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 Aut`onoma de Barcelona, 08193 Bellaterra, Spain*
[b] *IIIA, Institut d'Investigació en Intel·ligència Artificial - CSIC, Consejo Superior de Investigaciones Cient´ıficas, 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 through fuzzy relations that extends the applicability of RBAC to 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 risk-based access control.

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

1. Introduction

Role-Based Access Control (RBAC) [14, 4, 3] is widely used in corporate environments with many advantages, but it presents some problems and imposes some constraints in environments where the authorization-related information is imprecise. One example is the Aware Home project [2] where the level of accuracy of authenticating sensors is not perfect and thus this imprecision must be taken into account in the access decision.

Moreover, RBAC and in general traditional access control models, do no fit well in scenarios where actions have a fractional meaning and it makes no sense to permit or not their execution but to permit the execution of actions to a given degree. Data lying in databases is an example, where the responses to queries are modified in order to add a certain percentage of noise [19]. Another example is risk-based access control [13] where the system may allow the execution of an action with risk mitigation measures.

The aim of this paper is to introduce Fuzzy Role-Based Access Control (FRBAC) as a generalization of RBAC. It relies on 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. Furthermore, the FRBAC model contemplates role hierarchies and separation of duties.

Fuzzy relations, and fuzzy concepts in general, have been used to extend current access control models [17, 9, 15]. Focusing on RBAC, some proposals [18, 7, 16] aim to ease the user-role assignment through the notion of user trustworthiness and role required trustworthiness. A user is assigned to a role if the user trustworthiness is equal or greater than the role required trustworthiness. [9] makes an analogy between roles and trust degrees, so permissions are assigned to trust degrees rather than roles. The user trustworthiness determines the permissions that the user can activate. [11] provides additional security checks on top of RBAC when dealing with database security. The model determines the user's need to read and write records. A fuzzy multi-objective decision making process is used to determine whether a query can be executed or not.

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 flexibility to traditional access control models [6, 10, 1]. We are not aware of any proposal based on the ideas of RBAC which provides such flexibility in the user-role, role-permission assignment and the access decisions.

2. FRBAC

We introduce FRBAC departing from the RBAC definition of the standard in [5], which evolved to [4]. 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 use the following notation and definitions (as described in the RBAC definition from [4]).

- *US ERS* is a set of users.
- *ROLES* is a set of roles.

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

- *OBS* is a set of resources (objects).
- *OPS* is a set of operations.
- $PRMS = 2^{(OBS \times OPS)}$ is a set of permissions.
- $UA \subseteq USERS \times ROLES$ is a set of user-role assignments.
- $PA \subseteq PRMS \times ROLES$ is a set of role-permission assignments.

Although we follow the notation and definitions of RBAC, notice that we describe a different formulation in order to ease the understanding of the scheme. Regarding the use of sessions and subjects (questioned by some authors [12]), we do not follow the same notation and approach of the standard. Instead, we use a simpler approach, which helps to make our model more clear introducing a new relation (see Section 2.1) to determine the active roles of a given user.

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: $((r,p), \mu_{PA}(r,p))$ where $r \in ROLES$, $p \in PRMS$, and $\mu_{PA}(r,p) \rightarrow [0,1]$.

In FRBAC the notion of role activation has a different approach than RBAC and it is defined through the active assignments relation (AA) which is a subset of the user-role assignments containing the active roles for the users. The active assignments relation supports the principle of least privilege in that a user that is assigned to multiple roles may activate any subset of these roles to suit his or her tasks:

- $AA \subseteq UA : USERS \times ROLES \rightarrow [0,1]$

The user-permission relation (UP) is computed as the composition of the AA and PA relations and it is a collection of items of the form $((u,p), \mu_{UP}(u,p))$ where $u \in USERS$, $p \in PRMS$, and $\mu_{UP}(u,p)$ is a function that returns the user-role relation strength. The UP relation is defined as:

