A Trust-Based Access Control Model for Pervasive Computing Applications

Manachai Toahchoodee, Ramadan Abdunabi, Indrakshi Ray, and Indrajit Ray*

Department of Computer Science
Colorado State University
Fort Collins CO 80523-1873

Abstract. With the rapid growth in wireless networks and sensor and mobile devices, we are moving towards an era of pervasive computing. Access control is challenging in these environments. In this work, we propose a trust based approach for access control for pervasive computing systems. Our previously proposed belief based trust model is used to evaluate the trustworthiness of users. Fine-grained access control is achieved depending on the trust levels of users. We develop a class of trust-based access control models having very formal semantics, expressed in graph theory. The models differ with respect to the features they provide, and the types of the trust constraints that they can support.

1 Introduction

Traditional access control models like Mandatory Access Control (MAC), Discretionary Access Control (DAC), and Role-Based Access Control (RBAC) do not work well in pervasive computing systems. Pervasive computing systems are complex, involving rich interactions among various entities. The entities that a system will interact with or the resources that will be accessed are not always known in advance. Thus, it is almost impossible to build a well-defined security perimeter within which the system will operate. Since almost all traditional access control models rely on successful authentication of predefined users, they become unsuitable for pervasive computing systems. Moreover, these systems use the knowledge of surrounding physical spaces to provide services. This requires security policies to use contextual information. For instance, access to a resource may be contingent upon environmental contexts, such as the location of the user and time of day and can be exploited to breach security. Contextual information must, therefore, be protected by appropriate policies.

Researchers have recently started extending the RBAC model to accomodate contextual information such as time and location [1, 2, 4–7, 10, 9, 11, 12]. However, none of these models address the problem of unknown user in access control. Other researchers have proposed ways to incorporate the concept of trust to RBAC to address this particular problem [13, 3]. The general idea in these works is that the access privileges of a user depends on his trust level. However, the applicability of these models to pervasive computing environments remains to be investigated.

* This work was supported in part by the U.S. AFOSR under contract FA9550-07-1-0042

In this paper, we propose a trust-based RBAC model for pervasive computing systems. We adapt the context-sensitive trust model proposed by us earlier [8] in this work. We develop three versions of the model that cater to different circumstances within a system. Users (humans or their representatives and devices) are evaluated for their trustworthiness before they are assigned to different roles. Roles are associated with a trust range indicating the minimum trust level that a user needs to attain before it can be assigned to that role. A permission is also associated with a trust range indicating the minium trust level a user in a specific role needs to attain to activate the permission. The semantics of our model is expressed in graph-theoretic notations that allows us to formulate precise semantics for the model.

The rest of the paper is organized as follows. Section 2 describes how we can evaluate the trust value of the user entity in our model. Section 3 specifies our model using a graphical representation and also presents the different types separation of duty constraints that we can have in our model. Section 4 concludes the paper.

2 Trust Computation

We adapt the trust model proposed by Ray et al. [8]. Initially, an entity A does not trust a new entity B completely. Entity A needs to evaluate a trust relationship with entity B in some context. The context in our model is the role to which a user will be assigned to. We will refer to the context as a role context rc. Users can be associated with multiple roles. To determine the authorization between a user and a role, a user's trust value is evaluated based on each role context separately. The trust relationship between human user or device user, and the system in the role context rc depends on three factors: *properties*, *experience*, and *recommendations*. The semantics of these three factors are different for the human and the device user. We formally represent a trust relationship between truster, A, and trustee, B, on some role context rc, as a triple $({}_Ab_B^{rc}, {}_Ad_B^{rc}, {}_Au_B^{rc})$, where ${}_Ab_B^{rc}$ is A's belief on B about the latter's trustworthiness, ${}_Ad_B^{rc}$ is A's disbelief on B, and ${}_Au_B^{rc}$ is A's uncertainty on B. Each of these components has a value between [0,1] and sum of these components is 1.

A trustee discloses a set of physical properties to be verified by the truster. Examples of such properties for a device are CPU processing speed, memory capacity, transmission rate, signal strength, location of sensor, and physical security. Examples of properties associated with a human user are age, gender, education level, specialization, credentials, and so on. Experience is based on the set of events that had occurred in the past within a certain period of time in which the trustee was involved and that the truster has recollection about. For a device, this can be incidents like number of defects encountered, tamper occurrences, collected data quality, and alarms and control signals responsiveness. For a human user, this could be decisions made in the past, task execution time taken, finesse demonstrated, and so on. Recommendations are provided by trusted third-parties who have knowledge about the trustee with respect to the role context rc. Recommendations in case of a device can be provided by other organizations that have used the device under similar circumstances. For a human user, recommendations, for example, can be provided by an organization that he was worked with in the same (or similar) role context rc.

