大型分散式超文件系統上以角色爲主之授權模式
A Role-Based Authorization Model for Large Distributed Hypertext Systems

曾育民
Yuh-Min Tseng

詹進科
Jinn-Ke Jan

國立中興大學應用數學所
Institute of Applied Mathematics
National Chung Hsing University

國立中興大學應用數學所
Institute of Applied Mathematics
National Chung Hsing University

摘要

超文件系統的優點已被廣範確認，本篇論文提出一個以角色爲主存取控制方法且擴展使其適用在超文件系統的應用，我們的以角色爲主授權模式提供對不同種類資源的授權，正及負授權，及一種新的授權範圍。我們亦提出一個分散式的授權管理系統，當超文件系統的授權有改變時，我們的授權管理系統更易於爲修正及維護其授權。

關鍵字：超文件系統，授權模式，以角色爲主存取控制，正及負授權，授權範圍。

Abstract

The benefits of hypertext systems are widely recognized. This paper proposes an approach for role-based access control and extends it to deal with the applications for authorizations in hypertext systems. We propose a role-based authorization model which supports authorizations for different kinds of objects, positive and negative authorizations, and a new authorization domain. Based on this model, we also proposed an authorization system for hypertext systems. The administration of our system is a distributed approach which is easy to maintain and to manage the authorizations when the consent of subjects and objects in the hypertext system is altered.

Keywords: Hypertext systems, Authorization models, Role-based access control, Positive and negative authorizations, Authoriztion domains.

1. Introduction

A hypertext is modeled as a network of components related through a set of links anchored in a source and component destinations [9]. The notions of links, anchors and components are basic to the hypertext system. These links are connected to other components and they are not tied to any particular type of implementation or to any particular type of data. Users can navigate in the information space of hypertexts. For instance, the World Wide Web (WWW) ,"Intranet" and digital libraries are famous applications of hypertext systems.

Although hypertext systems have been attracting a lot of attention to the Intranet and digital libraries, the problem of authorization has not been widely investigated. Even though existing hypertext systems can provide fairly fine grained access control, they require some tools to centrally administer a particular WWW site. But as the amount of hypertext databases to be shared and distributed grows, the need for effective administration to restrict access to specific users will surely increase, and the administrative problems of authorization may limit the widespread usage of hypertext systems.

Both discretionary access control (DAC) and mandatory access control (MAC) policies are adequate for the traditional data management system [4,5,11]. But the lack of structure for hypertext systems makes the specification of authorization to specific unstructured parts of a component (e.g., a text, a picture, or an image) more difficult. Hence, the existing traditional authorization models are inadequate for the protection of information in large distributed hypertext systems [8,9,16]. Several projects [1,12,13] and models [2,10,16] for supporting authorization-based access control in hypertext systems are carried out and proposed. These systems and models have several limitations and drawbacks for the administration and maintenance of authorization systems. The detailed discussions are included in Section 2.

The remainder of the paper is organized as follows. Section 2 discusses the related work and presents drawbacks. This is followed by a short review of role-based access control in Section 3. It introduces

of role-based access control in Section 3. It introduces the concepts and advantages of role-based access control. In Section 4, we present our extended hypertext model that is related to the reference model in [16]. In Section 5, we introduce the structure and the elements of our role-based authorization model. Moreover, we present several new notations for authorizations for hypertext systems, such as, a new authorization domain, negative authorizations, considerations of different kinds of resources. In Section 6, we propose the architecture of our authorization system based on our authorization model. It also introduces the concept of how to authenticate the identities of users and determine the associated authorizations using credentials. In Section 7, we discuss administration and maintenances of authorizations related to the alterations of authorizations in hypertext systems. Section 8 gives our conclusions and presents future work for hypertext systems and role-based authorizations.

2. Related Work

Two projects [12,13] based on the Access-Control List (ACL) in the area of coordinated WWW distributed authorization have been done. One is the Phoenix project [13] which is a distributed hypermedia authoring system. The second project is the DCE Web project, which includes distributed authentication, consistent group administration across a domain, protection of nodes with ACLs, and remote administration of ACLs. The two projects have several drawbacks. The granting or revocation of access to a document requires the modification of the ACLs associated with several nodes. When the documents are stored in different servers, the existing systems do not advance any infrastructure for coordinating the administration of the ACLs.

Kahan [10] proposes a capability-based authorization model that attempts to improve these models based on the ACL mechanism, which provides authorization at the document and the presentation tree levels.