- $UP = AA \circ PA : USERS \times PRMS \rightarrow [0,1]$

Where the composing operand $\circ$ stands for the standard *max-min* composition of two fuzzy relations. Let $R_1 : X \times Y \rightarrow [0,1]$ and $R_2 : Y \times Z \rightarrow [0,1]$ be two fuzzy relations, the *max-min* composition $R_1 \circ R_2 : X \times Z \rightarrow [0,1]$ is defined as follows:

$$R_1 \circ R_2 = \{((x,z), \max_y(\min(\mu_{R_1}(x,y), \mu_{R_2}(y,z)))) \mid x \in X, y \in Y, z \in Z\}$$

Although we use the *maximum* operand as the union and the *minimum* as the intersection of fuzzy sets, note that other *t-conorm* and *t-norm* operands could be used respectively, giving up also to another relation composition operand [8].

Given the UP relation, we define the following function in order to compute the user-permission relation for a given user:

- $user\_permissions(u : USERS) \rightarrow \{(p, \mu_{UP}(u,p))\}$, where $p \in PRMS$ and $\mu_{UP}(u,p)) \rightarrow [0,1]$. 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:

$$user\_permissions(u) = \{(p, \mu_{UP}(u,p)) \mid ((u,p), \mu_{UP}(u,p)) \in UP\}$$

At access decision time, the response is subjected to find a privilege related to the user 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 o, the $access$ function returns the access degree that the user u has over the resource o through the operation op. The function is described as follows:

$$access(u, op, o) = \{\mu_P(u,p) \mid (p, \mu_{UP}(u,p)) \in user\_permissions(u) \wedge (op,o) \in p\}$$

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 executed at all. 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 o, the $access$

function returns a boolean value which dictates whether the user u is allowed to execute the action op over the resource o. The function is described as follows:

$$access_{\delta}(u, op, o) = access(u, op, o) \geq \delta$$

2.2. Hierarchical FRBAC

The RBAC standard defines a hierarchical relation between roles. Given a role r, the set of users belonging to the role (U_r) and the set of permissions assigned to the role (P_r), a role r' is a junior role of r if $U_r \subseteq U_{r'}$ and $P_{r'} \subseteq P_r$. 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.

Inheritance is described in the RBAC standard as a partial order (reflexive, antisymmetric and transitive) RH on ROLES:

- $RH \subseteq ROLES \times ROLES$

In FRBAC, the inheritance relation RH is described as a reflexive, antisymmetric and transitive fuzzy order relation:

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

The fuzzy inheritance mapping is a set of items of the form $((r, r'), \mu_{RH}(r, r'))$ where r and $r' \in ROLES$, and $\mu_{RH}(r, r')$ is a function that returns the inheritance strength. The strength is valued in the real unit interval $[0, 1]$. The senior role is represented by r, and r' represents the junior one. Note that the reflexive property guarantees that $\forall r \in ROLES \Rightarrow \mu_{RH}(r, r) = 1$.

It is noteworthy to mention that a crisp inheritance model, where the inheritance relation is binary, can be easily accommodated in FRBAC setting $\mu_{RH}(r, r')$ as 1 if r is a senior role of r' and 0 otherwise.

The user-role assignment, the active assignments and user-permission assignment under the presence of role hierarchies is determined by the following relations:

- $UA_{\downarrow} = UA \circ RH : USERS \times ROLES \rightarrow [0, 1]$
- $AA_{\downarrow} \subseteq UA_{\downarrow} : USERS \times ROLES \rightarrow [0, 1]$
- $UP_{\downarrow} = AA_{\downarrow} \circ PA : USERS \times PRMS \rightarrow [0, 1]$

The $UP_{\downarrow}$ relation replaces the UP relation described in the Core FRBAC model when computing the user-permission assignment ($user\_permissions(u)$) in order to make the access decision ($access(u, op, o)$ and $access_{\delta}(u, op, o)$).

2.3. Constrained FRBAC

The RBAC standard defines two types of separation of duties: Static Separation of Duties (SSD) and Dynamic Separation of Duties (DSD). SSD apply in RBAC in order to prevent users to be assigned to a role set which allows by itself to misuse the system. DSD restricts the roles that a user activates in a session.

