Preventing Conflict Situations During Authorization

SYLVIA ENCHEVA
Stord/Haugesund University College
Department Haugesund
Bjørnsonsg. 45, 5528 Haugesund
NORWAY
sbe@hsh.no

SHARIL TUMIN
University of Bergen
IT-Dept.
P. O. Box 7800, 5020 Bergen
NORWAY
edpst@it.uib.no

Abstract: Computer-based access control systems working with financial and privacy issues are concerned with access control policies. Structuring authorizations turns out to be of a key importance in a case of collaborating organizations.

Key–Words: Computer-based access control systems

1 Introduction

Access control to information systems is a fundamental management responsibility. Access control determines which resource in a system a legitimate user can access. A sophisticated and complex control implies structured authorizations.

The complexity of security administration appears to be among most challenging problems in managing large networks. Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise [31]. Role-based access control assists users is completing various authorized tasks by flexible managements of their actions.

The first draft of a consensus standard for RBAC was proposed in [44] and the reference model and functional specification were discused in [17].

In this work we consider collaborating organizations allowing their users to access resources outside of their home organizations.

The rest of the paper is organized as follows. Related work, basic terms and concepts are presented in Section 2 and Section 3 respectively. The management model is described in Section 4. The system architecture is discussed in Section 5. The paper ends with a description of the system implementation in Section 6 and a conclusion in Section 7.

2 Background

A formal model of role based access control (RBAC) is presented in [16], [40], [42] and [41]. Permissions in RBAC are associated with roles, and users are made members of appropriate roles, thereby acquiring the roles' permissions. The RBAC model defines three kinds of separation of duties - static, dynamic, and operational. Separation of duties was discussed in [9], [17] and [46]. A framework for modeling the delegation of roles from one user to another is proposed in [4]. A multiple-leveled RBAC model is presented in [12]. The design and implementation of an integrated approach to engineering and enforcing context constraints in RBAC environments is described in [34], [47] and [48]. Secure information flow is described in [14] and [39]. Intrusion detection in RBAC-administered databases is described in [8]. The approach is based on the assumption that Database Management Systems (DBMS) require a high degree of assurance and security and, hence, they have well defined usage and access control policies in place.

While RBAC provides a formal implementation model, Shibboleth [26] defines standards for implementation, based on OASIS Security Assertion Markup Language (SAML). Shibboleth defines a standard set of instructions between an identity provider (Origin site) and a service provider (Target site) to facilitate browser single sign-on and attribute exchange.

3 Preliminaries

The semantic characterization of a four-valued logic for expressing practical deductive processes is presented in [7]. In most information systems the management of databases is not considered to include neither explicit nor hidden inconsistencies. In real life situation information often come from different contradicting sources. Thus different sources can provide inconsistent data while deductive reasoning may result in hidden inconsistencies. The idea in Belnap's approach is to develop a logic that is not that depend-

Table 1: Truth table for the operation $\sim$ in Belnap's logic

$\sim$	T	F	B	N
	F	T	N	B

Table 2: Truth table for the operation $\vee$ in Belnap's logic

$\vee$	T	F	B	N
T	T	T	T	T
F	T	F	B	N
B	T	B	B	T
N	T	N	T	N

able of inconsistencies. The Belnap's logic has four truth values 'T, F, Both, None'. The meaning of these values can be described as follows:

- an atomic sentence is stated to be true only (T),
- an atomic sentence is stated to be false only (F),
- an atomic sentence is stated to be both true and false, for instance, by different sources, or in different points of time (Both), and
- an atomic sentences status is unknown. That is, neither true, nor false (None).

The truth values of various formulas using Belnap's logic may be obtained by applying rules described in tables 1, 2, and 3.

A partial logic assigns both a truth and a falsity extension to a predicate such that they need not be the set-theoretic complement of each other. In partial logic, there is a strong negation ('Kleene negation') and a weak negation proposed by Lukasiewicz. While strong negation represents explicit falsity, weak negation represents non-truth.

Another four-valued logic is the Nelson's constructive logics [1], [21], [35], [36], [38], [49], and [52]. Nelson's constructive four-valued logic is a logic with strong negation.

