

A Formal Approach to Modelling Delegation Policy Based On Subject Attributes And Role Hierarchy

OGUNDELE O.S., ALESE B.K., FALAKI S.O. AND ALOWOLODU O.D.

Department of Computer Science, The Federal University of Technology, Akure, Nigeria.

oloruntoba78@yahoo.com

ADEWALE O.S

Director, Computer Resource Centre, The Federal University of Technology, Akure, Nigeria.

Abstract

There are considerable number of approaches to policy specification both for security and policy driven network management. This specification sort security policies into two basic types: authorization and obligation policies. Most of the researches in security policies specification over the years focus on authorization policy modelling. In this paper, we report our approach in developing an information security policy model with specific emphasis on delegation of roles as a form of obligation policy. Whilst noting the previous research works on delegation modelling, we considered subjects and roles attributes in refining and formulating delegation relation attributes rules using concept of set theory. The work was further extended by developing a formal model for role hierarchy based on permissions and integrating it into the delegation model developed to eliminate flatness of subject roles. Future works proposed include the development of a formal model for revocation after delegation and extension of the model with the principle of separation of duties.

Categories and Subject Descriptors: F.4 [Mathematical Logic and Formal Languages] - F.4.1 Mathematical Logic – Model Theory.

D.4 [Operating System] – D.4.6 Security and Protection – Access Control.

General Terms: Information Security, Delegation Policy, Modelling, Role Hierarchy

Additional keywords and Phrases: Subjects, Objects, Roles, Attributes

1. INTRODUCTION TO SECURITY POLICIES

According to Slomam and Lupu (2002), security policies can be sorted into two basic types: authorization and obligation policies. Authorization policies specify what an action a given subject (agent, roles, user or process) is permitted or forbidden to perform on a set of target objects. The concept is similar to the role – based access control (RBAC). Authorization policy can either be positive (defining the actions subjects are permitted to perform on the target objects) or negative (specifying the actions subjects are forbidden to perform on target object). Therefore, authorization policies are used to define access control rules implemented by several types of mechanisms in a network security system such as packet filters, Kerberos and Virtual Private Networks (VPNs).

Authorization policies are of the form:

```
auth [+/-] <subject>      ───────────▶ if <condition> then
                                <target> <action>;
```

1.1

Obligation policies are rules that specify what activities a subject must or must not perform on a set of target object under an optionally specified condition or system event. In other words, obligation policies are used to specify job functions related to security management. Obligation policies are event triggered condition – action rules that can be used to define the activities subjects must perform on objects in the target domain i.e. the duties of these subjects. In network security context, obligation policies can be used to specify the behaviour of mechanisms such as logging agents, intrusion detection systems and watchdogs.

Obligation policies are of form:

```
on <event> do
    if <conditions> then
        <target> <action>;
```

1.2

2. SECURITY POLICY REPRESENTATION

There are considerable number of approaches to policy specification both for security management and policy driven network management purposes as reported by Sloman and Lupu (2002). However, Model Based Management (MBM) proposed in (Luck et al, 2001 and Luck et al, 2002) will be adopted as the basis of the research policy modelling. The MBM approach supports the building of policy hierarchies by means of an interactive graphical design. It adopts concept of object oriented design and employs a model of the system that is vertically structured into three sets of layers. Three abstraction levels are recognised: Roles and Objects (RO), subjects and Resources (SR) and Processes and Hosts (PH). Each level is a refinement of the superior level in the sense of policy hierarchy (Moffet and Sloman, 1993). The uppermost level (RO) is based on the concept of Role-Based Access Control (RBAC) (Sandhu et al, 1996 and Ferraiolo et al, 1999). The Roles which are the acts of the people working in the modelled environment, and Objects of the modelled environment which should be subjected to Access Control and AccessModes form the main classes at the first level.

The Second level (SR) offers a system view of the model defined from the standpoint of the services that the system will provide. Objects of these classes represent (a) the people working in the modelled environment (Users); (b) Subjects acting on the user's behalf (SubjectTypes), (c) Services in the modelled environment that are used to access Resources (Services); (d) dependency of service(s) on another service(s) (ServiceDependency), and (e) Resources in the modelled environment (Resources).

The lowest level (PH) is responsible for modelling the mechanisms that will be used to implement the security services defined in SR

AccessPermission can be further sub divided into one or more ServicePermissions which expresses an authorization for SubjectType (on behalf of user) to use a Service to access Resource. Further to this, a class ProtocolPermission is now defined in the lowest level PH as the rules that either validate or invalidate the ServicePermission.

Policies development, formalization and validation must start at the Subject Resources level where the following entities have been earlier identified at this level, the User Objects (U_{sr}) of the system, the SubjectType (SubTy), the Services (Srv) and the Resources (Rsc).