The RBAC standard defines both the collection SSD and DSD as:

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

SSD is a 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. DSD has a similar form with the property that no user is active in n or more roles of the set rs.

In order to deal with separation of duties restrictions in FRBAC, we define the following functions:

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

$$role\_users(r) = \{u \mid ((u, r), \mu_{UA}(u, r)) \in UA \wedge \mu_{UA}(u, r) > 0\}$$

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

$$active\_role\_users(r) = \{u \mid ((u, r), \mu_{AA}(u, r)) \in AA \wedge \mu_{AA}(u, r) > 0\}$$

SSD restrictions are fulfilled if, for all $(rs, n) \in SSD$, no user is assigned to n or more mutually exclusive roles from the set rs. That is:

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

DSD restrictions are fulfilled if, for all $(rs, n) \in DSD$, no user has active n or more mutually exclusive roles from the set rs. The function $active\_role\_users(r)$ and the relation DSD replace the function $role\_users(r)$ and the relation SSD in the above sentence.

Under the presence of role hierarchies, separation of duties restrictions must take into account implicit user-role assignments by virtue of the role inheritance. The $UA_{\downarrow}$ relation replaces UA in the $role\_users(r)$ function and the $AA_{\downarrow}$ relation replaces AA in the $active\_role\_users(r)$ function.

3. Applicability of the approach

In this section we show some application scenarios and examples of FRBAC.

3.1. Data lying in databases

One example of fractional actions can be found in data lying in databases [19]. Data lying is, in fact, an access control mechanism that allows to add a degree of uncertainty in the query responses. A censor module is in charge to properly distort the query responses in order to add noise. This adjustable degree of uncertainty fits well with the notion of fractional actions since

the more privileges the user has, the more truth she obtains in the responses.

In terms of access control, the censor module can be seen as the *policy enforcement point* of the application. That is, the censor receives a query submitted by a user and obtains the submitter's privileges. Taking into account the user's privileges, the censor module determines whether the user can execute the action and, if permitted, in what degree.

FRBAC can easily accommodate an RBAC-based data lying scheme. The user-role relation strength can represent useful authorization-related information such as user's trustworthiness, user's seniority, user's need-to-know, or any other application-dependent information, which can be considered useful at authorization time. The role-privilege relation strength must be coherent with the user-role relation in order to allow their composition. Finally, the user-privilege strength is used in the censor module to enforce the execution of the query.

3.1.1. Example

Consider a database of a hospital which can be accessed by an external party in order to carry out epidemical research. The user-role assignment strength represents the user's trustworthiness. The role-privilege assignment is computed as the average of user's trustworthiness of all the members of the role. The role-privileges and the active user-role assignments are shown in Table 1.

	Cardio.	Radio.
$user_1$	0.8	0
$user_2$	0.9	0
$user_3$	0	0.5

	QueryDB
Cardio.	0.85
Radio.	0.5

Table 1: Active user-role and role-permission assignments.

Composing the relations, it can be derived that $user_1$ is assigned to the permission *QueryDB* through a magnitude of 0.8. Thus, the $user_1$ is allowed to query the database and receive answers with a certainty degree of 0.8 [19].

3.2. Uncertain user authentication

In some environments, the fact that a user is assigned to a role is based on uncertain information. This phenomenon can be observed in the Aware Home Project [2], where information available from sensors in the home should be used to automatically infer the user's security-relevant attributes (e.g., identity, role or location.). Many such sensors can establish the security-relevant attributes of a subject with only a partial level of certainty, or confidence level.

The FRBAC model can naturally accommodate the accuracy degree of a user belonging to a role through the user-role assignment strength. This imprecision degree can be propagated through the user-permission relation to the access decision. The access decision can be formulated in a fuzzy manner if the action being requested has a fractional meaning. Otherwise, the access decision can be defuzzified, setting δ as the maximum imprecision degree that the system is willing to tolerate.