These models have several common limitations. One is that if two documents belong to the same directory, e.g., they are both associated with the same authorization domain, and both are subject to the same authorizations. Another is that a user who has access to a node can see and activate all links to other nodes, that is, these approaches do not consider the different nature of components and links, and do not protect the relationships between components. Finally, these models do not attend to the issue of the visual representation problem with links.

In 1996, Samarati et al. [16] presents an authorization model for distributed hypertext systems. The model has several drawbacks in management and maintenance of authorizations. One leads to authorizations for authoring. In a large distributed hypertext system, when the amount of users is large, this approach will also increase the complexity of the administrative process. Another problem is the modification of the authorization.

Recently, Barkley et al.[2] proposed an approach addressed as RBAC/Web in which they describe the benefits of role-based access control (RBAC) and implementation of RBAC on the WWW, and in particular as RBAC applies to an Intranet computing environment. But their approach does not involve the issue of modeling the distributed hypertext information between different sites, and does not consider the problem for administering authorizations in the distributed environment.

3. The Overview of RBAC

In this section, we will informally review the basic concepts of role-based access control that will be used later on in this paper. We refer the readers to [7,18] for additional details relating to the role-based authorization models and to [6,17,19] for details relevant to the features and applications of role-based access control.

With role-based access control (RBAC) [7,18], access permissions are based on the roles that individual users are as part of an organization. Users take on assigned roles. Access rights are grouped by role name, and the use of resources is restricted to individuals authorized to assume the associated roles. In an organization, roles usually have overlapping responsibilities and privileges, that is, users belonging to different roles may need to perform common operations. In this situation, by using the property of role hierarchies in RBAC, it would be efficient without specifying repeatedly the common operations for each role that is created. Role hierarchies can be established to provide for the natural structure of an organization or enterprise. Moreover, the features and concept of RBAC are policy neutral. It supports three well-known security principles: least privilege, separation of duties, and abstract permissions.

4. The Hypertext Architecture

Our hypertext model is inspired by the hypertext model proposed by Samarati et al.[16], and within it we modify some features to simply the future authorization processes. Our hypertext model divides a hypertext system into two different levels of abstraction,

the hyper level and the storage level. The hyper level is a logical representation of the hypertext documents, where a document comprises links to objects and to other documents. Documents and objects may be stored in different sites (servers). The storage level captures the persistent, storable objects making up the hypertext which consists of a set of hypertext documents. In the following, we focus on the storage level of this model.

The information stored at each site is structured into a presentation tree [10,12,13,16]. We assume that each presentation tree has a root document. We use the term node to refer to a document. Fig. 1 presents the classifications of contents for hypertexts. A node may comprises some basic objects and components. A basic object is the fundamental information in a node. The basic object may be different kinds of information, and will coincide with the whole node. A component may be a specific object or a link, and is a piece of a node that can be identified by a component identification. The concept of component is important for providing access control based on a fine granularity, that is, the authorizations of the subjects may be granted to the components of a node. Just as in a basic object, a specific object can be different kinds of information. A link may be a navigation link or a script link. Navigation links form the heart of a hypertext system. The traversable network structures formed by navigation links distinguish hypertext from other means of organizing information. A navigation link can connect to a node at a local site, or to a node at different sites in a hypertext system. Navigation links provide an approach to allow users to navigate the entire information space. A script link corresponds to a script to be executed. Executing a script means a query process to an on-line database, or a transaction process in the application of the business.

Fig. 2 illustrates an example of the logical representation of hypertext documents at a site in the hypertext system. The solid circle represents a container that is stored at a local site, and the dotted circle represents the connection to other nodes or the execute scripts at different sites in the hypertext system. All of the information for hypertext documents at a site are constructed into a tree structure [10,12,13,16]. Node R is the root document of the site. Other nodes may be the internal nodes at the local site, or the root nodes and internal nodes at different sites. In addition to basic objects included in a node, the node is divided into components which can comprise specific objects, such as navigation links connecting to other nodes, and/or the executions of scripts. In Fig 2., the components beginning with the string "Sobj" are specific objects, the components with the string "Node" are nodes (i.e., documents), and the components with the string "Scrt" are script executions. For instance, Node R has five components: connecting to a specific object (i.e., Sobj 1), linking two internal nodes (i.e., Node A and Node B), linking a node (i.e., Node C) at a different site, linking a script (i.e., Scrt D). Moreover, Node A has two components. One connects to a specific object 2 (i.e., Sobj 2) and the other links to Node E.