A user is defined as a valid domain identity at a particular organization $O_i, i = 1, 2, 3, 4, 5, 6, 7, 8, 9$.

Table 3: Truth table for the operation $\wedge$ in Belnap's logic

$\wedge$	T	F	B	N
T	T	F	B	N
F	F	F	F	F
B	B	F	B	F
N	N	F	F	N

A group is a set of users. A resource defines a set of protected Web objects. A permission defines a right of a user to perform an action on a resource. An authorization gives a set of permissions to a user to execute a set of operations on a specific set of resources.

A closed policy permits specification of only positive authorizations and allows only those accesses that are explicitly authorized. An open policy permits specification of only negative authorizations and allows only those accesses that are not explicitly denied [10].

4 Scenario

In this scenario we consider collaborating organizations using resources owned by some of these organizations Fig. 1. In particular we assume that three resources placed at different organizations can be accessed by group members of various organizations.

Any group is administered by a resource manager affiliated with the corresponding organization, where resource managers of groups A_i apply closed policy and resource managers of groups B_j apply open policy.

Possible conflict situations:

- a user belongs to group A_i and at the same time belongs to group B_1 or group B_j, or
- another user may be affiliated with several organizations and belongs to several groups.

Such conflicts can be avoided by use of many valued logic.

Applying the truth tables for Belnap's logic we propose the following:

- A user belongs to group A and does not belong to group B. A user is authorized to access any of the resources $R_m, m = 1, 2, 3$ if that particular

