Chapter 3

INTEGRITY-ORGANIZATION BASED ACCESS CONTROL FOR CRITICAL INFRASTRUCTURE SYSTEMS

Abdeljebar Ameziane El Hassani, Anas Abou El Kalam, and Abdellah Ait Ouahman

Abstract The organization-based access control (OrBAC) model is an access control model that helps evaluate the security policies of organizations. OrBAC affords a high degree of expressiveness and scalability. The model, however, does not readily express integrity constraints. Integrity is one of the most important properties for critical infrastructure systems, mainly due to their criticality and low tolerance of corruption and alterations. This paper describes an extension of OrBAC, called Integrity-OrBAC (I-OrBAC), which models integrity attributes associated with critical infrastructure systems. I-OrBAC facilitates the modeling of multiple integrity levels to express the requirements of different critical infrastructure organizations. An example security policy is presented to demonstrate the expressiveness of the model.

Keywords: Access control, organization-based control, security models, integrity

1. Introduction

The growing sophistication and interconnection of information systems have increased their vulnerability to attacks. This applies especially to critical infrastructures, which are increasingly dependent on information systems but tend not to tolerate disturbances.

Critical infrastructures are assets whose proper functioning is essential to a societal welfare (e.g., energy distribution and transmission, telecommunications and railway infrastructures). These assets often require the collaboration of multiple organizations to receive and/or provide services. In order to protect these assets throughout the various collaborative activities, security policies and enforcement mechanisms are required that clearly identify the needs, vulnerabilities and threats.

J. Butts and S. Shenoi (Eds.): Critical Infrastructure Protection VI, IFIP AICT 390, pp. 31–42, 2012.

The security policy of an organization defines guidelines that specify authorized and unauthorized activities. Security models provide mechanisms to evaluate security policies for completeness and adequacy with regard to security properties. Various security models exist for evaluating the confidentiality, integrity and availability of systems. Critical infrastructures, however, have unique characteristics that are not considered in the development of traditional security models. Of special interest are the extensive integrity requirements associated with critical infrastructures.

The organization-based access control (OrBAC) model has been demonstrated to be very effective for specifying security policies of organizations [1]. However, the OrBAC model has certain deficiencies with regard to ensuring integrity. This paper describes an extension of OrBAC, called Integrity-OrBAC (I-OrBAC), which is specifically designed to express integrity requirements in critical infrastructure environments.

2. Background

Security models facilitate the expression and evaluation of security policies. The first security models such as discretionary access control (DAC) [15] and mandatory access control (MAC) [4, 6] enforced a single level of abstraction for representing user permissions. Although they enabled the formal specification of security policies, expressibility was limited and the update functions were complicated and time consuming. Subsequent security models such as role-based access control (RBAC) [11, 12] introduced a second level of abstraction to facilitate manageable update functions and to include dynamic access control rules. Other models support policy specification by integrating notions of obligations [16] and prohibitions [5] to express exceptions.

2.1 OrBAC Security Model

The OrBAC model [1], designed as an extension to RBAC, uses two levels of abstraction to express a security policy: (i) a concrete level; and (ii) an abstract level. The concrete level includes subjects, actions and objects. The abstract level specifies security policies using roles, activities and views. Subjects are abstracted into roles that can perform the same activities (i.e., actions defined by security rules). Objects are similarly abstracted into groups, called views and activities, according to the applicable security rules. Abstract entities enable the expression of organization-specific policies via abstract privileges. Concrete privileges can then be derived to help evaluate the validity of system requests based on situations and conditions. Figure 1 summarizes the various relations and entities in the OrBAC model.

OrBAC adopts a centralized approach; it does not express access control in distributed and collaborative environments. Security models such as PolyOrBAC [2, 3], however, extend OrBAC for access control in collaborative environments. Nevertheless, it is important to further develop OrBAC because it