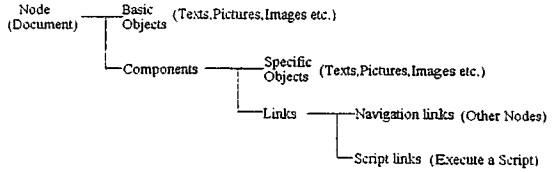

Fig. 1 Classifications of contents for hypertexts

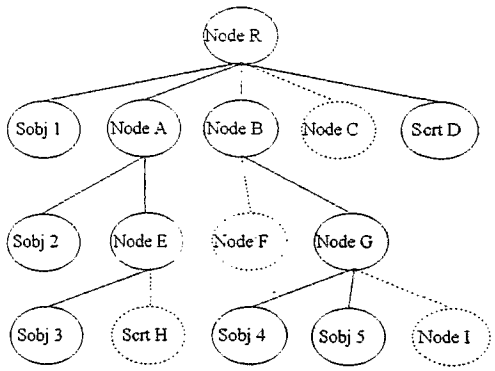

Fig. 2 A logical representation of hypertext documents at a site

5. The Role-Based Authorization Model

The potential benefits of RBAC have been stated, but without a precise definition of what RBAC constitutes [6,7,17-19]. In this section, we formally redefine and extend some characteristics of the RBAC model [7,18] adequate for the protection of information in hypertext systems. Let U be the set of users in the hypertext system, R the set of roles, O the set of objects (i.e., the resources or information of a hypertext system), and P the set of operational permissions on the objects.

5.1 Basic Notations and Formal Definitions

In this subsection, we present formal definitions of the basic concepts for role-based access control described informally in Section 3.

DEFINITION 1. (User-Role Authorization) An U-R authorization is a two-tuple $<u, rs>$, where $u \in U$ is the subject(user), $rs \subseteq R$ is the subset of roles, and an UR tuple is an one-to-many assignment relation. The UR set is the set of U-R authorization tuples.

The above tuple states that the user u has been made a member of roles rs. A user can be associated with one or more roles, and a role can have one or

more users. Roles are created for various job positions and privileges in an organization or a system, respectively.

DEFINITION 2. (Role-Permission Authorization) An R-P authorization is a three-tuple $< r, ps, os >$, where $r \in R$ is the role in the hypertext system, $ps \subseteq P$ is the subset of permissions, $os \subseteq O$ is the subset of objects. The RP set is the set of R-P authorization tuples.

An RP tuple states that the role r has been granted the permissions ps on the objects os. The permissions can be for specific types of operations which are dependent on the type of objects. For example, for text, operations might be visible or hidden; for links, operations can be visual/traversal (i.e. highlighted button), visual/untraversal, or hidden from the user.

According to Def. 1 and Def. 2, we present a derivation rule which can derive the user authorizations from User-Role Authorizations and Role-Permission Authorizations. The following is the first authorization implication rule.

Rule 1. (User Permission Authorization) An U-P authorization states as follows:

For each $u \in U$, $r \in R$, $ps \subseteq P$, and $os \subseteq O$,

if $\exists < u, rs > \in UR$, $\exists < r, ps, os > \in RP$ and $r \in rs$, then the user u can be granted permissions ps to access the objects os.

Rule 1 establishes that if a user is the member of one role, and the role has the authorization for the objects, then the user can be granted the authorization to access the objects os.

DEFINITION 3. (Role Hierarchy Authorization) An R-H authorization is a two tuple $< r1, r2 >$, where $r1, r2 \in R$ are the roles in the hypertext system, and $r1$ is the parent role, and $r2$ is the child role. The RH set is the set of R-H authorization tuples.

The above tuple states that roles $r1$ and $r2$ have the ancestor relationship. The immediate parent relationship can also be represented as an ordered pair $(r1 > r2)$, where $r1$ is the immediate parent and $r2$ the child and ">" is a relation "contains". The role $r1$ inherits all permissions from the role $r2$. These relationships can create some hierarchies. Role hierarchies are a natural way of organizing roles to reflect authority and responsibility. These hierarchies are partial orders. A partial order is a reflexive and transitive relation. Therefore, the most powerful roles are represented at the top of the hierarchies with the less powerful roles being represented at the bottom.

According to the above definition, we present a derivation rule which can derive some new authorizations from Role-Permission Authorizations. The derivation rule is defined as follows.

