Fuzzy Role-Based Access Control

Carles Martínez-García[a,*], Guillermo Navarro-Arribas[b], Joan Borrell[a]

[a] *Department of Information and Communications Engineering (dEIC), Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain*
[b] *IIIA, Institut d'Investigació en Intel·ligència Artificial - CSIC, Consejo Superior de Investigaciones Científicas, Campus UAB s/n, 08193 Bellaterra, Spain*

Abstract

RBAC (Role-Based Access Control) is a widely used access control model, which reduces the maintenance cost of classical identity-based access control. However, despite the benefits of RBAC, there are environments in which RBAC can hardly be applied. We present FRBAC (Fuzzy Role-Based Access Control), a generalization of RBAC that fits the requirements of environments where authorization-related information is vague. Moreover, FRBAC deals with environments where the actions that can be executed over the resources have a fractional meaning, as data lying in databases and QoS-subjected operations. FRBAC generalizes RBAC through the use of fuzzy relations.

Keywords: Safety/security in digital systems, Role-Based Access Control, Uncertainty

1. Introduction

Role-Based Access Control (RBAC) [12, 5, 6] is widely used in corporate environments with many advantages, but it presents some problems and impose some constraints in some concrete environments [13]. We extend the RBAC model to cope with two types of scenarios (although others could also be accommodated). The former type covers all scenarios where the authorization-related information is imprecise. One example is the Aware Home project [3] where the level of accuracy of authenticating sensors is not perfect. Another example can be the multi-domain environment, where a dynamic trust relation [2] may exist between the administrative domains and thus the trust relation affects the way in which the users interact with the systems. In those scenarios, the imprecision must be propagated through the system to the access decision. The latter type of scenarios are those where the actions have a fractional meaning and it makes no sense to permit or not its execution but to permit the execution to a given degree. For instance, data lying in databases, where the responses to queries are modified in order to add a certain percentage of noise [18]. A second example can be those environments subjected to QoS (Quality of Service) [11] restrictions.

The aim of this paper is to introduce Fuzzy Role-Based Access Control (FRBAC) as a generalization of RBAC. It relies in the fuzzy user-role and role-permission assignments. These fuzzy assignments allow to deal in a natural way with imprecise information and propagate it through the user-permission relation to the access decision. The access decision can be formulated with a fractional meaning or it can be defuzzified in order to deal with permissions that only have a binary sense.

Fuzzy relations, and fuzzy concepts in general, have been used to extend RBAC [17, 15, 10], and other access control models [16, 8, 14]. We provide a novel approach by considering the fuzziness of the RBAC model itself rather than adding fuzzy concepts on top of a traditional RBAC model contributing with a more clear and generic definition. Our work builds on previous ideas which aim to add a flexibility to traditional access control models [7, 9, 1]. We are not aware of any proposal based on the ideas of RBAC which provides such flexibility in the user-role and role-permission assignment.

2. FRBAC

We introduce FRBAC departing from the RBAC definition of the standard in [4], which evolved to [5]. The standard divides the RBAC model into three parts: Core RBAC, which includes the basic functionality; Hierarchical RBAC, extending the Core with role hierarchies; and Constrained RBAC, incorporating separation of duties constraints. Analogously, we introduce Core FRBAC, Hierarchical FRBAC, and Constrained FRBAC. We do not consider the notion of sessions included in the RBAC standard in order to ease the understanding of the scheme, however it should be easy to export the concept of sessions to FRBAC.

We use the following notation and definitions (as described in the RBAC definition from [5]).

- $USERS$ is a set of users.
- $ROLES$ is a set of roles.
- OBS is a set of resources (objects).
- OPS is a set of operations.
- $P = 2^{(OBS \times OPS)}$ is a set of permissions.

*Corresponding author.
Email addresses: carlos.martinez@uab.cat (Carles Martínez-García), guille@iiia.csic.es (Guillermo Navarro-Arribas), joan.borrell@uab.cat (Joan Borrell)

- $UA \subseteq USERS \times ROLES$ is a set of user-role assignments.
- $PA \subseteq PRMS \times ROLES$ is a set of role-permission assignments.