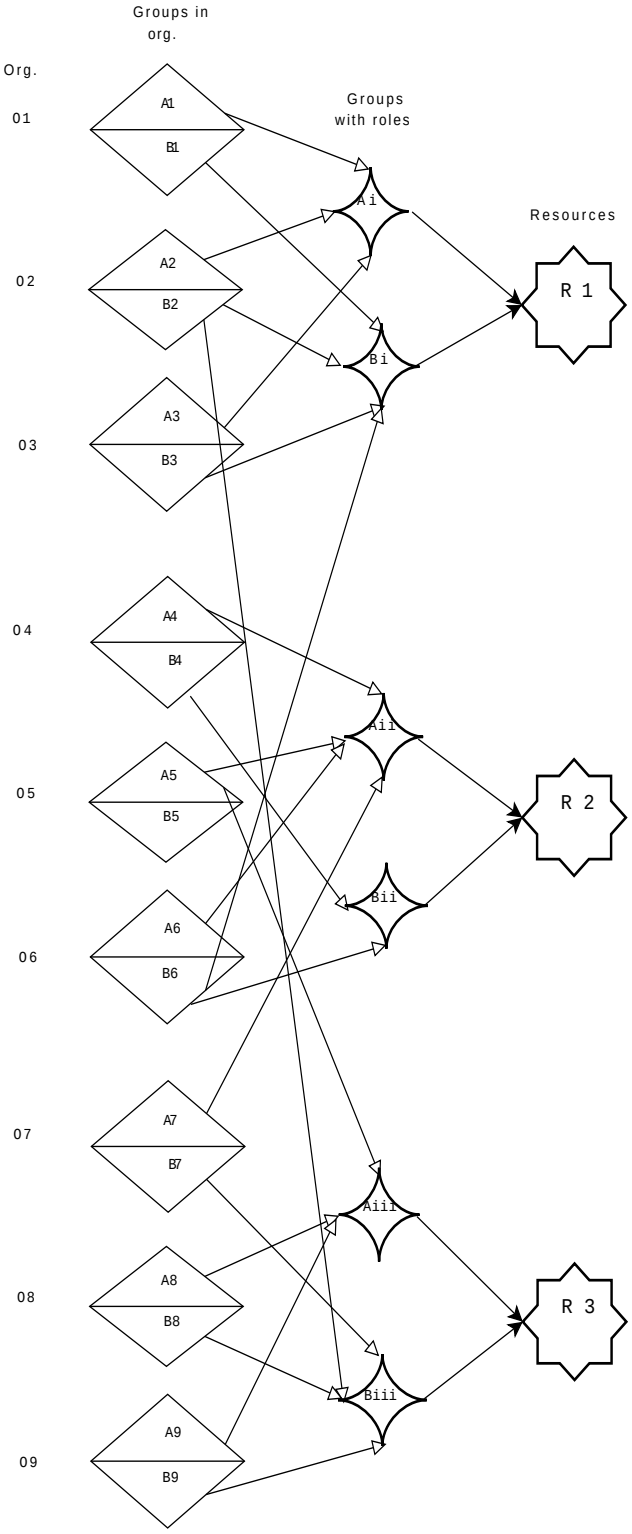

Figure 1: Authorizations for users

user belongs to a group A_l authorized to access that particular resource and does not belong to any of the groups $B_j, j = 1, 2, 3, 4, 5, 6$.

- A user is not automatically authorized to access a resource if the user belongs to a group A_i and to a group B_j. The user membership must be first considered by the corresponding resource managers and an individual authorization may be granted afterwards.

- A user that does not belong any of the groups A_i or B_j is authorized to access a resource, if he/she belongs to at least one of the organizations applying open policy.

- A user does not belong to any group A and belongs to a group B. A user is not authorized to access a resource if that user belongs to group B_j and does not belong to any of the groups A_i.

5 System Model

In our model, collaborating organizations administer their own users and resources independently of each other. Let define the resource owner organization as the publisher organization (PO) and an organization that uses the shared resources as a subscriber organization (SO). The users of a Web resource belong to a SO and the server that provides the shared Web resource, for example a Web-based application, belongs to a PO. In our collaboration model, organizations have references of each other and their dynamic relations depend on the roles their users play. One and the same organization plays both the publisher organization and subscriber organization if both the user and the Web resource belong to the same organization.

Each subscriber organization administers its own domain users. Using delegation of responsibility the user group memberships for a particular subscribed resource are done by sub-units, for example a faculty or a department, that belong to that particular subscriber organization. To each of the subscribed shared protected resources are assigned two groups, namely, group A that applies close policy on that particular resource and group B that applies open policy on that same resource. In another words, group A implements a white list on the resource while group B implements a black list on the resource. A protected resource will be treated as either a close or an open resource depending on collaboration relationships among participating organizations. The admin-

istrations of these groups are done by independent groups of administrators assigned by central domain administrators through delegation of responsibility.

Applications at the publisher organization will make use of these two groups from different subscriber organization to do access control on the particular protected resource. The two groups will be represented as a list of pairs {SO-domain, URI} in a database table named subscriber organization table (SOT) at the publisher organization where the value of SO-domain is the subscriber organizations' domain name as defined by the Domain Name System (DNS) and the value of URI is the Uniform Resource Identifier (URI) that points to a Web application address from which the publisher organization's access controller can make queries about group membership and thus access rights to a particular resource, of a user from a particular subscriber organization specified in SO-domain. The domain name consists of two or more globally accepted parts separated by dots, for example 'uib.no' and 'hsh.no'. A machine name 'tulipan' will then have a fully qualified DNS name 'tulipan.uib.no' which is a different machine from the one with name 'tulipan.hsh.no', even though both machines locally are known as 'tulipan'. An example URI of an application is

'http://zues:xMo12G@tulipan.uib.no\getGroups'.

Note that this URI will be used inside a resource control program for obtaining user groups membership for a particular protected resource. The example URI also shows the use of simple authentication mechanism that protects the 'getGroups' application at 'tulipan.uib.no'. In addition to SOT, the publisher organization also needs to maintain a resource names and resource policies table (RPT). For example a resource 'math-1' with a default closed policy will have an entry in the RPT as a pair of

{resource-name, default-group-type}

for example ['math-1', 'A'].

Each subscriber organization will provide local users and groups management application. Users and groups are maintained in a database tables. Users table will be used for authentication process while groups table will be used for authorization process. Groups table in the database contains access control table (ACT) for each resource they wish to subscribe. Each resource is referred to by a pair {name, PO-domain} and will have two groups, referred by a triplet {A, name, PO-domain} for close policy and {B, name, PO-domain} for open policy. For example, a user at a particular subscriber organization 'uib.no', with a user identifier 'edpst' is a member of the closed policy group A of a resource named 'math-1' published by domain 'hsh.no', will then have an entry in ACT, ['edpst', 'A', 'math-1', 'hsh.no']. Data entries in ACT will determine weather a user of a particular subscriber domain has access right to a particular resource at a particular publisher domain.