Rule 2. (Role Inheritance Authorization) An R-I authorization states as follows:

For each $r1, r2 \in R$, $ps \subseteq P$, and $os \subseteq O$,

if $\exists < r1, r2 > \in RH$ and $\exists < r2, ps, os >$ $\in RP$, then $< r1, ps, os > \in RP$.

Rule 2 establishes that if one role has the authorizations on the objects, all other roles which precede this role in the partial order hierarchies are granted the same authorizations.

5.2 Authorization Domains

When the number of documents in the hypertext system is very large, to specify authorizations for each single object becomes ineffective. The approach to overcome this problem is to allow authorizations to be specified in the authorization domain. An authorization domain is a set of objects grouped for administrative purposes. Therefore, we propose a new authorization domain which is the shape of "subtree". Because the documents at each site in hypertext systems are constructed into a presentation tree, according to the features of the tree hierarchy, we can define the subtree of the presentation tree into a new authorization domain. This approach is more adequate for the protection of information in hypertext systems.

In our model, the objects specified for authorizations may be nodes, the components comprising a node, and the authorization domains. Each authorization domain may be directories [16], and subtrees of each presentation tree in the hypertext system. Each subtree is defined at some site. The authorizations are defined as follows.

DEFINITION 4. (Object Hierarchy Inheritance) An O-H inheritance is a two tuple $< o1, o2 >$, where $o1, o2 \in O$ are the objects in the hypertext system, and $o1$ is the parent object, and $o2$ is the child object. The OH set is the set of O-H inheritance tuples.

This definition is added to the RBAC model representing the structure of the information at a site in the hypertext system. The information is generally structured as a tree[10,12,13,16]. An OH tuple states that the object $o2$ is a component of object $o1$. That is, object $o1$ may connect to object $o2$ using a link, or object $o2$ is the basic object or specific object. Notice that object $o1$ must be a node and object $o2$ may be any kind of information.

According to the above definition, we present a derivation rule which can glean some new authorizations from Role-Permission Authorizations. The derivation rule is defined as follows.

Rule 3. (Object Inheritance Authorization) An O-I authorization states as follows:

For each $o1 \in O$, $o2 \in O$, $ps \subseteq P$, $r \in R$, and $os \subseteq O$,

if $\exists < o1, o2 > \in OH$, $\exists < r, ps, os > \in RP$, and $o1 \in os$, then $< r, ps, osa > \in RP$, where $osa = os \cup \{o2\}$.

In Rule 3, when a role is granted a specific permission on $o1$, a user is granted membership in the role. Besides the user can be granted the permissions ps on the object $o1$, and also ownership in the permissions for the object $o2$ with dependence upon the type of data in the object $o2$.

5.3 Negative Authorization

Negative authorizations[3] are attractive since they allow exceptions to be specified, in particular for authorization domains, which make the management of authorizations more efficient. In our authorization model, the specification of negative authorization is issued with respect to the subjects and objects of authorizations. The subjects of authorizations are roles. The objects of authorizations may be nodes, components of a node, or authorization domains (i.e., subtrees or directories). In the following, we will extend our model for negative authorization with respect to subjects and objects in the hypertext system.

DEFINITION 5. (Role-Negative Authorization) An R-N authorization is a three-tuple $< r, -ps, os >$, where $r \in R$ is the role in the hypertext system, $ps \subseteq P$ is the subset of permissions and negative sign "-" means the denied authorizations, and $os \subseteq O$ is the subset of objects. The RN set is the set of R-N authorization tuples.

An RN tuple states that the role r has been denied the permissions ps on the objects os. A subject is denied access to an object, if (1) the subject has no authorization for the object; or (2) the subject has a negative authorization for the object.

According to the above definition, we present a derivation rule which is useful for expressing exceptions to implicit authorizations. The derivation rule is described as follows.

Rule 4. (Negative Role Authorization) An N-R authorization states as follows:

For each $r \in R$, $ps \subseteq P$, and $os \subseteq O$,

if $\exists < r, -ps, os > \in RN$,

then it constraints the implication rule Rule 1..

Rule 4 establishes that if a role has negative authorizations on some objects, the positive authorization granted to the subject becomes blocked.

5.4 Authorization Types and Modes

Based on the access types and modes for the authorization model proposed by Samarati et al. [16], authorization subjects are roles predefined in system, but not users. Authorizations refer to the objects stored at sites, i.e., basic objects, specific objects, nodes, and the execution of scripts. In addition, authorization objects can also be directories and subtrees.