3.2.1. Example

Covington et al. [2] describe the following scenario: *"[...] consider an adult who wants to view the output of a video camera in a childs bedroom, for the purpose of checking on the child. The security policy may state that only the childs parents or babysitter can view the video. Perhaps a strong identification mechanism may provide enough authentication evidence to allow the user to see a streaming video, while a weak identification mechanism may provide only enough authentication evidence to permit the user to view a recent still image of reduced quality and definition. [...]"*.

Imagine that the user *Alice* plays the role *babysitter*. *Alice* has been identified through a voice recognition sensor which provides 70% of accuracy. The system relates *Alice* with the role *babysitter* through a magnitude of 0 .7. The users of the role *babysitter* are allowed to view the output of the video camera (that is a role-privilege strength of 1). Composing both, the user-role and role privilege assignments leads the user-permission strength of the relation *Alice-AccessCam* as 0.7. Finally, *Alice* will be allowed to view the output of the video camera with a reduction of quality and definition according to the magnitude of the privilege assignment.

3.3. Other applications

We briefly describe some more examples showing the applicability of FRBAC.

3.3.1. Risk-based access control

Access control can be understood as a mechanisms used to manage risk, i.e., to balance the information needs of the users with the need of the organization to protect its sensitive information [13]. FRBAC can be used as the basis of a risk-based access control. The user-role relation strength can represent the risk associated to the fact that a user belongs to a role. The magnitude of strength is application-dependent and can be derived from the user trustworthiness, for example. In the same way, the role-permission relation strength can represent the risk involving the assignment of a given permission to a role. The user-privilege relation would reflect the risk involving the assignment of a given privilege to a given user. The *enforcement point* of the application must evaluate the risk in order to determine possible risk mitigation measures conditioning the execution of the action, if permitted.

3.3.2. Exploration of role 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. It can be used just with an analytic propose or to enable fuzzy inheritance.

4. Conclusions

In this paper we have described FRBAC, a generalization of RBAC built 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 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. Hierarchical and constrained versions of FRBAC has been also described in the paper. 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 TIN2010-15764, TSI2007-65406-C03-02, ARES- CONSOLIDER INGENIO 2010 CSD2007-00004) and Universitat Aut ònoma 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 vol. 106, no. 1, pp. 8–12 (2008)

[2] 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)

[3] Ferraiolo, D., Kuhn, D.: Role-Based Access Control, In: Proceedings of the NIST-NSA National (USA) Computer Security Conference, pp. 554–563 (1992)

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

[5] Ferraiolo, D., Sandhu, R., Gavrila, S., Kuhn, D., Chandramouli, R.: Proposed nist standard for role-based access control. ACM Transactions on Information and System Security. vol. 4, no. 3, pp. 224–274 (2001)

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

[7] H. Takabi, M. Amini, R. Jalili: Trust-Based User-Role Assignment in Role-Based Access Control, In Proceedings of the ACS /IEEE International Conference on Computer Systems and Applications AICCSA, (2007).

[8] Klir, G., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall (1995).

[9] 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)

[10] 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)

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

[12] Ninghui Li, JiWon Byun, Bertino, E.: A Critique of the ANSI Standard on Role-Based Access Control, Security & Privacy, IEEE , vol.5, no.6, pp.41-49, Nov.-Dec. (2007)

[13] Pau-Chen Cheng, Rohatgi, P., Keser, C., Karger, P.A., Wagner, G.M., Reninger, A.S.:Fuzzy Multi-Level Security: An Experiment on Quantified Risk-Adaptive Access Control. Security and Privacy, 2007. SP '07. IEEE Symposium on, pp.222–230 (May 2007)

[14] 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. 54–62 (Dec 1994)

[15] 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)

[16] 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)

[17] 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)

[18] Wang, C., Liu, S.: Study on fuzzy theory based web access control model. In: International Symposiums on Information Processing, pp. 178–182 (2008)

[19] Wiese, L.: Keeping Secrets in Possibilistic Knowledge Bases with Necessity-Valued Privacy Policies. In: International Conference on Information Processing and Management of Uncertainty, (2010)