2.1. Core FRBAC

The foundations of FRBAC are the user-role and the role-permission assignments defined through fuzzy relations of the form:

- $UA : USERS \times ROLES \rightarrow [0, 1]$
- $PA : ROLES \times PRMS \rightarrow [0, 1]$

That is, there is a mapping relating users with roles and another mapping relating roles with permissions. The user-role mapping is a set of items of the form $((u, r), \mu_{UA}(u, r))$ where $u \in USERS$, $r \in ROLES$, and $\mu_{UA}(u, r)$ is a function that returns the user-role relation strength. The strength is valued in the real unit interval $[0, 1]$. The role-permission relation has an analogous form.

Given the UA relation, we define the following functions in order to compute the user-role relation for a given user:

- $user\_roles(u : USERS) \rightarrow R_u$, where $R_u = \{(r_i, \mu_{R_u}(r_i))\}$, $r_i \in ROLES$ and $\mu_{R_u}(r_i) \rightarrow [0, 1]$. That is, given a user u, the function $user\_roles(u)$ returns a fuzzy set containing the roles (r_i) assigned to the user u as well as the strength of the assignment ($\mu_{R_u}(r_i)$). The function is described as follows:

$$user\_roles(u) = \{(r_i, \mu_{UA}(u, r_i))| \\ ((u, r_i), \mu_{UA}(u, r_i)) \in UA\}$$

Given the PA relation, the role-permission function has a similar form:

- $role\_permissions(r : ROLES) \rightarrow P_r$, where $P_r = \{(p_i, \mu_{P_r}(p_i))\}$, $p_i \in PRMS$ and $\mu_{P_r}(p_i) \rightarrow [0, 1]$. That is, given a role r, the function $role\_permissions(r)$ returns a fuzzy set containing the permissions assigned to the role as well as the strength of the assignment. The function is described as follows:

$$role\_permissions(r) = \{(p_i, \mu_{PA}(r, p_i))| \\ ((r, p_i), \mu_{PA}(r, p_i)) \in PA\}$$

Once computed, the user-role and role-permission relation, the user-permission assignment can be derived. The user-permission relation is defined as the max-min composition of the UA and the PA relations. We formally define the following function:

- $user\_permissions(u : USERS) \rightarrow P_u$, where $P_u = \{(p_i, \mu_{P_u}(p_i))\}$, $p_i \in PRMS$ and $\mu_{P_u}(p_i) \rightarrow [0, 1]$. That is, given a user u, the function $user\_permissions(u)$ returns a fuzzy set containing the permissions assigned to the user as well as the strength of the assignment. The function is described as follows.

$$user\_permissions(u) = \\ = \bigcup_{(r_i, \mu_{R_u}(r_i)) \in user\_roles(u)} \{(p_i, \min(\mu_{P_r}(p_i), \mu_{R_u}(r_i))) | \\ (p_i, \mu_{P_r}(p_i)) \in role\_permissions(r_i)\}$$

where the union operand $\cup$ stands for:

$$(x, \mu_A(x)) \cup (y, \mu_A(y)) = \\ = \begin{cases} \{(x, \mu_A(x)), (y, \mu_A(y))\}, & \text{if } x \neq y \\ \{(x, \max(\mu_A(x), \mu_A(y)))\}, & \text{if } x = y \end{cases}$$

At access decision time, the response is subjected to find a user's privilege which allows the execution of the given action over the given resource. For those scenarios where the access has a fractional meaning, we define the following function:

- $access : USERS \times OPS \times OBS \rightarrow [0, 1]$. That is, given a user u, an operation op and an object obj, the $access$ function returns the access degree that the user u has over the resource obj through the operation op. The function is described as follows:

$$access(u, op, o) \Rightarrow \{\mu_p | (p, \mu_p) \in user\_permissions(u), \\ (op, o) \in p\}$$

The enforcement point must guarantee that the requested action is executed (if permitted) under the execution parameters represented by the access decision strength.