The power of RBAC as an access control mechanism is the concept that an operation may theoretically be anything. That is, the concept of the RBAC supports the feature of abstract permission [7,18]. Therefore, our role-based authorization model can define the different operational approval that is associated with the different kinds of data. When a new kind of resource is joined to the hypertext system, based upon the feature and functionality of the resource we define the new permission. The consent for authorization for the different kinds of data are defined as follows.

DEFINITION 6. (Object-Permission Authorization) An O-P authorization is a two-tuple $< o, p >$, where o is an object, and p is an operational permission. This authorization changes the set of permissions P and the set of objects O as follows. Extend the permission set P, $P = P \cup \{p\}$. In additions, if o is a new kind of data, extend the set O, $O = O \cup \{o\}$.

Authorization $< o, p >$ states that the administrator can define the new operational permission p that is associated with the object O. In general case, this authorization is operated when new kind of resource is joined into the hypertext system. In fact, the administrator usually may predefine all fundamental access types and modes at initial stage of creation for a hypertext system.

6. Authorization system

In this section, we will propose a system framework that provides the role-based authorization model described in the previous section for a large distributed hypertext system. In accordance with the

authorization model discussed in Section 5, how the functions of authorizations and the inference of the derivation rules in our authorization model are embedded into the different administrators, is described in the following subsections.

6.1 Component of Authorization Administration

Our authorization system consists of two administration components: System Authorization Server (SAS) and Local Authorization Server (LAS). The responsibilities of the two administration components are introduced in details as follows.

1. System Authorization Server (SAS): The SAS in a hypertext system is a unique server that generates all roles in the hypertext system. The SAS assigns the authorization of users as members of roles.
   - Creating roles.
   - Constructing role hierarchies.
   - Granting user-role authorization.

   Finally, the SAS transmits the role set R and the RH set to the local authorization server at each site.
2. Local Authorization Server (LAS): The LAS of each site can grant permission to those roles on the objects stored at the site, it has also a mechanism for checking authorization when a user requests to have access the objects at the site. These are listed as follows.
   - Constructing/Modifying documents.
   - Creating/modifying authorization domain.
   - Creating new permissions.
   - Granting/revoking role-object authorizations.
   - Granting/revoking role-negative authorizations.
   - Authenticating users.

   Finally, the basic operation which an authorization system must provide includes checking the authorizations of users.

6.2 User Authentication and Navigation

As a new user is added to the hypertext system, the user must register himself to the System Authorization Server (SAS). According to the security policy of the hypertext system, the SAS administrator subjects the user to membership in roles using user-role authorization, sets the validity period (if necessity), and returns the credential to the user. Suppose a user first wants to access the root document of one site in the hypertext system. The LAS of the site operates the user authentication using the credential of the user, and checks whether his validation period has expired. If the above processes are approved, the access control function of the LAS is performed to ensure that only authorized information is released.

In the following, let us consider the situation where users navigate the information space in the hypertext system. When a user traverses a link to another node at a different site, since the identity of the user has been verified when entering the hypertext system, the LAS of the local site only transmits the roles list granted for the user to the LAS of another site, and the user does not need to authenticate himself at the other sites. Therefore, the LAS of the other sites can also determine the permission of the user according to the roles list transmited from the previous LAS.

7. Maintenance of Authorizations

In this section, we are concerned with the issue of the maintenance and management of authorizations. As the role set R and the RH set are needed to make add/delete, or modify operation. The SAS must notify and send the related messages to the LAS administrator of each site, and the LAS administrator grants or revokes the role-object authorizations on related roles or objects according to the related messages. It will cause the authorization system to be greatly altered. However, since roles in an enterprise or an organization are usually unchanged, this situation usually does not occur. That is, these roles have a long useful lifetime. Moreover, when users are joined to the system, the SAS authorizes the user as a member of roles by granting operations for user-role authorizations and assigning an account and credentials to the user. That is, only the SAS must be concerned with the problem, and the LAS at each site does not take care of it.

On the other hand, when contents or permissions for objects stored at some site are altered, the change of authorizations is only restricted to the site. The LAS administrator at each site is the entity that can generate a new document, delete an old document, modify a document, and execute the operations for granting/revoking role-object authorizations. Meanwhile, the amount for the administrator at a site may be not unique. In addition, as to the applications for intranets and digital libraries, users do not "own" the information [7,8], users authorized specific roles may own specific permission on the related information. Moreover, if general users have some requests for modifications on the information at sites in the hypertext system but they does not have the authorization, they may send e-mail messages to request the corresponding LAS administrator at the site storing the information, and the LAS administrator may determine whether the request is acceptable or not. If the request is accepted, the LAS administrator completes the request.