A problem of MBM occurs when dealing with large system since the representation of the policies and objects tends to lose much of its understand ability and getting obscure due to the great number of elements (Albuquerque et al, 2005). Another issue is that it does not model obligation policy since it is built on RBAC strictly.

Albuquerque et al, 2005 designed the Diagram of Abstract (DAS). It was introduced as a layer immediately above the PH level as an extension of the MBM. Its main objective was to describe the overall structure of the system in a modular fashion. i.e. to cast the system into its building blocks (Abstract Subsystem) and to indicate the connection between them. Therefore DAS is a graph comprises as Abstract Subsystems as nodes with edges that represent the possibility of bidirectional communication between ASs. An AS, in turn, is an abstract view of a system segment; i.e. a simplified representation of a given group of system components. As such, an AS is itself a subgraph of a DAS.

This model represents a typical network environment, for which three *AccessPermissions* are defined at the uppermost level (RO), in order to regulate the access rights of the users in the internal network with respect to service request. To complete the modelling, each AS in a DAS is expanded into a detailed view of the actual mechanisms of the system; i.e. the PH level. it can be observed that modeling through abstract subsystems offers concrete advantages in the conciseness and understandability of the model, as well as providing an intelligible view of the system architecture.

Aside from the aforementioned modeling improvements, the DAS is also advantageous to policy support and model validation.

DAS also has a problem dealing with large system since the representation of the policies and objects tends to lose much of its understand ability and getting obscure due to the great number of elements (Albuquerque et al, 2005). Another issue is that it does not model obligation policy.

3. INFORMATION SECURITY POLICY MODELING

The information security policy model (PolM) proposed is a 5 – tuples represented as follows:

$$PolM = \{RolObj, SeClr, SBjRsv, PerM, Dlg\} \quad 3.1$$

Where

- (i) RolObj is a set containing the different roles (Rol_n) and objects (Obj_n) belonging to the various ROLES and OBJECTS within the modelled environment.

Therefore,

$$RolObj = \{Rol_n \in Rol \wedge Obj_n \in Obj\} \quad 3.2$$

- (ii) SeClr denotes a set of SECURITY CLEARANCE which corresponds to a set of disjoint classes of sensitivity level and category sets. Instead of being restrictive in using the terms employed in the military system such as “TOP SECRET”, “SECRET”, “CONFIDENTIAL”, “UNCLASSIFIED”, e.t.c; SeClr shall be defined as follows

$$SeClr = \{seclr_1, seclr_2 \dots \dots \dots seclr_n\} \quad 3.3$$

Where n is a finite integer and the relationship between two security clearance within this set is denoted by \leq where \leq is a partial order on $seclr_i$ for $1 \leq i \leq n$ such that $(SeClr, \leq)$ is a partial ordered set.

- (iii) SBJRsv denotes the set of subjects (Sbj_n), Subject dependency ($SbjDep_n$) and Resources (Rsv_n) and Resources dependency ($RsvDep_n$) within the modelled environment.

Thus,

$$SBJRsv = \{sbj_n, SbjDep_n \in SBJ \wedge Rsv_n, RsvDep_n \in RSV\} \quad 3.4$$

- (iv) PerM is defined as the permission which shall be referred to as the authorization constraint. It modelled the authorization policy aspect of the modelled environment and is a set containing Service Permission ($SrvPm$) and Access Permission ($AccPm$) which is a function of Access mode ($AccM$) as granted by the Security Clearance ($SeClr$).

Therefore,

Access Permission is a function $AccPm$ which allows explicit permission for Subject (Sbj) to access Objects (Obj) in a way defined by an AccessMode ($AccM$) as granted by the Security Clearance ($SeClr$).

$$AccPm = f\{Sbj \Leftrightarrow Obj \text{ as defined by } AccM\} \quad 3.5$$

For all $Sbj_n \in Sbj, Obj_n \in Obj$ and $AccM_n \in AccM$

Subject to;

$$SeClr = \begin{cases} Authorized, & 1 \leq i < n \\ Not Authorized, & otherwise \end{cases}$$

On the other hand,

the triples of Subject, Object and Access Mode not supported by the function Access Permission denotes negative authorizations and are completely forbidden but must be enforce by the security mechanisms.

From the above, authorization policy is define in RoObj as a class Access Permission which allow a subject through the define Role to access a particular Object in a way approved by the Access Mode subject to Security Clearance.