The publisher organization needs to publish all shared resources the organization is willing to share. The potential subscriber organizations will consult a well known URI where the resources are published by a particular provider. This can be accomplished by providing descriptions of all resources in an Extensible Markup Language (XML) document at an URI collectively agreed before hand by all partners, for example, 'http://subscriber:gogetit@mat.hsh.no\resources'.

All subscribers can consult this URI from a particular publisher organization for the XML file at any time for information pertinence to all published resources from a particular publisher organization, which then can be used to initialize or update information in subscriber organizations' databases represented in ACT.

Within this model, both publishers and subscribers need some form of communication framework in order to communicate user identities and authorizations to control access on shared Web resources. There are many ways of providing these services, where among most common ones are, XML remote procedure (XML-RPC), Java based remote method invocation (RMI) and Simple Object Access Protocol (SOAP). We propose a simpler mechanism inspired by Representational State Transfer (REST). Compare to the other mechanism, HTTP queries are much simpler because they do not define its own transport protocol. HTTP query depends on generic Web interface of HTTP GET, POST, PUT and DELETE. In order to provide and utilize Web services one needs only to deploy Web server and client that support XML documents formatting and parsing.

6 System Implementation

Both subscriber and publisher organizations need to implement Web portals through which a user will be authenticated and then authorized for a particular protected Web resource. These portals can be implemented using a standard Web-based applications server architecture with a Web server front-end to interface with users, a programmable middleware runtime environment and a database back-end to store temporary and persistence data.

The three-tier architecture implemented in our prototype system is composed of: Presentation Tier - Apache Web server, Application Tier - Python pro-

gramming environment, and Data Tier - Oracle EX.

We propose using Apache Web server for implementing the presentation tier. Apache Web server provide modular framework where different module can be implemented separately. Two important modules are mod_ssl and mod_python. Transport layer security is implemented by mod_ssl and programmable runtime environment is provided by mod_python which implement Python interpreter internally link to the Web server.

Data submitted by users from their Web browsers through Web queries trigger processing events in the application tier. Depending on a particular URI, programs or scripts in the application tier will be executed. The GET parameters will be used as input parameters to these programs. The programs will then provide response or error messages in the form of Web pages. The application server will be programmed using Python scripting language with the help of mod_python. All the Apache's application programming interfaces (API) are directly connected to the Python runtime environment, which makes Web application programming simpler.

Data store for the system is implemented at the data tier. Information is stored into and retrieved from a back-end relational database management system (RDBMS). Oracle XE is a free small RDBMS from Oracle which support Structured Query Language (SQL) standard. Using a free and small RDBMS from Oracle, make it easy to upgrade the database to enterprise level, if that need arises. The database stores long term persistence information about users, groups, resources and applications states. Data saved in the database store activities history of users and can affect future constrains of users access behavior of the whole systems within the cooperative framework. Applications initialize and modify information in the database. There are many Python Oracle modules that integrate the application tier to the data tier in the system. We propose using DCOracle2 module.

To support cross-domain authentication and authorization mechanism between a user from subscriber organization and protected resource from publisher organization the implemented framework must provide support for URL redirections and client cookies. Signed client cookies are used to store session information in the users' Web browser. URL redirect mechanism is used to redirect user to proper portal for logon process. There are several ways to implement URI redirection. A Web browser can be redirect to another URI different form the original URI by:

1. Meta refresh -
<meta http-equiv="refresh" content="0;url=https://siam.uib.no'\logon">

2. Frame redirect -
<frame name="redirect" src= https://siam.uib.no'/logon"<> \frame>

3. Header redirect -
HTTP\1.1 200 OK ... Location: https://siam.uib.no'\logon

Client cookies are used to preserve state across otherwise stateless communication between client and server using HTTP communication. The server send set cookies in respond header to the client, for example:

Set-Cookie: admin=edpst; expire=Fri, 11-Apr-2008 12:05:00;
path=\app; domain=tulipan.uib.no; secure

The next time the client requesting a resource from the server and if all the constraints are met the cookie 'admin' will be written back to the server as a part of HTTP request, for example:

GET \app HTTP\1.0
From: edpst@it.uib.no
User-Agent: HTTPTool1.2
Cookie: admin=edpst

An unauthenticated user will first be redirect to her SO portal to be authenticated. After a valid credential is presented the user will receive a session token by the logon application and be redirect back to the PO portal. Using the session token given previously by the SO portal, the PO portal will then consult SO portal for user identification for the user that owns the session token. Then PO portal will then create publisher organization session token for the user and redirect the Web browser back to the application originally requested by the user.

After that a user is authenticated at her SO portal, the PO portal applications can consult SO portal regarding users' session, identification and groups membership by simply sending HTTP queries with parameters, for example,

'http://zues:xMo12G@tulipan.uib.no/groups?
user=edpst'

which the SO portal will reply with XML formatted response message containing group membership of user with identifier 'edpst' defined at domain 'uib.no'. By providing Web services architecture using simple HTTP queries we believe that the system will be easy to implement and maintain. Any Web

tools can be used, if they collectively support HTTP, XML and SQL.

Basic authentication scheme is used when a client connects to a server. This simple mechanism is a part of security measure to the system. The basic authentication is implemented by the Web server and it is easy to deploy. In addition of host address access control, this provides us with good enough security measure for the system described in this paper. Thus, any client that knows the basic authentication parameters and belongs to a list of clients on a server can make use of the services provided by the server.

A typical example of XML reply document for a HTTP request

'http://zues:xMo12G@tulipan.uib.no\getGroups'