As noted in Section 2, several approaches [10,12,13,16] have the management and maintenance problem springing from the changes of authorizations. For a large distributed hypertext system, according to the above discussions, our approach ensures the problem easy to resolve.

8. Conclusions and Future Work

In this paper, we have proposed a role-based authorization model for a large distributed hypertext system. We have extended and redefined some characteristics with definitions for RBAC model [7,18] adequate for the protection of information in hypertext systems. These characteristics include the object inheritance, the consideration of different kinds of data, and negative authorizations for denying roles to access objects.

Further works are in process that will ensure that our authorization model and system will be more complete. The development of a systematic approach for RBAC configuration design and analysis [15] is the first issue. Another issue is that several practical protocols in our authorization system should be integrated, i.e., the tranmission of credentials between the SAS and users, the transmission and authentication of credentials between users and the LAS, and the transmissions of roles list between sites. In our opinions, we may use protocol , such as Kerberos [14], to handle the latter issue.

References

[1] K. Andrews, F. Kappe, and H. Maurer, " Serving Information to the Web With Hyper-G," Proc. Third World-Wide Web Conf., pp. 919-926, 1995.

http://www.igd.fhg.de/www/www95/papers/105/hgw3.html.

[2] J. F. Barkley and A. V. Cincotta, "Role Based Access Control for the World Wide Web," NIST Report, 1997. http://hissa.nist.gov/rbac/.

[3] E. Bertino, C. Bettini, E. Ferrari, and P. Samarati, "A Temporal Access Control Mechanism for Database Systems," IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 1, pp. 67-80, 1996.

[4] S. Castano, M. G. Fugini, G. Martella, and P. Samarati, Database Security, Addison-Wesley,1995.

[5] D. E. Denning, Cryptography and Data Security, Addison-Wesley, Reading, 1982.

[6] D. Ferraiolo and D. R. Kuhn, "Role-based access controls," 15th NIST-NCSC National Computer Security Conference, pp. 554-563, Baltimore, MD, 1992.

[7] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role based access control: Features and motivations," 11th Annual Computer Security Applications Proceedings, IEEE Computer Society Press, 1995.

[8] E. Fox, M. Akscyn, R. Furuta, and J. Leggett, eds., Communications of the ACM- Special Issue on Digital Libraries, Vol. 38, No. 4, 1995.

[9] K. Gronbaek and R. H. Trigg, eds., Communications of the ACM- Special Issue on hypermedia, Vol. 37, No. 2, 1994.

[10] J. Kahan, "A Distributed Authorization Model for WWW," Proc. INET'95 Conf., Honolulu,Hawaii, 1995. http://www.isoc.org/HMP/PAPER/107.

[11] C. E. Landwehr, "Formal models for computer security," ACM Computing Surveys, Vol. 13, No. 3, pp. 247-278, 1981.

[12] M. G. Lavenant and J. A. Kruper, "The Phoenix Project: Distributed Hypermedia Authoring," Proc. First World-Wide Web Conf., 1994.

http://www.cern.ch/PapersWWW94/j-kruper.ps.

[13] S. Lewontin and M. E. Zurko, "The DCE Web Project: Providing Authorization and Other Distributed Services to the World Wide Web," Proc. Second World-Wide Web Conf., 1994. http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Security

/lewontin/Web_DCE_Conf_94.html.

[14] B. C. Neuman and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks," IEEE Comm., Vol. 32, No. 9, pp. 33-38, 1994..

[15] M. Nyanchama and S. Osborn, "Access Rights Administration in Role-Based Security Systems," Database Security VIII: Status and Prospects, pp. 37-56, 1994.

[16] P. Samarati and E. Bertino, "An Authorization Model for a Distributed Hypertext System," IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 4, pp. 555-562, 1996.

[17] R. Sandhu, E. J. Coyne, and C. E. Youman, eds., Proceedings of the First ACM Workshop on Role Based Access Control, ACM, 1996.

[18] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role based access control models," IEEE Computer, Vol. 29, No. 2, pp. 38-48, 1996.

[19] S. H. von Solms and I. van der Merwe, "The management of computer security profiles using a role-oriented approach," Computers&Security, Vol. 13, No. 8, pp. 673-680, 1994.