control can incorporate *Access Levels*, as described in the Bell-La Padula model (Bell & LaPadula 1976).

3 Definitions

This section details the definitions that are used in this paper. All the definitions are new, with the exception of the definitions for Access Type and Access Right, which are from existing glossary sources.

Partial Permissions (PPs) - Privileges that exist in sets of two or more. In accordance with the following definitions, Partial Privileges can be represented in the form: (PID,PAN,AQ,Right/ PR).

partial Permission set ID (PID) - An identifier associated with each Partial Privilege set.

partial Permission Activation Number (PAN) – An integer which shows the number of Partial Privileges required to activate the privilege.

Access Qualifier (AQ) - Specifies how the system treats an access right (for example, permit (p), deny (d), audit, alert etc). More complex qualifiers could be defined. eg alarm = deny + audit + alert.

Access Type - The nature of an access right to a particular object or object group (for example, read, write, execute, append, modify, delete, or create). Derived from (CSIS).

Access Right (Right/Full Right) - A granted permission/ privilege for a Subject to carry out an Access Type. (CSIS) In this paper Rights are denoted by uppercase letters. eg READ (R).

Partial Rights (PRs) - Rights that exist in sets of two or more with rules that specify how they constitute a Full Right. In this paper Partial Rights are denoted by lower-case letters. eg consent (c) + write(w) = WRITE (W).

Figure 8 shows the diagrammatic representation of Partial Permissions used in this paper. The letters *x* and *y* represent the PIDs, *2* and *3* are the PANs, *p* and *d* are the AQs, and *W* and *c* are examples of a Right and a Partial Right respectively.

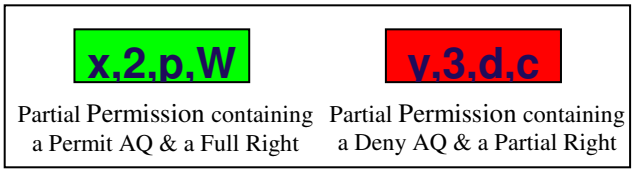

Figure 8: Partial Permission Diagrammatic Representation

4 Partial Permissions

The concept of Partial Permissions is basically to split up a normal permission into parts and apply the parts to separate roles/groups. If an access request then brings all the required parts together the permission is applied in the normal way. The purpose of doing this is to reduce the need to create extra roles/groups as well as to provide greater flexibility in the application of permissions.

Partial Permissions used in combination with Partial Rights can be used to define general access control rules for a system. These *general permissions* are discussed in Section 5. For simplicity, this section deals with *specific permissions* which utilise Partial Permission without Partial Rights. The processing of Partial Permissions is the same regardless for whether or not Partial Rights are present.

4.1 Assigning Permissions to Sets

An access control mechanism which utilises set operations can apply privileges to sets in the same way that privileges are assigned to roles. In other words, each set would have zero or more privileges assigned to it. Now, in addition to applying privileges to a single set, the aim is to enable privileges to be applied to the unions, intersections, or relative compliments of sets.

In the case of union, it is obvious that applying a privilege to the union of two sets is effectively the same as applying the privilege to the two sets individually. In the case of intersection, a privilege must be applied to both sets, but must be activated only when the user/object is an element in both sets. In the case of set subtraction, a privilege again must be applied to both sets, although one will be a positive (*permit*) privilege and the other will be a negative (*deny*) privilege. Table 1 details the different privilege allocation requirements.

Operation	Method of Allocation
A ∪ B	Apply a permit privilege to A & to B individually
A ∩ B	Apply a permit privilege to A & an associated permit privilege to B
A - B	Apply a permit privilege to A & an associated deny privilege to B

Table 1: Privilege Allocation

The concept of associating privileges is a new concept. Associations are required so that the privileges are only activated when the specified combination of sets are present, not every time one of the sets is present.

So far only binary set operations have been described. Set Theory shows that binary set operations can be extended to form operations which apply to multiple sets. With multiple sets, to activate a normal *full permission* a certain number of associated *partial permissions* must be present. In other words, a full permission can be made up of a bag of partial permissions. The partial permissions in a bag may be identical in cases of set intersection or opposing in cases of set subtraction.