is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<zebra rows="2" reply="user"
hash="537851618991316fae7bd23b0b03adc4"
signed="yes">
<id>Ver2.06 GetAgent@mat.hsh.no
\#12</id>
<ts>20080304170853</ts>
<user>
<id>edpst</id>
<domain>uib.no</domain>
<group>
<type>A</type>
<valid>20081013120000</valid>
<resource>
<name>math-1</name>
<domain>hsh.no</domain>
</resource>
</group>
<group>
<type>A</type>
<valid>20081013120000</valid>
<resource>
<name>alg-2</name>
<domain>hsh.no</domain>
</resource>
</group>
</user>
<user>
<id>sbe</id>
<domain>uib.no</domain>
<group>
<type>A</type>
<valid>20090101120000</valid>
<resource>
<name>math-1</name>
<domain>hsh.no</domain>
</resource>
</group>
<group>
<type>A</type>
<valid>20081013120000</valid>
<resource>
<name>alg-2</name>
<domain>hsh.no</domain>
</resource>
</group>
<group>
<type>A</type>
<valid>20080603120000</valid>
<resource>
<name>logic-1</name>
<domain>hsh.no</domain>
</resource>
</group>
</user>
</zebra>
```

The requester needs to trust that the reply message comes from the server and that the XML message has not been tempered with during transit. We propose in using hash functions and private\public keys encryption for signing the calculated hash value with sender private key to provide this trust. When the reply messages arrives the requester can then calculate its own hash values and verify it using public key of the sender. The hash is calculated for the block within the <zebra>, < \zebra> tags.

7 Conclusion

The proposed model simplifies user management in cooperating organizations by generating a group for a single role. Users and group data are shered among organizations via a common communication mechanism. This way the risk of illegal access is minimized.

References:

[1] A. Almukdad and D. Nelson, "Constructable falsity and inexact predicates", *Journal of Symbolic Logic*, Vol. 49, pp. 231-233, 1984.

[2] M. Al-Kahtani and R. Sandhu, "Rule-based RBAC with negative authorization", *20th Annual Computer Security Applications Conference*, Arizona, 2004.

[3] M. Andress, "Access control", *Information security magazine*, April, 2001.

[4] E. Barka and R. Sandhu, "Role-based delegation model/ hierarchical roles", *20th Annual Computer Security Applications Conference, Arizona*, 2004.

[5] J. Barkley, K. Beznosov, and J. Uppal, "Supporting relationships in access control using Role Based Access Control", *Fourth ACM Workshop on Role-Based Access Control*, 1999.

[6] N. J. Belnap, " How a computer should think", *In Contemporary Aspects of Philosophy. Proceedings of the Oxford International Symposia*, Oxford, GB, pp. 30-56, 1975.

[7] N. J. Belnap, "A useful four valued logic", *Modern uses of multiple-valued logic*, J.M. Dunn and G. Epstain (eds), D. Reidel Publishing Co., Dordrecht, pp. 8-37, 1977.

[8] E. Bertino, A. Kamra, E. Terzi, and A. Vakali, "Intrusion Detection in RBAC-administered Databases", http://www.acsac.org/2005/papers/127.pdf (2005)

[9] E. Bertino, P.A., Bonatti, and E. Ferrari, "TRBAC: A temporal Role-Based Access Control model", *ACM Transactions on Information and System Security*, Vol. 3, No. 3, pp. 191-223, 2001.

[10] E. Bertino, S. Jajodia, P. Samarati, "A Flexible Authorization Mechanism for Relational Data Management System", *ACM Transactions on Information Systems*, Vol. 17, No. 2, pp. 101-140, 1999.

[11] R. Bhatti, E. Bertino, A. Ghafoor, and J. B. D. Joshi, "XML-based specification for Web services document security", *IEEE Computer*, Vol. 37, No. 4, 2004.

[12] S-C. Chou, "L^nRBAC: A multiple-levelled Role-Based Access Control model for protecting privacy in object-oriented systems", *Journal of Object Technology*, Vol. 3, No. 3, pp. 91-120, 2004.

[13] B. A. Davey, and H. A. Priestley, Introduction to lattices and order, Cambridge University Press, Cambridge, 2005.

[14] D. Denning, "A lattice model of secure information flow", *Communications of the ACM*, Vol. 19, No. 5, 1976.

[15] J. Dowling and V. Cahill, "Self-managed decentralised systems using K-components and collaborative reinforcement learning", *Proceedings of the Workshop on Self-Managed Systems (WOSS'04)*, pp. 41-49, 2004.

[16] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-Based Access Control (RBAC): Features and motivations", *1995 Computer Security Applications Conference*, pp. 241-248, 1995.

[17] D. Ferraiolo, R. Sandhu, S. Gavrila, R.D. Kuhn, and R. Chandramouli, "Proposed NIST standard for Role-Based Access Control", *ACM Transactions on Information and System Security*, Vol. 4, No. 3, pp. 224-274, 2001.

[18] D. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-Based Access Control*, Artech House, Computer Security Series, 2003.

[19] S. Gavrila and J. Barkley, "Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management", *Third ACM Workshop on Role-Based Access Control*, 1998.