Service Permission is a function SrvPm which allows explicit permission of Subject (Sbj) to access Resources (Obj) depending on the availability of the Resources and Security Clearance (SeClr). Service Permission (SrvPm) can therefore be viewed as the relations between UserObjects, SubjectType, the Services and Resources according to the object security clearance and is depicted as

$$SrvPm = f(Sbj_n \Leftrightarrow Obj_n \text{ subject to Seclr}) \quad 3.6$$

In developing the policy model, the following assumptions will be made:

- i) The classification of an object is always at least as high as the maximum classification of the objects it contains.
- ii) All Access permission requests must have a classification label achieve by security clearance.
- iii) A request on an object or resources by a subject or services can only be granted if the clearance level of the subject or services is equal or greater than the classification label of the object or resources.
- (v) Dlg is defined as the delegation (Dlg), which is the mechanism that enables active entity(ies) in a system to transfer authority to another active entity(ies) in order to carry out some functions on its behalf based on trust (Trt) (Barka and Sandhu, 2000). This model the Obligation policy aspect of the modelled environment. Abdallah and Takabi (2010) developed a formal representation for delegation and integrate it into RBAC models; Chunxiao et al, (2006), Barka and Sandhu (2000), and Barka and Sandhu (2000) developed an attribute based delegation model and its extensions, a role based delegation model and some extensions and role based delegation model/Hierachical roles respectively. Hence, the delegation model proposed extract the best features of their works, refine it and introduce subject attributes as an extension of delegation in order to satisfy the condition for obligation policy.

Hence, when Subjects (Sbj) are assigned Roles (Rol) and are authorised through Permission (PerM) to perform tasks on Object (Obj) and or Resources (Rsv). Thus, we define

$$Role_Permission : \{Sbj \leftrightarrow Rol\} \text{ are authorised by } PerM \exists AccPm \vee SrvPm$$

3.7

While

$$Task_Permission : \{Sbj \leftrightarrow ((Obj) \vee (Rsv)) \text{ as defined by Role_Permission}\}$$

3.8

Therefore, the Relations: Role_Permission that assigns roles to each subjects; and Task_Permission that allowed subjects performed tasks on each objects or resources form the main classes and serve as the basis of the delegation modelling.

Information Security Policy Delegation (PoDl_g) is the ability of delegator (Subjects) to assigned roles to a delegatee (another subject or group of subjects). Policy Delegation can only be said to be successful if the delegator has the capability to allocate roles and the delegatee should be capable of being assigned the roles. For the purpose of the modelling, the following assumptions will be made:

- (i) Each Subject is assigned at least one Role.
- (ii) All Roles assigned to known Subject are recognized roles (Subset of Roles).
- (iii) Each recognised role is associated with a non – empty set of tasks permitted to perform. i.e. the task permitted by each roles are valid (subset of Tasks.)
- (iv) Roles are defined in hierarchical manner as stated in (Barka and Sandhu, 2000) but which may be subjective.
- (v) All Subjects within the modelled environment have a well defined attributes.
- (vi) Subjects may have one or more attributes but that are related.
- (vii) Objects and Resources within the modelled environment all have defined attributes.
- (viii) Roles are defined within the context of separation of duties (SoD) of subjects.

Delegation can be sorted into two by refining the Subjects, Roles and Role_Permissions into two differentiable units; Original (Subjects, Roles and Role_Permissions) and Delegated (Subjects, Roles and Role_Permissions).

From this we represent Core_Role_Permission and Core_Task_Permission as follows:

$$\begin{aligned}
 \text{Cor_Role_Permission} = \\
 SBj_o((Rol_o)) \text{ authorised by } PerM_o[Role_Permisision_o] \Leftrightarrow \\
 SBj_d((Rol_d)) \text{ authorised by } PerM_d[Role_Permisision_d]
 \end{aligned}$$

3.9

Where Role_Permission is as defined in equation 3.7 and the subscript o and d denote original and delegated respectively.

Equation 3.9 can further be simplify as follows by factorisation

$$\begin{aligned}
 \text{Cor_Role_Permission} = \\
 SBj_{o\wedge d}((Rol_{o\wedge d})) \text{ authorised by } PerM_{o\wedge d}[Role_Permisision_{o\wedge d}]
 \end{aligned}$$

3.10

While

$$\begin{aligned} \text{Core_Task_Permission} &= \text{Sbj}_o \leftrightarrow \\ &((\text{Obj}_o) \vee (\text{Rsv}_o)) \text{ authorised by } [\text{Role_Permission}_o] \end{aligned} \quad 3.11$$

Therefore, Core Delegation is defined as a relation as follows:

$$\text{Core_Dlg} : \{ \text{Core_Role_Permission} \leftrightarrow \text{Core_Task_Permission} \} \quad 3.12$$

Hence, the relation Core_Delegation relates Subjects to the Roles which originally have been assigned to them as well as to the Roles which have been delegated to them. Thus, a Subject is permitted to assume Role to carry out Task originally assigned to it in addition to those Role(s) and Task(s) delegated to it.