4.2 Specific Permissions

Partial Permissions which contain only full rights are used to define specific access requirements. For example, if Doctors on Patient A's Treating Team are to be given WRITE access to Patient A's Medical Records, then an identical partial permission containing the W right is assigned to each of the four sets involved (see Figure 9).

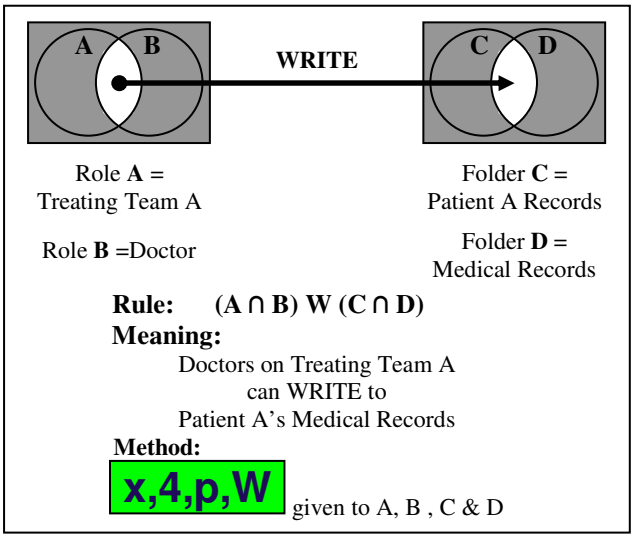

Figure 9: Specific Permissions Example #1

The example shown in Figures 9 utilises set intersection while that in Figure 10 shows set subtraction as well.

In figure 10 the "deny" privilege given to set B is specific to the permission in question. This means that members of set B do not get "WRITE" access to the records by this set of Partial Permissions. It is still possible that they may gain access to the records through some other permission. This specific denial is enabled due to the fact that the PID (x) of both Partial Permissions is identical. If a more general denial were required then the "deny" privilege could be specified separately. This example illustrates the fine level of control that can be achieved through the use of Partial Permissions.

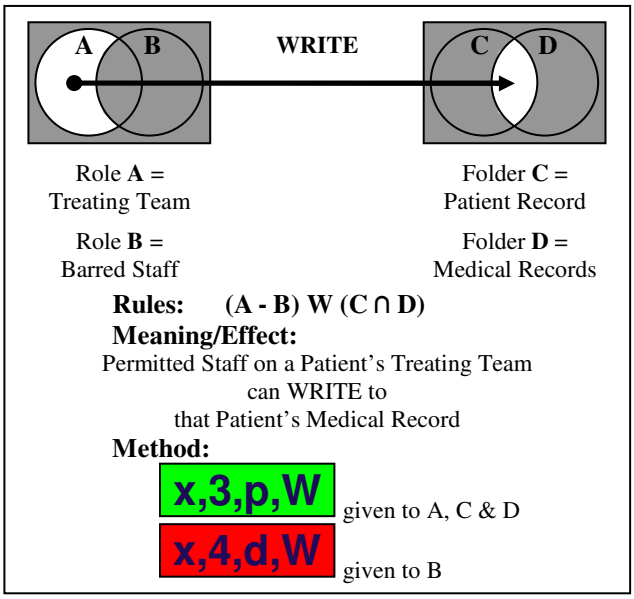

Figure 10: Specific Permissions Example #2

4.3 Ability to Create Specific Override Rules

The following section describes how general permissions can be created. General permissions are broad in their application and do not just refer to single specific permissions. In practical situations it is advisable to use general permissions to facilitate accesses which are standard in nature while employing specific permissions to override or compliment the general permissions.

4.4 Processing Partial Permissions

Partial Permissions are assigned to sets in the same way that normal permissions are assigned to roles/groups. They are also processed in the same way. For the sake of brevity, only an overview of the processing mechanism is given here.

When an access request is processed, all the permissions and Partial Permissions associated with any of the subject's sets or the object's sets are collected. Each Partial Permission with the same PID is counted to see if the number present meets the total required by the PAN contained in each Partial Permission. If the number reaches the PAN then the Partial Permission becomes a normal *Full Permission*. If not, they are ignored.