Of course, there are scenarios where the applicability of the operations over resources is binary: the action is entirely executed or it is not. We define a security threshold δ in order to defuzzificate the access decisions, so a permission is applicable only if the decision strength is equal or greater than δ. This threshold states the maximum imprecision level that the system is willing to tolerate. The given semantics of the security threshold are imposed by the application itself and the meaning of the UA and PA relations (see Section 3). The access function is then redefined as:

- $access_\delta : USERS \times OPS \times OBS \rightarrow BOOLEAN$. That is, given a user u, an operation op and an object obj, the $access$ function returns a boolean values which dictates whether the user u is allowed to execute the action op over the resource obj. The function is described as follows:

$$access_\delta(u, op, o) \Rightarrow access(u, op, o) \geq \delta$$

2.2. Hierarchical FRBAC

The RBAC standard defines a hierarchical relation between roles. Given a role r_1, the set of users belonging to the role (U_{r_1}) and the set of permissions assigned to the role (P_{r_1}), a role r_2 is a junior role of r_1 if $U_{r_1} \subseteq U_{r_2}$ and $P_{r_2} \subseteq P_{r_1}$. That is, the permissions of the senior role are inherited from the permissions of the junior role, and the users of the senior roles also belong to the users of the junior roles. The inheritance relations between the roles are specified as a partial order.

2.2.1. Crisp inheritance

The RBAC standard describes a partial order RH which defines inheritance relations between the roles:

- $RH \subseteq ROLES \times ROLES$

In practice, the roles that a user has are those explicitly assigned to the user through the UA assignments and the roles implicitly assigned to the user through the inheritance relation RH. We define in the FRBAC model the following inheritance function over the RH relation:

- $inherited(r : ROLES) \rightarrow 2^{ROLES}$. That is, given a role r, the function $inherited(r)$ returns the role set containing the junior roles of r. The function is defined as follows:

$$inherited(r) = \{r_i \mid (r, r_i) \in RH\}$$

The user-role strength of an inherited role must be subjected to the user-role strength of its corresponding senior role. Analogously to Core FRBAC, the user-role and the user-permission assignment functions under the presence of role inheritance are defined as:

- $user\_roles_{\downarrow}(u : USERS) \rightarrow R_u$:

$$user\_roles_{\downarrow}(u) = \{(r_i \cup inherited(r_i), \mu_{UA}(u, r_i))| \\ ((u, r_i), \mu_{UA}(u, r_i)) \in UA\}$$

- $user\_permissions_{\downarrow}(u : USERS) \rightarrow P_u$:

$$user\_permissions_{\downarrow}(u) = \\ = \bigcup_{(r_i, \mu_{R_u}(r_i)) \in user\_roles_{\downarrow}(u)} \{(p_i, \min(\mu_{P_r}(p_i), \mu_{R_u}(r_i)))| \\ (p_i, \mu_{P_r}(p_i)) \in role\_permissions(r_i)\}$$

2.2.2. Fuzzy inheritance

In FRBAC, the RH relation may be defined as a fuzzy relation of the form:

- $RH \subseteq ROLES \times ROLES \rightarrow [0, 1]$

In this case, the inheritance function is defined as follows:

- $inherited(u : USERS, r : ROLES) \rightarrow R_r$, where $R_r = \{(r_i, \mu_{R_r}(r_i))\}$, $r_i \in PRMS$ and $\mu_{R_r}(r_i) \rightarrow [0, 1]$. That is, given a role r and a user u, the function $inherited(u, r)$ returns a fuzzy set containing the junior roles of r as well as the inheritance strength, taking into account the strength of the UA assignment. The function is described as follows:

$$inherited(u, r) = \{(r_i, \min(\mu_{UA}(u, r), \mu_{RH}(r, r_i))| \\ ((r, r_i), \mu_{RH}(r, r_i)) \in RH, \\ ((u, r), \mu_{UA}(u, r)) \in UA\}$$

The inheriting magnitude of every inherited role is computed as the minimum between the inheritance strength and the user-senior role assignment.

The user-role function is slightly different than under a crisp inheritance:

- $user\_roles_{\downarrow}(u : USERS) \rightarrow R_u$:

$$user\_roles_{\downarrow}(u) = \{(r_i, \mu_{UA}(u, r_i)) \cup inherited(u, r_i)| \\ ((u, r_i), \mu_{UA}(u, r_i)) \in UA\}$$

2.3. Constrained FRBAC

The RBAC standard defines two types of separation of duties: Static Separation of Duties (SSD) and Dynamic ones (DSD). DSD are related to the usage of sessions and are out of the scope of this paper. SSD apply in RBAC in order to prevent users to be assigned to a role set which allows by itself to misuse the system. The standard defines the collection SSD as:

- $SSD \in (2^{ROLES} \times \mathbb{N}^+)$

SSD is collection of pairs (rs, n), where each rs is a role set and n is a natural number greater or equal than 2, with the property that no user is assigned to n or more roles from the set rs.