Core_Delegation can be sorted into Independent_Core_Delegation and Dependent_Core_Delegation.

Independent_Core_Delegation identify the Subject Original Roles and the Subject inherited Role (Subject Delegated Roles).

$$\text{Ind_Cor_Dlg} \cup \text{Dep_Cor_Dlg} = \text{Core_Dlg} \quad 3.13$$

while

$$\text{Ind_Cor_Dlg} \cap \text{Dep_Cor_Dlg} = \emptyset \quad 3.14$$

Core_Delegation can also be Permanent or Temporary.

Permanent Core delegation can be formalized as follows:

$$\begin{aligned} \text{Pr_Core_Dlg} &= \\ f \left\{ \begin{array}{l} \text{Core_Dlg} | \exists (\text{Rol}_o, \text{Rol}_d \in \text{Rol}_{Pr}) \text{ and } (\text{Sbj}_o^i, \text{Sbj}_d^k \in \text{Sbj}_{Pr}^n) \\ \text{where} \\ (\text{Sbj}_o^i(\text{Rol}_{o(Pr)})) \rightarrow (\text{Sbj}_d^k(\text{Rol}_{d(Pr)})) \text{ authorised by Role_Permission}_o \end{array} \right\} \end{aligned} \quad 3.15$$

This can be further simplify as

$$\text{Pr_Core_Dlg} = f \left\{ \begin{array}{l} \text{Core_Dlg} | \exists (\text{Rol}_o, \text{Rol}_d \in \text{Rol}_{Pr}) \text{ and } (\text{Sbj}_o^i, \text{Sbj}_d^k \in \text{Sbj}_{Pr}^n) \\ \text{where} \\ (\text{Sbj}_{o=d(Pr)}^{i=k}(\text{Rol}_{o=d(Pr)})) \text{ authorised by Role_Permission}_o \end{array} \right\} \quad 3.16$$

While Temporary Core Delegation can formalized as follows:

$$\text{Tr_Core_Dlg} = f \left\{ \begin{array}{l} \text{Core_Dlg} | \exists (Rol_o, Rol_d, Rol_{Tr} \in Rol_{Pr}) \text{ and } (Sbj_o^i, Sbj_d^k \in SBj_{Pr}^n) \\ \text{where} \\ (Sbj_o^i(Pr)(Rol_o(Pr))) \rightarrow (Sbj_d^k(Tr)(Rol_d(Tr))) \text{ authorised by Role_Permission}_o \end{array} \right\}$$

3.17

This can further be simplify as

$$\text{Tr_Core_Dlg} = f \left\{ \begin{array}{l} \text{Core_Dlg} | \exists (Rol_o, Rol_d, Rol_{Tr} \in Rol_{Pr}) \text{ and } (Sbj_o^i, Sbj_d^k \in SBj_{Pr}^n) \\ \text{where} \\ (Sbj_o^{i \geq k}(\Pr \geq Tr)(Rol_o(Tr))) \text{ authorised by Role_Permission}_o \end{array} \right\}$$

3.18

4. MODELING DELEGATION POLICY WITH ATTRIBUTE OF SUBJECTS, ROLES AND OBJECTS

Zhang et al, 2003; identify three types of situations in which delegation takes place: backup of roles, decentralization of authority and collaboration of work. Many studies have been done in delegation (Stein, 1987; Moffett, 1990; Gasser and McDermott, 1990), and considerable attention is paid to human to-human delegation (Zhang *et al*, 2001, Zhang *et al*, 2003; Barka and Sandhu, 2000a; Barka and Sandhu 2000b; Barka , 2002; Barka and Sandhu 2004). But there are still some problems in delegation needing to be solved (Chunxiao et al, 2006):

1. Because delegation is controlled by the delegator itself, a malicious user can delegate some important permission to low level delegates.
2. The delegation security relies heavily on the security administrator.
3. Delegation prerequisite condition cannot restrict the scope of delegates more strictly.
4. It may be difficult for a delegator to select qualified delegates.

Chunxiao et al, 2006 designed an attribute based model and its extension (ABDM_X). They proposed a novel delegation model ABDM and its extension ABDM_X. As a delegation model based on permission and user's attributes, the main feature of it is that it uses user and permission attributes expression as a part of delegation constraint. ABDM is a securer delegation model for it can restrict delegatee candidates more strictly. ABDM_X is more flexible than ABDM in delegation. In ABDM_X, a delegator can temporarily delegate *Non Monotonic Permission* to low level users without causing any security problems. Both ABDM and ABDM_X can be used in temporary and permanent delegation and make delegation securer and more flexible.