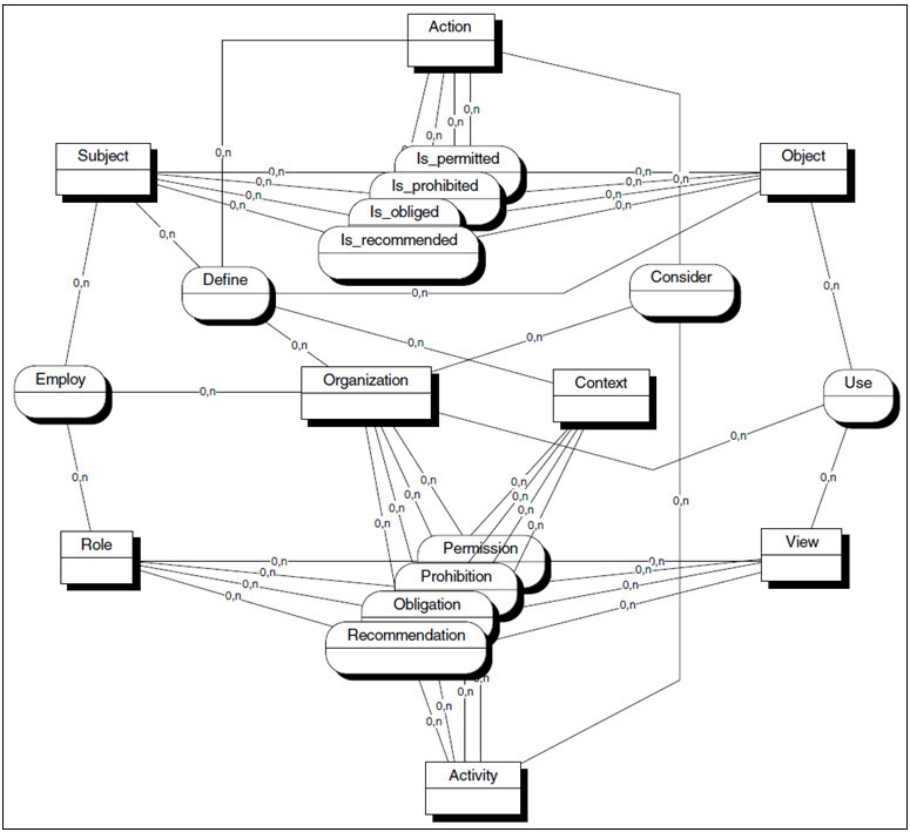

Figure 1. Representation of OrBAC model relations.

expresses organization security policies consistent with the requirements associated with critical infrastructure systems.

Unfortunately, OrBAC does not express policies that enforce data and system integrity. Indeed, subjects inherit privileges granted to roles without *a priori* verification of their empowerment or consideration of their credibility. In addition, all views (i.e., objects) and activities (i.e., actions) are considered to be equally important; this is not representative of operating parameters for critical infrastructure systems. For this purpose, we enrich OrBAC with concepts and mechanisms to help address integrity requirements.

2.2 Integrity Requirements

Critical infrastructures rely extensively on the proper operation and availability of system services. Due to society's reliance on the associated resources, service disruptions can lead to cascading and escalating phenomena with serious financial losses and possibly catastrophic consequences. Although many of

the operating parameters are within the control of asset owners, partnerships, interconnectivity and reliance on assets belonging to other organizations are often necessary. Indeed, the ability to trust data received from other entities is essential to operations. Key aspects associated with data management and trust include secure cooperation, audit and assessment, autonomy and loose coupling, enforcement of permissions, explicit prohibition and obligation rules [2].

This paper focuses on integrity requirements in critical infrastructure environments. Integrity is the property of information to be correct [7]. In this sense, a system must: (i) prevent the unauthorized modification of information (e.g., creation, update and destruction); and (ii) allow the legitimate modification of information. The next section extends the OrBAC model to facilitate the expression of integrity properties for critical infrastructure assets.

3. Integrity-OrBAC Model

According to Krause and Tipton [14], the Biba model [6] was the first security model designed to ensure integrity. Subsequent models (e.g., Goguen-Meseguer [13], Sutherland [14], Clark-Wilson [9], Brewer-Nash (Chinese Wall) [8] and Totel [17]) also provide a means for specifying integrity in security policies. Integrity-specific models, however, are not expressive enough to model the operating parameters, requirements and interactions associated with critical infrastructure assets. OrBAC can model critical infrastructure characteristics, but it does not have the requisite properties for specifying integrity.

Critical infrastructure systems incorporate a wide range of data types that require different integrity requirements depending on the functionality with which they are associated. Additionally, actions within an organization do not all carry the same risks; for example, actions that may involve serious consequences receive higher scrutiny. Moreover, subjects have different expertise and skill levels for performing different tasks. Finally, in addition to technical criteria, subjects should be categorized according to their trustworthiness.

3.1 Assigning Integrity Levels

When developing a critical infrastructure security policy, it is important to properly distinguish several components: (i) information type for each object; (ii) difference between highly sensitive and routine actions; (iii) expertise and skill levels for performing actions; and (iv) degree of trustworthiness associated with each subject. In this sense, the assignment of multilevel integrity values for concrete OrBAC entities must adequately reflect critical infrastructure requirements.