In order to deal with SSD restrictions in FRBAC, we define the following function:

- $role\_users(r : ROLES) \rightarrow 2^{USERS}$. That is, given a role r, the $role\_users(r)$ function returns the set of users assigned to the role r. The function is described as follows:

$$role\_users(r) = \{u_i|((u_i, r), \mu_{UA}(u_i, r)) \in UA\}$$

SSD restrictions are fulfilled if no user is assigned to mutually exclusive roles. That is:

$$\forall (rs, n) \in SSD, \forall s \in rs : |s| \geq n \Rightarrow \bigcap_{s \in rs} role\_users(s) = \emptyset$$

Where the intersection operand $\cap$ stands for:

$$(x, \mu_A(x)) \cap (y, \mu_A(y)) = \\ \begin{cases} \emptyset, & \text{if } x \neq y \\ \{(x, \min(\mu_A(x), \mu_A(y)))\}, & \text{if } x = y \end{cases}$$

Under the presence of roles hierarchies, SSD restrictions must take into account implicit user-role assignments coming through role inheritance. Junior roles inherit the member users from senior roles. We redefine the following function:

- $role\_users_{\downarrow}(r : ROLES) \rightarrow 2^{USERS}$:

$$role\_users_{\downarrow}(r) = \{role\_users(r) \cup role\_users(r'_i)| \\ (r'_i, r) \in RH\}$$

Then, SSD restrictions are fulfilled if:

$$\forall (rs, n) \in SSD, \forall s \in rs : |s| \geq n \Rightarrow \bigcap_{s \in rs} role\_users_{\downarrow}(s) = \emptyset$$

3. Discussion

FRBAC defines a fuzzy (or fractional) vision of the assignments. However, the concrete semantics of the assignments strongly depend on the application itself. We provide some examples of its applicability. The reader must note that the meaning of the user-role and role-permissions assignments must be coherent between them to make it possible their composition.

- Uncertainty: if the fact that a user or a privilege are assigned to a given role is based on uncertain information, the assignment degree can express such uncertainty. In this case a degree of 1 will express the complete lack of uncertainty that the user or the privilege belongs to the role, while a value of 0 will express the complete uncertainty about the user or the privilege belonging to such role.

- Risk: in some cases the assignment degree can be used to express the risk associated to the assignment of a given user or permission to a role. A degree of 1 means that there is no risk associated to the fact that a user or a privilege belong to a role, while a degree 0 means full risk.

- Similarity: in some applications, role engineering techniques are used to determine which roles should a user (or a privilege) be assigned to. Such methods sometimes rely in how similar are some user attributes compared to a fix set of attributes associated to each role. It is sometimes difficult to provide a crisp partition of the users into the roles. So the fuzzy nature of the user-role degree in FRBAC can more naturally accommodate all cases by expressing the similarity between the user attributes and the target attributes of the role.

- Role engineering results and policy exploration: in some applications, clustering methods can be applied over the permissions set of the users in order to find access profiles. In RBAC, access profiles are clearly represented by the roles. The distance between the kernel of the cluster and its members can be represented through the magnitude of the user-role assignment.

- QoS and user profiles: in some environments, restrictions in the resource allocation must be enforced for every user in the system. The magnitude of the user-privilege assignment can be used by the resource scheduler in order to prioritize the resource allocation. The user-attribute magnitude can be understood as a user profiling method.

- Exploration of hierarchies: inheritance relations can be found out comparing the user's membership and the permissions assigned between different roles. The RBAC standard defines an inheritance relation between two roles if all the members of the senior role are a subset of the members of the junior one and all the privileges of the junior role also belong to the privileges of the senior one. However, there are roles that do not completely meet these conditions but they do it in some extent. Analyzing the user's membership and the role-permission assignment, the inheritance degree of two roles can be computed determining how close are the roles to meet the two inheritance conditions.

4. Conclusions

In this paper we have described FRBAC, a generalization of the RBAC based on fuzzy sets. FRBAC defines the user-role, role-permission and thus the user-permission assignments as fuzzy relations. It allows to deal with imprecise authorization-related information and propagate it to the access decision. FRBAC allows to formulate fractional access decisions in order to deal with scenarios where actions have a fractional meaning such as QoS and data lying in databases. Moreover, in order to deal with operations that cannot be understood through a fractional view, FRBAC allows to defuzzificate the access decision making it binary. Although we present FRBAC to deal with these scenarios, others scenarios could also be accommodated due to the flexibility of the model.