The Delegation Model designed and described above is extended by refining ABDM_X and extending it into our model with the formalization of Subjects' attributes, and extension with Role hierarchy Model.

The concept of attribute – based delegation identify that all Objects, Subjects and Roles within the modelled environment must have unique attributes and access control are enforced based on computation of attributes

expression relations. In some of the existing models (Al-Kahtani and Sandhu, 2002; Al-Kahtani, 2003), only users can have attributes and attributes expression. Chunxiao et al, (2006), made an improvement over this by extending attributes and attributes expression to permissions. They identify user's Delegation Attributes Expressions (DAEs) to include user's qualifications and abilities, while permission's DAE indicate a Delegation's qualifications and abilities required by the permission in delegation. We extend the works with the introduction of Roles and Objects attributes; attributes expression and the formulation of Delegation Relation and Delegation Relation Attributes Rules. Thus, we define the following notations:

- Sbj_Atr , Rol_Atr , Obj/Rsv_Atr , Rol_Hry , Sbj_Atr_Exp , Rol_Atr_Exp , are Subject attributes, Role attributes, Object/Resources attributes, Role Hierarchy, Subject attributes expression and Role attributes Expression respectively.
- Delegation Attribute is a relation of Sbj_Atr , Rol_Atr and Obj/Rsv_Atr given as

$$Dlg_Atr = \{Dlg_Atr | where \exists (Sbj_Atr_i \in Sbj_Atr) \times (Rol_Atr_i \in Rol_Atr) \times (Obj/Rsv_Atr_i \in Obj/Rsv_Atr) \forall Sbj_Atr, Rol_Atr, Obj/Rsv_Atr \in Atr\}$$

4.1
- Delegation attribute expression is formulated as follows based on the validity of the rules given below:

Table 1: Delegation Attribute Relation Expression

RULES	ATTRIBUTES RELATION EXPRESSION	DESCRIPTION
Rule 1	<p>if $\exists (Sbj_Atr_Exp_i \cup Sbj_Atr_Exp_k)$ and $(Rol_Atr_Exp_i \cup Rol_Atr_Exp_k) \forall Sbj_i, Sbj_k \in Sbj$ and $Rol_i, Rol_k \in Rol$</p> <p>Then</p> $Dlg_Atr_Exp = \{Dlg_Atr Rol_i: Rol_i(Sbj_i) \rightarrow Sbj_k\}$ $\therefore Sbj_k \rightarrow Obj/Rsv_i$ <p>where $i = k$</p> <p style="text-align: right;">4.2</p>	A situation where Sbj_i of Rol_i and Sbj_k of Rol_k share the same or similar attributes and attributes expression.
Rule 2	<p>if $\exists (Sbj_Atr_Exp_i \cap Sbj_Atr_Exp_k)$ and $(Rol_Atr_Exp_i \cap Rol_Atr_Exp_k) \forall Sbj_i, Sbj_k \in Sbj$ and $Rol_i, Rol_k \in Rol$</p> <p>Then</p> $Dlg_Atr_Exp = \{Dlg_Atr Rol_i: Rol_i(Sbj_i) \rightleftharpoons Sbj_k\}$ $\therefore Sbj_k \rightleftharpoons Obj/Rsv_i$ <p>Where $i \geq k$</p> <p style="text-align: right;">4.3</p>	A situation where Sbj_i of Rol_i and Sbj_k of Rol_k share some similar attributes and attributes expression but not all.
Rule 3	<p>if $\exists (Sbj_Atr_Exp_i \cap Sbj_Atr_Exp_k) = \emptyset$ and $(Rol_Atr_Exp_i \cap Rol_Atr_Exp_k) = \emptyset, \forall Sbj_i, Sbj_k \in Sbj$ and $Rol_i, Rol_k \in Rol$</p> <p>Then</p> $Dlg_Atr_Exp = \{Dlg_Atr Rol_i: Rol_i(Sbj_i) \nrightarrow Sbj_k\}$	A situation where Sbj_i of Rol_i and Sbj_k of Rol_k did not share nor

	$\therefore Sbj_k \nrightarrow Obj/Rsv_i$ Where $i \neq k$	4.4	have any attributes and attributes expression at all.
--	---	-----	---

5. MODELING DELEGATION POLICY WITH ROLE HIERARCHY

It is pertinent to state that all the roles that we have been in consideration for delegation model above are flat roles and inheritance between roles are not absolutely considered. However, role hierarchy is to be considered with inheritance between roles to extend delegation and delegation attributes rules that we have modelled above. Abdallah and Takabi (2010) captured the concept of role hierarchy as a partial ordering relation denoted by “SENIOR”, on a set ROLE such that $r1 \text{ SENIOR } r2$ indicates that role $r1$ is higher in the role hierarchy than role $r2$. Being a partial order means that the relation is transitive, reflexive and anti – symmetric. A schematic diagram depicting role hierarchy is shown below