Subject Integrity Levels. The integrity level of each subject is determined on the basis of defined criteria as it relates to the organization. Integrity levels are assigned to the concrete abstraction for each subject. Consider, for example, the role of a "Pilot" in an aviation environment. Not all pilots have

Table 1. Vector representation of subject integrity levels.

	Generalist	Surgeon	Resuscitator	Anesthetist	Therapist	Cardiologist	Neurologist	Trauma	Dentist
Bob	3	3	2	2	–	3	–	–	–
Alice	2	–	–	–	3	–	–	3	–
Eve	3	3	2	2	–	3	–	–	–

the same expertise level; a reputation is earned based on hours of flight time and types of airframe flown. Each pilot is subsequently assigned an integrity level based on the defined parameters. Note that a subject receives a unique integrity level associated with each role performed within the organization. Table 1 presents integrity levels for medical professionals (subjects) using a vector representation.

View and Object Integrity Levels. The integrity level of each object is determined based on the degree of trust inherited from the respective view. The inheritance process affords high expressiveness and also reduces administrative costs. To illustrate this, we revisit the aviation domain. Consider the view *flight_parameters* containing the objects $flight\_plan_x$, $takeoff\_speed_y$ and $altitude_z$; and the view *passengers_data* containing the objects $travel\_class_x$ and $customized\_service_y$. Clearly, the objects contained in the first view are more critical than those in the second view. Thus, the *flight_parameters* view is assigned a higher integrity level than the *passengers_data* view. Note that all the objects in a view inherit the associated integrity level.

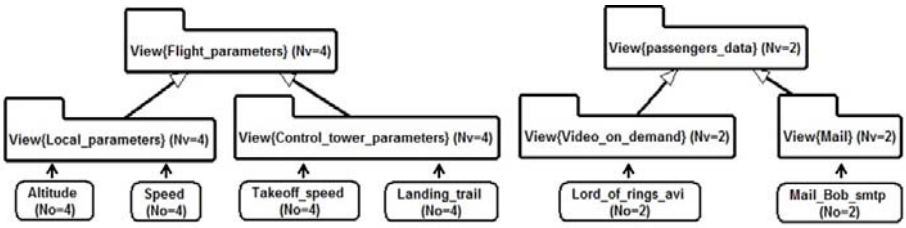

Figure 2. Representation of a view integrity structure.

As illustrated in Figure 2, the views are structured by organization. Consider the object *takeoff_speed*, which is data communicated by airport authorities. The pilot must rely on this information and it is placed in a high integrity level view. Similarly, the *speed* object, as observed from the local parameters of the aircraft, has the same integrity level inherited from *flight_parameters*.

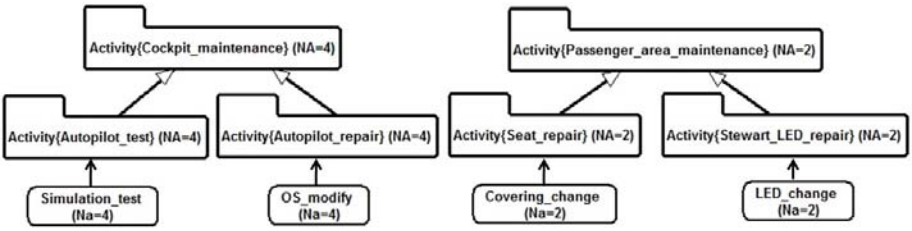

Figure 3. Representation of an activity integrity structure.

Activity and Action Integrity Levels. Actions with the same sensitivity and presenting the same risks to an organization are grouped together in a common activity. As with objects and views, integrity levels are extrapolated to the abstract activity and not to concrete actions. Integrity levels are assigned to activities based on the criticality of the actions they contain and the severity of the impact to the organization. In the aviation example, consider two activities, *cockpit_maintenance* and *passenger_area_maintenance*. Cockpit maintenance requires priority actions to enable safe flight, whereas actions related to the maintenance of passenger area equipment do not have the same criticality. Figure 3 presents an organized representation of activities.

Context Integrity Levels. The ability to understand the context of a system request depends on temporal and spatial characteristics, the purpose and previous actions. Considerations associated with the assignment of context integrity levels include the time the subject requests access, location of the subject, purpose of the access and previous access to the requested object.