Acknowledgements

Partial support by the Spanish MICINN (projects TSI2007-65406-C03-02, ARES- CONSOLIDER INGENIO 2010 CSD2007-00004) and Universitat Autònoma de Barcelona (PIF 472-01-1/07) is acknowledged.

References

[1] Biskup, J., Embley, D.W., Lochner, J.H.: Reducing inference control to access control for normalized database schemas. Information Processing Letters 106(1), 8 – 12 (2008)

[2] Blaze, M., Kannan, S., Lee, I., Sokolsky, O., Smith, J.M., Keromytis, A.D., Lee, W.: Dynamic trust management. Computer 42(2), 44–52 (2009)

[3] Covington, M.J., Moyer, M.J., Ahamad, M.: Generalized role-based access control for securing future applications. In: Proceedings of the 23rd National Information Systems Security Conference (NISSC). Baltimore, Maryland, USA (October 2000)

[4] Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed nist standard for role-based access control. ACM Transactions on Information and System Security. 4(3), 224–274 (2001)

[5] Ferraiolo, D., Kuhn, D., Chandramouli, R.: Role-Based Access Control. Artech House (2007)

[6] Ferraiolo, D., and Kuhn, D., Role-Based Access Control, In: Proceedings of the NIST-NSA National (USA) Computer Security Conference, pp. 554563 (1992)

[7] Foley, S.: Supporting imprecise delegation in keynote using similarity measures. In: Sixth Nordic Workshop on Secure IT Systems, pp. 101–119 (2001)

[8] Mendoza, F.A., López, A.M., Campo, C., García, R.C.: Trustac: Trust-based access control for pervasive devices. In: Hutter, D., Ullmann, M. (eds.) Lecture Notes in Computer Science, vol. 3450, pp. 225–238. Springer (2005)

[9] Navarro-Arribas, G., Foley, S.: Approximating SAML Using Similarity Based Imprecision. Intelligence in Communication Systems, vol. 190 of IFIP International Federation for Information Processing, pp. 191–200, Springer (2005)

[10] Nawarathna, U., K., S.R.: A fuzzy role based access control model for database security. Ceylon Journal of Science (Physical Sciences) (2007)

[11] Rajan, R., Verma, D., Kamat, S., Felstaine, E., Herzog, S.: A policy framework for integrated and differentiated services in the internet. IEEE Network, 13(5), pp. 36–41, (1999).

[12] Sandhu, R., Coynem E., Feinstein, H., Youman, C., Role-Based Access Control: A Multidimensional View, In: Proceedings of the 10th Annual Computer Security Applications Conference, pp. 5462 (Dec 1994)
[13] Sinclair, S., Smith, S.W., Trudeau, S., Johnson, M.E., Portera, A.: Information risk in financial institutions: Field study and research roadmap. In: FinanceCom. Lecture Notes in Business Information Processing, vol. 4, pp. 165–180. Springer (2007)
[14] Su, R., Zhang, Y., He, Z., Fan, S.: Trust-based fuzzy access control model research. In: WISM '09: Proceedings of the International Conference on Web Information Systems and Mining. pp. 393–399. Springer-Verlag, Berlin, Heidelberg (2009)
[15] Takabi, H., Amini, M., Jalili, R.: Enhancing role-based access control model through fuzzy relations. Information Assurance and Security, 2007. IAS 2007. Third International Symposium on pp. 131–136 (Aug 2007)
[16] Tran, H., Hitchens, M., Varadharajan, V., Watters, P.: A trust based access control framework for p2p file-sharing systems. In: HICSS '05: Proceedings of the 38th Annual Hawaii International Conference on System Sciences. vol. 9, pp. 302c. IEEE Computer Society, Washington, DC, USA (2005)
[17] Wang, C., Liu, S.: Study on fuzzy theory based web access control model. In: International Symposiums on Information Processing, pp. 178–182 (2008)
[18] Wiese, L.: Keeping Secrets in Possibilistic Knowledge Bases with Necessity-Valued Privacy Policies. In: International Conference on Information Processing and Management of Uncertainty, (2010)