The speed of processing is proportional to the total number of permissions and Partial Permissions present. As there is no searching required there is no excessive overhead. When compared to access control methods which require that contexts or constraints have to be checked, Partial Permissions may indeed be faster to process. When compared to rule-based solutions, there are no complex functions to be evaluated, so again Partial Permissions would normally be faster.

5 Partial Rights

Access rights or Full Rights are the READ, WRITE, APPEND… rights that are in common use. These rights are absolute in their nature. A subject either has the right or does not have the right. They are generally given once and thereafter always apply.

In contrast, the concept of Partial Rights (PRs - see definition in Section 3) implies that rights are not absolute. PRs allow rights to be given subject to a number of constraints being fulfilled. Single PRs can be given to individuals or groups/roles. Each PR does not, by itself, allow any access. In the context of PRs, Full Rights can be thought of as consisting of one or more sets of partial rights. For example, the Full Right to "READ" may be activated by the possession of both the "consent" and the "read" Partial Rights. The feature of interest is that each of the Partial Rights can be associated with different groups/roles and that Full Rights are only activated when the subject is a member of the all the required groups/roles.

While a Full Right can be attained by the possession of all the required PRs, Full Rights can still be granted directly in the normal way. Partial Rights are an extension of the normal mechanism which allow more flexible and meaningful access requirements to be specified. They do not replace normal Full Rights, they merely supplement them.

The following example shows how Partial and Full Rights work together. The Partial Rights consent (c), read (r), modify (m), and append (a) can be defined, along with the rights READ (R), MODIFY (M), APPEND (A), and WRITE (W). If (i) consent is required to READ and APPEND, (ii) READ is required to MODIFY, and (iii) MODIFY and APPEND are required to WRITE, then the following right sets can be defined:

READ = {c∩r, R, M, W},

MODIFY = {R∩m, M, W},

APPEND = {c∩a, A, W}, and

WRITE = {M∩A, W}.

These sets imply, for example, that a user who wishes to READ an object must possess either both the *c* and *r* Partial Rights or one of the *R*, M or W Full Rights on the object. They also imply that a user with WRITE access to an object also has READ, MODIFY and APPEND rights to the object, as "W" appears in each rights set.

5.1 Consent and Authorisation Rights

Partial Rights allow abstract concepts such as consent and authorisation to be built into a system at the base level of the systems rights. For example, consent can be represented as a Partial Right. This allows general consent permissions to be given to subjects which are only activated when the subjects also possess other concrete (read, write…) permissions. This approach to consent works because consent can be given broadly, say to all clinicians in a hospital who may look after a patient, without access being directly given to all clinicians.

Authorisation is another key concept which can be modelled through Partial Rights. Many business processes require that users be authorised by one or more other users before making an access or carrying out a task. Partial Rights provide a mechanism for authorisations to be sought and given to the system. They may be sought or given at any time – either before, at the time of, or after they are needed or requested. Once a user possesses the required authorisations within the system, access is allowed.

5.2 General Permissions

In order to alleviate the need to define specific access requirements when they are needed, Partial Rights are used. Two or more sets of Partial Permissions containing Partial Rights are applied independently. They then interact to form a general rule. Figure 11 shows an example of how a general permission can be applied.

In Figure 11, there are two sets of Partial Permissions. The first set defines the permission for doctors to write to medical records and only needs to be applied once to the "Doctor" and ""Medical Record" sets. The second set defines that the patient has consented for members of their individual treating team to access their records. Each time a patient is admitted to the hospital consent is obtained (it may be implied) for their carers as a whole to have access to their records.