Assigning Integrity Levels. In order to quantify integrity levels, meaningful scales must be established that adequately express the associated risks. Figure 4 provides example scales established for roles, views, activities and contexts. Note that other scales can be defined according to the requirements of each critical infrastructure organization.

To ensure overall integrity, privileges are granted by evaluating three parameters: (i) integrity level of the view; (ii) integrity level of the activity; and (iii) integrity level of the context. These three parameters impose constraints on the required integrity level of the subject. An operation is authorized if the subject has the appropriate integrity level.

3.2 I-OrBAC Model Components

I-OrBAC extends OrBAC to incorporate integrity. I-OrBAC is, therefore, based on OrBAC entities, relations, language and axioms. The following expressions summarize the primary OrBAC components used in I-OrBAC. The notation *Org* denotes the organization defined by the security policy. Note that $S \cap O = \emptyset$ in the definitions below.

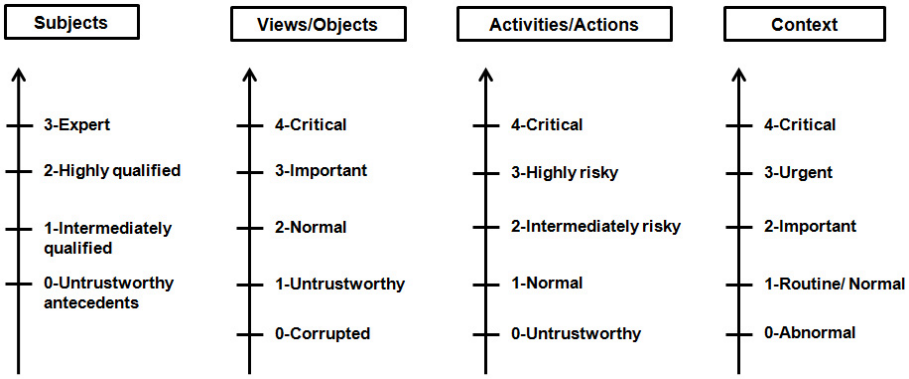

Figure 4. Example integrity level scales.

Specification of entities:

- S: Set of Org subjects
- AC: Set of Org actions
- O: Set of Org objects
- R: Set of Org roles
- AY: Set of Org activities
- V: Set of Org views
- C: Set of contexts related to Org

Relations defining access modes in the abstract level:

- Permission(Org, R, V, AY, C)
- Prohibition(Org, R, V, AY, C)
- Recommendation(Org, R, V, AY, C)
- Obligation(Org, R, V, AY, C)

Relations defining access modes in the concrete level:

- Is_Permitted(S, O, AC)
- Is_Prohibited(S, O, AC)
- Is_Recommended(S, O, AC)
- Is_Obliged(S, O, AC)

Relation affecting a role R to a subject S in Org:

- Empower(Org, S, R)

Relation affecting an object O to a view V in Org:

- Use(Org, O, V)

Relation affecting an action AC to an activity AY in Org:

- Consider(Org, AC, AY)

Relation including the concrete elements of a task – subject S, object O, action AC and context C:

- Define(Org, S, O, AC, C)

Several expressions are introduced to extend OrBAC. These include:

Specification of integrity levels for entities:

- N_S: Set of integrity levels for subjects
- N_V: Set of integrity levels for views
- N_{AY}: Set of integrity levels for activities
- N_C: Set of integrity levels for contexts

Full ordering relation defining "greater than or equal to" for determining the required integrity access level:

- $N_S \geqslant (N_V \times N_V, N_{AY} \times N_{AY}, N_C \times N_C)$

In addition, the relations Empower(), Use(), Consider() and Define() are modified to account for integrity levels.

Value of subject S integrity level in role R:

- Empower(Org, S, R, N_S)

Value of view V integrity level and, by inheritance, object O integrity level:

- Use(Org, O, V, N_V)

Value of activity AY integrity level and, by inheritance, action AC integrity level:

- Consider(Org, AC, AY, N_{AY})

Value of context C integrity level:

- Define(Org, S, O, AC, C, N_C)

4. Access Control Policy Example

This section describes an example from the medical domain. Note, however, that the expressiveness of I-OrBAC enables it to be applied to myriad critical infrastructure assets.