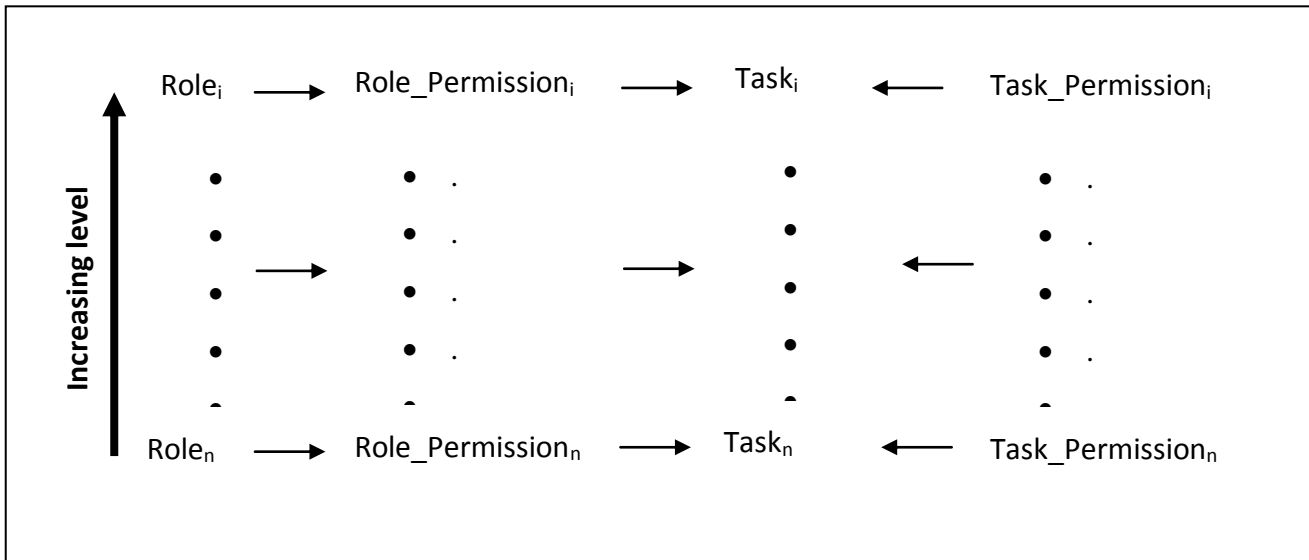


Fig 1: Regular Role Hierarchy

Therefore, we formalized Regular Role hierarchy *Rol_Hry* as follows:

$$\forall Rol_i \dots \dots \dots Rol_n \in Rol \quad \text{where } Rol_i \dots \dots \dots Rol_n \text{ are sets of Roles}$$

and

$$Role_Permission_i \dots \dots \dots Role_Permission_n \in Role_Permission \quad 5.1$$

$$if \exists Rol_i > Rol_{i-1} > \dots \dots \dots > Rol_n$$

and

$$Role_Permission_i > Role_Permission_{i-1} > \dots \dots \dots > Rol_n \quad 5.2$$

such that

$$Rol_i \neq Rol_{i-1} \neq \dots \dots \dots \neq Rol_n \quad 5.3$$

and

$$Role_Permission_i(Task_i) \cap Role_Permission_{i-1}(Task_{i-1}) \cap \dots \dots \dots \cap Role_Permission_n(Task_n) = \emptyset$$

5.4

where

$$n \leq level \leq i$$

Then, we have a valid Regular Role Hierarchy

From the above, we extend our delegation model above by formalizing and integrating it with regular role hierarchy

Therefore, Delegation Model based on Regular Role Hierarchy defined as *Dlg_Rol_Hry* can be formalized as follows:

$$(if \exists (Sbj_i, Sbj_n \in Sbj) \text{ and } (Obj_i, Obj_n \in Obj)) \text{ belonging to } (Rol_i, Rol_n \in Rol)$$

$$\forall Role_Permission_i, Role_Permission_n \in Role_Permission \quad 5.5$$

where

$$Rol_Hry : \{Rol, Role_Permission\} \text{ satisfied } Rol_Hry \text{ as defined in (5.1 to 5.4)} \quad 5.6$$

then

$$Dlg_Rol_Hry : Rol_i(Sbj_i) \rightarrow Sbj_n \quad 5.7$$

and

$$Sbj_n \leftarrow Sbj_Atr_Exp_i(Sbj_i) \quad 5.8$$

such that

$$Role_Permission_n(Rol_n) \subseteq Role_Permission_i(Rol_i) \quad 5.9$$

$$\text{Then } Sbj_n \rightarrow Obj_i \quad 5.10$$

This mean that based on valid computation of the above, we can have a delegation model based on attributes and Role Hierarchy.