[20] T. Guerin and R. Lord, "RBAC identity management", http://www.portalsmag.com/articles/default.asp?ArticleID=4923, 2003.

[21] Y. Gurevich, "Intuitionistic logic with strong negation", *Studia Logica*, Vol. 36, pp. 49-59, 1977.

[22] K. Harrison-Broninski and F. Hayden, "Role-based transection management in collaborative systems", http://www.rolemodellers.com/abstracts/Role-Based %20Transaction%20Management %20In%20Collaborative% 20Systems.pdf

[23] T. Hildmann and J. Barholdt, "Managing trust between collaborating companies using outsourced role based control", *4th ACM Workshop on RBAC*, pp. 105-111, 1999.

[24] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure, Or: Assigning roles to strangers", *IEEE Symposium on security and privacy*, 2000.

[25] http://dev.mysql.com

[26] http://shibbolethinternet2.edu

[27] http://www.apache.org

[28] http://www.firebirdsql.org

[29] http://www.openssl.org

[30] http://www.postgresql.org

[31] http://whatis.techtarget.com/definition/0, sid9_gci1166529,00.html

[32] http://www.xml-rpc.org

[33] B. Kropp and M. Gallaher, "Role-based access control systems can save organizations time and money", *Information Security Magazine*, 2005.

[34] A. Mattas, I. Mavridis, C. Ilioudis, and I. Pagkalos, "Dynamically Administering Role Based Access Control", *WSEAS Transaction on Information Science & Applications*, Vol. 3, No. 10, pp. 1777-1784, 2006.

[35] D. Nelson, "Constructible Falsity", *Journal of Symbolic Logic*, Vol. 14, pp. 16-26, 1949.

[36] S. Odintsov, "Algebraic Semantics for Paraconsistent Nelson's Logic", *Journal of Logic and Computation*, Vol. 13, pp. 453-468, 2003.

[37] S. Odintsov and H. Wansing, "Constructive Predicate Logic and Constructive Modal Logic. Formal Duality versus Semantical Duality", in: V. Hendricks et al. (eds.), *First-Order Logic Revisited*, Berlin: Logos Verlag, pp. 269-286, 2004.

[38] R. Routley, "Semantical Analyses of Propositional Systems of Fitch and Nelson", *Studia Logica*, Vol. 33, pp. 283-298, 1974.

[39] R. Sandhu, "Lattice-Based access control models", *IEEE Computer*, Vol. 26, No. 11, 1993.

[40] R. Sandhu, "Transaction control expressions for separation of duties", *4th aerospace computer security conference*, pp. 282-286, 1988.

[41] R. Sandhu, "Separation of duties in computerized information systems", *IFIP WG11.3 Workshop on database security*, 1990.

[42] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models", *IEEE Computer*, Vol. 29, No. 2, pp. 38-47, IEEE Press, 1996.

[43] R. Sandhu, "Role activation hierarchies", *3rd ACM Workshop on RBAC*, pp. 33-40, 1998.

[44] R. Sandhu, D. Ferraiolo, R. Kuhn, "The NIST model for role-based access control: Towards a unified standard", *In proceedings of 5th ACM Workshop on Role-Based Access Control*, pp. 47-63, Berlin, Germany, July, 2000.

[45] S. Schwoon, S. Jha, T. Reps, and S. Stubblebine, "On generalized authorization problems", *Proceedings of 16th IEEE Computer Security Foundations Workshop*, Asilomar, Pacific Grove, CA, pp. 202-218, 2003.

[46] R. Simon and M. Zurko, "Separation of duty in role-based environments", *In Proceedings of 10th IEEE Computer Security Foundations Workshop*, Rockport, Mass., pp. 183-194, 1997.

[47] M. Strembeck, "Conflict checking of separation of duty constraints in RBAC-implementation experiences", http://wi.wu-wien.ac.at/home/mark/publications/se2004.pdf

[48] M. Strembeck and G. Neumann, "An integrated approach to engineer and enforce context constraints in RBAC environments", *ACM Transactions on Information and System Security*, Vol. 7, No. 3, pp. 392-427, 2004.

[49] R. Thomason, "A Semantical Study of Constructive Falsity", *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, Vol. 15, pp. 247-257, 1969.

[50] A. Topchy, and W. Punch, "Dimensionality reduction via genetic value clustering", GECCO-2003, *Lecture Notes in Computer Science*, Vol. 2724, pp. 1431-1443, Springer-Verlag, Berlin Heidelberg New Jork, 2004.

[51] G. Wagner, "Vivid Logic: Knowledge Based reasoning with two kinds of negation", *Lecture Notes in Artificial Intelligence*, Vol. 764, Springer-Verlag, Berlin Heidelberg New York, 1994.

[52] H. Wansing, "Negation", in: L. Goble (ed.), *The Blackwell Guide to Philosophical Logic*, Cambridge, MA: Basil Blackwell Publishers, pp. 415-436, 2001.

[53] Haibo Yu, Qi Xie, and Haiyan Che, "Description Logic Based Conflict Detection Methods for RB-RBAC Model", *IJCSNS International Journal of Computer Science and Network Security*, Vol. 6, No. 1A, 2006.

[54] X. Zhang, J. Park, and R. Sandhu, "Schema based XML security: RBAC approach", *Seventeenth IFIP 11.3 Working Conference on Data and Application Security*, Estes Park, Colorado, USA, August 4-6, 2003.