The organization in this example is a hospital *Purpan* that is assessing the treatment of a cancer patient $S = y$ such that y is in the view $V = pat\_surg$, implying the requirement of a surgical intervention. The activity involves critical surgeries $AY = cr\_surg$ and, more specifically, an ablation procedure action $AC = ab\_proc$. The context is considered to be high risk $C = h\_r$ because the patient's life depends on the surgery. Therefore, the constraint that must be evaluated is the integrity level to impose on the doctor $S = x$ who performs the ablation action.

First, the doctor must be a surgeon with the role $R = surg$. It is then necessary to determine the minimum integrity level of the role *surg* that would allow a doctor to perform an ablation. This is accomplished on the basis of integrity levels for *pat_surg*, *cr_surg* and *h_r*.

Consider the following expressions of OrBAC rules corresponding to the *Purpan* security policy:

- Permission(*Purpan*, *surg*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *x*, *surg*)
- Use(*Purpan*, *y*, *pat_surg*)
- Consider(*Purpan*, *ab_proc*, *cr_surg*)
- Define(*Purpan*, *x*, *y*, *ab_proc*, *h_r*)

Permission() enables a surgeon to perform surgery on a patient in the view *pat_surg* associated with context *h_r*. Empower() enables a doctor x to perform in the role *surg*. Use() identifies the patient y in the view *pat_surg*. Consider() includes *ab_proc* as a part of the activity *cr_surg*. Define() provides the context *h_r* of the action *ab_proc*.

The previously established integrity level scales are used to express the rules of the *Purpan* security policy. We consider the context *h_r* as critical with an integrity level N_C=4. The activity *cr_surg* includes the set of critical surgeries and, as a member, the ablation surgery inherits the associated integrity level assigned N_{AY}=4. The view *pat_surg* groups patients who require ablation due to cancer; these patients are at risk of death and require difficult interventions. As such, the patient y is assigned a high integrity level N_V=4.

Given the integrity levels, N_C=4, N_V=4 and N_{AY}=4, we consider that the security policy of *Purpan* only allows surgeons whose integrity level $N_S \geqslant 3$ (i.e., expert subjects in their role) to perform this surgery. The following expressions of I-OrBAC rules articulate these constraints:

- Permission (*Purpan*, *surg*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *x*, *surg*, N_S=3)
- Use(*Purpan*, *y*, *pat_surg*, N_V=4)
- Consider(*Purpan*, *ab_proc*, *cr_surg*, N_{AY}=4)
- Define(*Purpan*, *x*, *y*, *ab_proc*, *h_r*, N_C=4)

A verification of the subject's integrity level N_S is performed to ensure that for x, $N_S \geqslant 3$. Once this is verified, a rule is generated to authorize the action Is_permitted(x, *ab_proc*, y).

4.1 Flexible Integrity Levels

The ability to assign varying integrity levels affords flexibility while protecting objects from unauthorized modification. In the event of an emergency, intervention may be required because of time constraints or the absence of the primary authorized subject. The ability to assign subject integrity levels enables alternative authorizations. For example, assume that the ablation surgery can be performed by a surgeon who specializes in ablation (*s_ab*), an aesthetic surgeon(*s_aesth*) and a general surgeon (*surg*). Individuals in these roles are neither equally skilled nor do they have the same expertise in ablation surgery. It is clear that the most appropriate person to perform the surgery is a surgeon who specializes in ablation, followed by an aesthetic surgeon and, finally, a general surgeon. For this scenario, the security policy imposes different integrity level thresholds for each role in order to perform ablation. The rules are expressed as follows using a six integrity level scale for subjects:

- Permission(*Purpan*, *s_ab*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *bob*, *s_ab*, $N_S = 3$)

or

- Permission(*Purpan*, *s_aesth*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *alice*, *s_aesth*, $n_S = 4$)

or

- Permission(*Purpan*, *surg*, *pat_surg*, *cr_surg*, *h_r*)
- Empower(*Purpan*, *eve*, *surg*, $N_S = 5$)

The different integrity level thresholds imposed on each role provide a means for enforcing organization guidelines. The security policy strongly recommends that an ablation be performed by a surgeon specialized in ablation. If one is not available, then it is recommended that the ablation be performed by an aesthetic surgeon, followed by a highly skilled surgeon. This flexibility is a variation of the notion of recommendation [10] introduced by OrBAC.

4.2 Integrity Principle Expressed via I-OrBAC

Separation of privilege [7] is a primary security principle that is associated with safeguarding systems and enforcing integrity standards. Separation of privilege states that privileges should be distributed among multiple, independent components such that multiple agreement is necessary to perform an action

(i.e., permission should not be granted based on a single condition). The principle, which is sometimes termed the "two-person rule," ensures high integrity, most notably in highly critical tasks (e.g., launching a nuclear weapon).