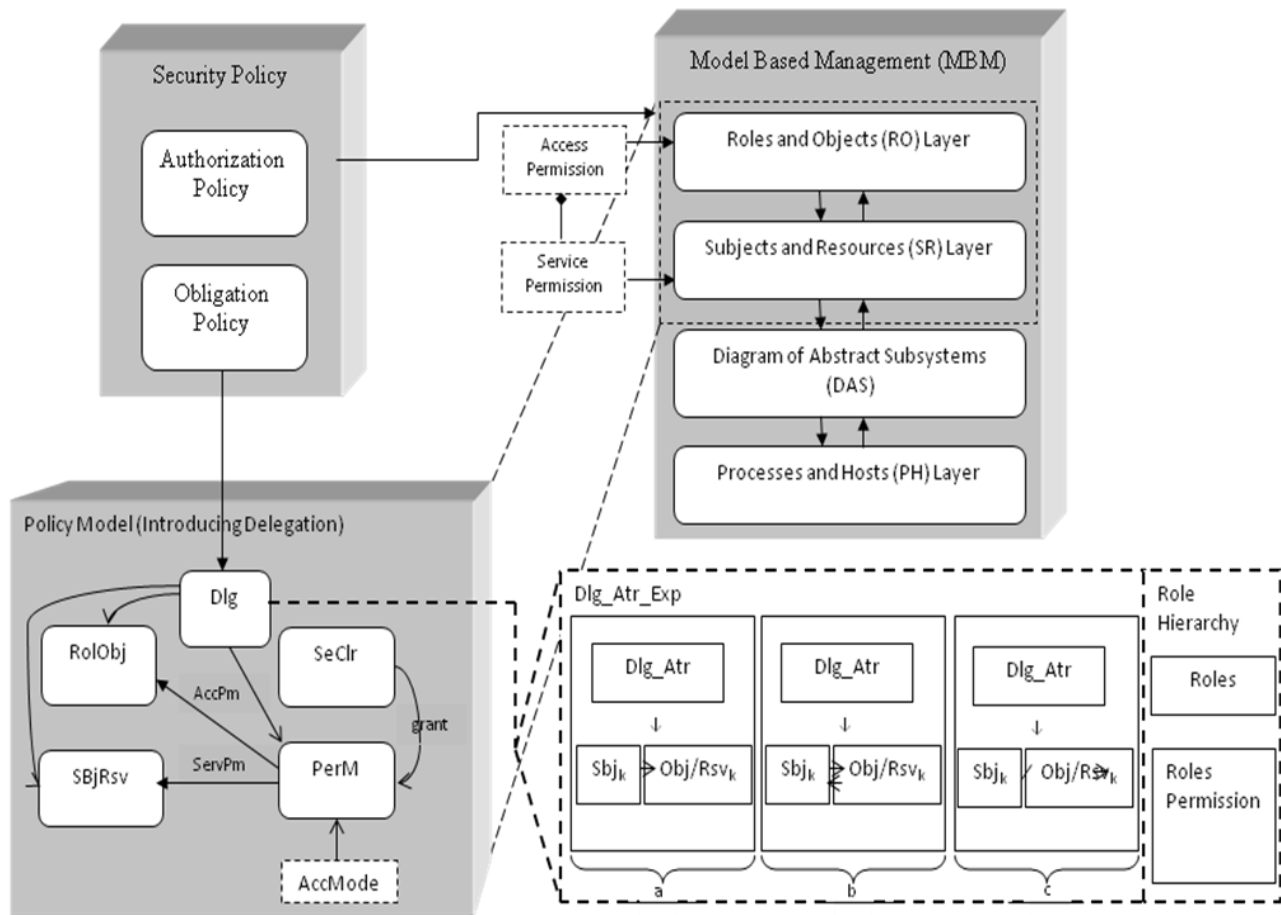


Fig 2: Schematic Architecture of the Information Security Model Designed.

6. CONCLUSION

In this paper, we have described the development of a formal model for information security design with particular emphasis on delegation. The delegation model developed outline new simple rules for a delegator to assign roles to a delegatee based on subjects and roles attributes. The work considered subjects and roles attributes in refining and formulating delegation relation attributes rules using concept of set theory. The work was further extended by developing a formal model for role hierarchy based on permissions and integrating it into the delegation model developed to eliminate flatness of subject roles. The model considered the hierarchy in subject roles as an extension to the formalization, and a pre – requisite condition for successful delegation to ensure secure access control.