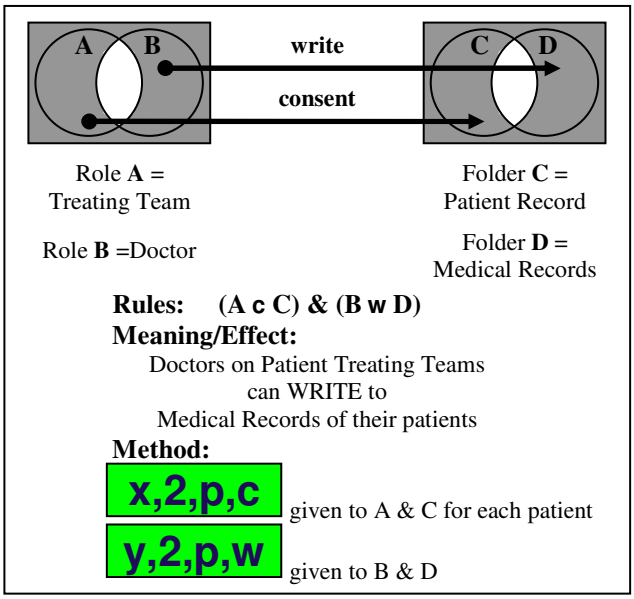

Figure 11: General Permissions

The effect of the two sets of Partial Permissions is firstly, to restrict access by clinicians to patients in their care, and secondly, to restrict access to the patients' records according to the clinician's organisational role(s). For example, doctors can write medical records only, while administrators can write to admin records only.

While the second set of Partial Permissions need to be applied for each patient they can be applied automatically to sets that are based on a template. Standard Partial Permissions can therefore be automatically assigned to a default treating team which is created for the patient on their admission to the hospital.

6 Use of Partial Permissions and Rights

While Partial Permissions can be used together with Partial Rights, the two concepts are also independent. Partial Permissions are used to enable permissions to be assigned to subject and/or object group intersections and subtractions, simplifying role/group management. Partial Rights, on the other hand, are used to enable rights to be conferred in flexible ways which more closely model real-world procedures for things such as consent and authorisation.

6.1 Partial Permissions Reduce Role Numbers

In order to meet security needs in RBAC based models it is usually necessary to define fine grained roles which meet specific requirements. Where role intersections could be used these models need to either define a new role to represent each role intersection, or else define a number of finer grained roles to cover the various requirements.

Partial Permissions, which enable the use of set intersection and subtraction operations, reduce the need for these additional roles to be created. As fewer roles are necessary, role management overheads are reduced.

6.2 Granularity of Access Control

Many computer security experts point out that the users of a system pose a significant security threat. For example Schneier (2000) states that "People often represent the weakest link in the security chain and are chronically responsible for the failure of security systems". The need to develop systems that meet the Principle of Least Privilege is therefore great.

From a security point of view both the specific and general permission mechanisms meet the requirements of the Principle of Least Privilege. They therefore provide the highest level of protection possible. The level of security attained practically however, will also depend on the efficiency of the group management mechanism.

6.3 Easy Management with General Rules

General permissions can be set up very easily and require no ongoing administration except the management of team memberships. Team management can be highly automated and utilise workflow operations, but that subject is beyond the scope of this paper.

The concepts of set intersection and subtraction are easy to understand. For example, the idea of "Doctors on a Treating Team" is conceptually simple. It is therefore possible to build a system which utilises these characteristics in a way where administrators can easily define the rules that are required.

6.4 Consent and Authorisation Mechanisms

The ability to quantify concepts such as consent and authorisation by building them the heart of the access control system is very important. By creating Partial Rights which model the concepts, and having these rights resident in the system's set of rights, the concepts can no longer be thought of as mere add-ons to the system.

It is common for there to be much talk about consent, in particular, but there are few, if any, examples of it being incorporated as a standard system component. While mechanisms exist for system administrator and their surrogates to facilitate authorisations there are few systems which allow authorisations to be made without the need to pass administrative privileges to general system users. A notable exception to this is the Li & Mitchell's Role-Based Trust Management Framework (Li & Mitchell 2003).

6.5 Processing Partial Rights

While there are overheads incurred in the processing of Partial Rights, these are not significant. When Partial Rights are present, it is merely just a matter of checking to see whether a combination of the Partial Rights present constitute a Full Right. This can be done by checking the relevant Rights Set for the particular Right that the access request contains.

6.6 Incorporation into Existing Systems

Partial Permissions and Partial Rights do not invalidate existing permissions and access rights, they merely add levels of flexibility and control. As such, they can be added to existing systems without making them redundant. Partial Permissions can easily be applied to roles in RBAC systems. Partial Rights can supplement existing system rights with the addition of a simple partial rights processing facility.