Consider a critical context activity that cannot be accomplished without the collaboration of two subjects. Initial constraints require prohibiting all subjects with role R from performing an action AC on their own:

- Prohibition(Org, R, V, AY, C)
- Obligation(Org, $R\hat{}R$, V, AY, C)
- Empower(Org, s_1, R, N_{s1})
- Empower(Org, s_2, R, N_{s2})
- Use(Org, O, V, N_V)
- Consider(Org, AC, AY, N_{AY})
- Define(Org, $s_1\hat{}s_2$, O, AC, C, N_C)
- Is_Obliged($s_1\hat{}s_2$, AC, O)

In the example, two subjects are needed to authorize the action, each of them with the necessary integrity threshold level. As demonstrated, the expressiveness of I-OrBAC enables the articulation of realistic constraints in order to preserve the integrity of objects, actions and contexts.

5. Conclusions

The I-OrBAC extension of the OrBAC model considers integrity aspects and expresses requirements associated with critical infrastructure assets. In particular, I-OrBAC incorporates concepts and components of the OrBAC model while addressing integrity concerns. I-OrBAC quantifies the credibility of subjects along with the criticality of views, activities and contexts in order to preserve the integrity of critical infrastructure assets. The approach supports the modeling of multiple integrity levels to effectively express the requirements of different organizations.

Our future research will focus on secure collaboration in critical infrastructure environments. It will also attempt to develop and evaluate common security policies to determine the most effective implementations for organizations.

References

[1] A. Abou El Kalam, S. Benferhat, A. Miege, R. El Baida, F. Cuppens, C. Saurel, P. Balbiani, Y. Deswarte and G. Trouessin, Organization based access control, *Proceedings of the Fourth International Workshop on Policies for Distributed Systems and Networks*, pp. 120–131, 2003.

[2] A. Abou El Kalam, Y. Deswarte, A. Baina and M. Kaaniche, PolyOrBAC: A security framework for critical infrastructures, *International Journal of Critical Infrastructure Protection*, vol. 2(4), pp. 154–169, 2009.

[3] A. Baina, A. Abou El Kalam, Y. Deswarte and M. Kaaniche, Collaborative access control framework for critical infrastructures, in *Critical Infrastructure Protection II*, M. Papa and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 189–201, 2008.

[4] D. Bell and L. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation, Technical Report ESD-TR-75-306, MITRE Corporation, Bedford, Massachusetts, 1975.

[5] S. Benferhat, R. El Baida and F. Cuppens, A stratification-based approach for handling conflicts in access control, *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies*, pp. 189–195, 2003.

[6] K. Biba, Integrity Considerations for Secure Computer Systems, Technical Report ESD-TR-76-372, MITRE Corporation, Bedford, Massachusetts, 1977.

[7] M. Bishop, *Computer Security: Art and Science*, Addison-Wesley, Boston, Massachusetts, 2003.

[8] D. Brewer and M. Nash, The Chinese Wall security policy, *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 206–214, 1988.

[9] D. Clark and D. Wilson, A comparison of commercial and military computer security policies, *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 184–195, 1987.

[10] N. Essaouini, A. Abou El Kalam and A. Ait Ouahman, Access control policy: A framework to enforce recommendations, *International Journal of Computer Science and Information Technologies*, vol. 2(5), pp. 2452–2463, 2011.

[11] D. Ferraiolo and D. Kuhn, Role based access control, *Proceedings of the Fifteenth National Computer Security Conference*, pp. 554–563, 1992.

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

[13] J. Goguen and J. Meseguer, Security policies and security models, *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 11–20, 1982.

[14] M. Krause and H. Tipton, *Handbook of Information Security Management*, Auerbach Publications, Boca Raton, Florida, 1998.

[15] B. Lampson, Protection, *Proceedings of the Fifth Princeton Symposium on Information Sciences and Systems*, pp. 437–443, 1971.

[16] R. Sandhu and J. Park, Usage control: A vision for next generation access control, *Proceedings of the Second International Workshop on Mathematical Methods, Models and Architectures for Computer Network Security*, pp. 17–31, 2003.

[17] E. Totel, J. Blanquart, Y. Deswarte and D. Powell, Supporting multiple levels of criticality, *Proceedings of the Twenty-Eighth IEEE Fault Tolerant Computing Symposium*, pp. 70–79, 1998.