Future works proposed include an analytical computation of the model, the development of a formal model for revocation after delegation, extension of the model with the principle of separation of duties and adaptability of the model to real world environment to validate and further refine the model.

7. REFERENCES

- ABDALLAH ALI E. AND TAKABI HASSAN, 2010. “Formalizing Delegation and Integrating it into Role-Based Access Control Models”, *Journal of Information Assurance and Security* 5 (2010) 021-030.
- ALBUQUERQUE JOAO PORTO DE, KRUMM HEIKO, AND LICIO DE GEUS PAULO, 2005. Policy Modeling and Refinement for Network Security Systems, *Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*.
- AL-KAHTANI, M.A., 2003. A family of models for rule-based user-role assignment, PhD Thesis. School of Information Technology and Engineering, George Mason University.
- AL-KAHTANI, M. A. AND SANDHU, R., 2002. A model for attribute-based user-role assignment, *Proceedings of the 18th Annual Computer Security Applications Conference*. San Diego California USA, IEEE Computer Society.
- BARKA, E. AND SANDHU, R., 2000a. Framework for role-based delegation models, *Proceedings of 16th Annual Computer Security Application Conference (ACSAC2000)*. New Orleans, USA, IEEE Computer Society.
- BARKA, E. AND SANDHU, R., 2000b. A role-based delegation model and some extensions, *Proceedings of 23rd National Information Systems Security Conference (NISSC)*. Baltimore, USA, NIST.

- BARKA E. AND SANDHU R., 2000. A Role-based Delegation Model and Some Extensions, In *proceedings of 16th Annual Computer Security Application Conference*, pp. 168-176, 2000.
- BARKA E. AND SANDHU R., 2004. Role-Based Delegation Model/ Hierarchical Roles (RBDM1), In *proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04)*, pp. 396-404, 2004.
- CHUNXIAO YE AND ZHONGFU WU AND YUNQING FU., 2006. An Attribute-Based Delegation Model and Its Extension, *Journal of Research and Practice in Information Technology*, Vol. 38, No. 1, February 2006.
- FERRAILOLO D. F., BARKLEY J. F, AND KUHN D. R., 1999. A role-based access control model and reference implementation within a corporate intranet. *ACM Transactions on Information and System Security*, 2(1):34–64, February 1999.
- GASSER MORRIE AND MCDERMOTT ELLEN., 1990. An Architecture for practical Delegation in a Distributed System, *IEEE Computer Society Symposium on Research in Security and Privacy*. Oakland, CA. May 7-9, 1990.
- MOFFETT J. D. AND SLOMAN M. S., 1993. Policy hierarchies for distributed system management, *IEEE JSAC Special Issue on Network Management*, 11(9), 11 1993.
- MOFFETT, J.D., 1990. Delegation of authority using domain based access Rules. PhD Thesis. Dept of Computing, Imperial College, University of London.
- LUCK I., SCHAFER C., AND KRUMM H., 2001. Model-based tool assistance for packet-filter design, In E. Lupu M. Sloman, J. Lobo, editor, *Proc. IEEE Workshop Policy 2001: Policies for Distributed Systems and Networks*, number 1995 in Lecture Notes in Computer Science, pages 120–136, Heidelberg, 2001. Springer Verlag.
- LUCK I., VOGEL S., AND KRUMM H., 2002. Model-based configuration of VPNs, In R. Stadtler and M. Ulema, editors, *Proc. 8th IEEE/IFIP Network Operations and Management SymposiumNOMS 2002*, pages 589–602, Florence, Italy, 2002. IEEE.
- SANDHU RAVI S, COYNE J. COYNE, FEINSTEIN HAL L., AND YOUMAN CHARLES E., 1996. Role-based access control Models, *IEEE Computer*, 29(2):38-47, February 1996.
- SLOMAN M. AND LUPU E. C., 2002. Security and management policy specification. *IEEE Network, Special Issue on Policy-Based Networking*, 16(2):10–19, March/April 2002.

- STEIN, L.A., 1987. Delegation is inheritance, *Proceedings of Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '87)*. New York, USA, ACM press.
- ZHANG, L.H., AHN, G.J. AND CHU, B.T., 2001. A rule-based framework for role-based Delegation, *Proceedings of 6th ACM Symposium on Access Control Models and Technologies (SACMAT)*. Chantilly, VA, USA, ACM press.
- ZHANG, X.W., OH, S. AND SANDHU, R. S., 2003. PBDM: A flexible delegation model in RBAC, *Proceedings of the 8th ACM Symposium on Access Control Models and Technologies (SACMAT'03)*. Como, Italy, ACM press.