While it is feasible that Partial Rights and Permissions may be useful at the operating system level, they don't necessarily have to be utilised at that level. Systems such as Oracle DBMS build in access control mechanisms at the application level. There is no reason why Partial Permissions and Partial Privileges cannot also be utilised at the application level.

6.7 Scope of Set Based Access Control

The scope of Set Based Access Control is somewhat wider than that of Role Based Access Control as it envisages the organisation of objects and not just users and privileges. Even though this is the case, Partial Permissions and Partial Rights can still fit into the RBAC framework. In other words, Partial Permissions and Partial Rights can be applied to subjects and rights

through roles without the need to utilise set operations on objects.

7 Conclusion and Further Work

Partial Permissions provide a means for reducing the number of roles or groups necessary in a domain. They achieve this by allowing permissions to be applied to role/group intersections and subtractions in addition to single roles/groups. The concepts of these two set operations are intuitive, which leads to the easy formulation of required access control rules in practical implementations.

Partial Rights can be used with or without Partial Permissions. They increase the flexibility with which rights can be applied. Instead of rights being binary in nature, in that they are either present or absent, Partial Rights allow rights to be activated when specified role/group memberships are present. Partial Rights provide a mechanism for allowing "consent" and "authorisation" rights to be built into systems at the core level rather than as add-ons.

Future work includes the completion of a Set-Based Access Control model which utilises both Partial Permissions and Partial Rights. The model will provide options for implementing consent as well as mechanisms which allow authorisations to be given prior to, at the time of, and after an access is required. Implementation of the model in the health domain will be considered.

8 References

Al-Kahtani, M. A. and Sandhu, R. 2002, 'A Model for Attribute-Based User-Role Assignment', *18th Annual Computer Security Applications Conference*, IEEE, Las Vegas, Nevada, USA, p. 353

Alotaiby, F. T. and Chen, J. X. 2004, 'A Model for Team-based Access Control (TMAC 2004)', *International Conference on Information Technology: Coding and Computing (ITCC'04)*, IEEE, Las Vegas, Nevada, USA

Bell, D. E. and LaPadula, L. J. 1976, *Secure Computer System: Unified Exposition and Multics Interpretation*, The Mitre Corporation.

Chen, F. and Sandhu, R. S. 1996, 'Constraints for role-based access control', *Symposium on Access Control Models and Technologies*, ACM Press, New York, NY, USA, Gaithersburg, Maryland, US

CSIS *Centre for Secure Information Systems Security Glossary*. viewed 4th July 2005, <http://csis.gmu.edu/glossary/merged_glossary.html>

Dovier, A., Piazza, C., Pontelli, E. and Rossi, G. 2000, *Sets and constraint logic programming*. viewed 5, <<Go to ISI>://000167879300003>

Ferraiolo, D. and Kuhn, R. 1992, 'Role-Based Access Control', *15th National Computer Security Conference*, Baltimore, MD

Ferraiolo, D. F., Ahn, G.-J., R.Chandramouli and Gavrila, S. I. 2003, 'The Role Control Center: Features and Case Studies', *8th ACM Symposium on Access Control Models And Technologies*, ACM Press New York, NY, USA, Como, Italy, pp. 12 - 20

Georgiadis, C. K., Mavridis, I., Pangalos, G. and Thomas, R. K. 2001, 'Flexible Team-Based Access Control Using Contexts', *SACMAT '01*, ACM, Chantilly, Virginia, USA, pp. 21-27

Kern, A. and Walhorn, C. 2005, 'Rule Support for RoleBased Access Control', *Symposium on Access Control Models and Technologies 2005*, ACM Press, New York, NY, USA, Stockholm, Sweden, pp. 130-138

Li, N. and Mitchell, J. C. 2002, 'Design of a Role-based Trust-management Framework', *IEEE Symposium on Security and Privacy, 2002*, IEEE

Li, N. and Mitchell, J. C. 2003, 'RT: A Role-based Trust-management Framework', *Third DARPA Information Survivability Conference*

Schneier, B. 2000, *Secrets and Lies*, John Wiley & Sons, Inc., New York.

Thomas, R. K. 1997, 'Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments', *RBAC '97*, ACM, Fairfax Va USA, pp. 13-19