Quantifying Properties: Each role in an organization requires certain properties of a user. The properties are scored based on information provided by the user to the system at access request. Each role R is associated with a set of positive properties, $PS_R = \{ps_1, ps_2, \ldots, ps_n\}$, and negative properties $NE_R = \{ne_1, ne_2, \ldots, ne_n\}$, collectively called the role properties. Each positive and negative property is associated with a weight, determined by the organizational policy, that reflects its importance with respect to the role R. Let $w_{ps_1}, w_{ps_2}, \ldots, w_{ps_n}$ be the weights of the positive properties, where $w_{ps_i} \in [0,1]$ and $\sum_{i=1}^{n} w_{ps_i} = 1$. Let $w_{ne_1}, w_{ne_2}, \ldots, w_{ne_n}$ be the weights for negative properties, with $w_{ne_i} \in [0,1]$ and $\sum_{i=1}^{n} w_{ne_i} = 1$.

Let the set of properties possessed by a user B be $UP = up_1, up_2, \ldots, up_n$. Let $p_B = \{UP \bigcap PS_R\}$ be the set of positive properties for the user that are relevant for the role, and $n_B = \{UP \bigcap NE_R\}$ be the set of negative properties. Let w_{ps_i} be the weight of the positive property $p_{B_i} \in UP \bigcap PS_R$, and w_{ne_i} be the weight of the negative property $n_{B_i} \in UP \bigcap NE_R$. Let $m = |UP \bigcap PS_R|$, and $n = |UP \bigcap NE_R|$. The contribution of the user's properties towards its trust is represented by (b_P, d_P, u_P) where b_p, d_p, u_p denotes the belief that the set of properties contribute towards enhancing the opinion about trustworthiness of the trustee, the disbelief that the properties do so, and the uncertainty respectively. Each b_p, d_p and u_p is $\in$ [0, 1] and $b_P + d_P + u_P = 1$. The values of b_P,d_P and u_P are computed using the following formulae:

$$b_P = \frac{\sum_{i=1}^{m} w_{ps_i}}{\sum_{i=1}^{m} w_{ps_i} + \sum_{i=1}^{n} w_{ne_i}}; \quad d_P = \frac{\sum_{i=1}^{n} w_{ne_i}}{\sum_{i=1}^{m} w_{ps_i} + \sum_{i=1}^{n} w_{ne_i}}; \quad \text{and } u_P = 1 - b_P - d_P.$$

Quantifying Experience: We model experience in terms of the number of events encountered by a truster A regarding trustee B in particular context within a specific period of time $[t_0, t_n]$. The time period $[t_0, t_n]$ is equally divided into a set S_i of n intervals, $S_i = \{[t_0, t_1], [t_1, t_2], \ldots, [t_{n-1}, t_n]\}$. The intervals overlap at the boundary points only. The truster A keeps a history file of events performed by the trustee B within these intervals. Within each interval $[t_j, t_{j+1}] \in S_i$ where $j \in \mathbb{N}$, there exists a (possibly empty) set of events that transpired between the user and the system. Events that occurred in the distant past are given less weights than those that occurred more reecntly. We evaluate the experience component of trust, given by the triple (b_E, d_E, u_E), where b_E, d_E and u_E have the same connotation as for properties, in the same manner as in [8].

Quantifying Recommendation: Recommendations play major role on the trust evaluation when the truster does not know much about the trustee. Truster obtains recommendations from one or more recommender that claim to know about the trustee with respect to the particular roles. The recommendation is evaluated based on the recommendations returned by recommender M about B as well as the trust relationship between truster A and the recommender M in providing a recommendation about trustee B. Again we use the same procedure as in [8] to evaluate the recommendation score for the trustee based on a set of recommendations. The recommendation score is given by the triple (b_R, d_R, u_R) with each component having the same connotation as in the evaluation of properties.

Computing Trustworthiness: Using the same ideas as in [8] the trust evaluation policy of the truster is represented by the triple ${}_AW_B^{rc} = (W_P, W_E, W_R)$ where $W_P + W_E + W_R = 1$

and $W_P, W_E, W_R \in [0,1]$. The trust relationship between a truster A and trustee B for a particular role context rc is then given by cross product:

$$(A \xrightarrow{rc} B) = ({}_A b_B^{rc}, {}_A d_B^{rc}, {}_A u_B^{rc}) = (W_P, W_E, W_R) \times \begin{pmatrix} b_P & d_P & u_P \\ b_E & d_E & u_E \\ {}_{AG}b_R & {}_{AG}d_R & {}_{AG}u_R \end{pmatrix}$$

The elements ${}_A b_B^{rc}, {}_A d_B^{rc}, {}_A u_B^{rc} \in [0,1]$, and ${}_A b_B^{rc} +_A d_B^{rc} +_A u_B^{rc} = 1$. After evaluating the trust of the properties, experience, and recommendation factors as earlier the trust value is computed as: $T_{au} = \dfrac{{}_A b_B^{rc} +_A u_B^{rc}}{{}_A b_B^{rc} +_A d_B^{rc} +_A u_B^{rc}}$. The value T will be in the range of [0,1]. The value closer to 0 indicates low trust value of user B with respect to role R, while the value closer to 1 indicates very high trust value of user with respect to role R.

3 The Trust-Based RBAC Model

We adapt the graph-theoretic approach proposed by Chen and Crampton [5] to define the access control model. The set of vertices $V = U \cup R \cup P$ correspond to the following RBAC entities: (i) Users (U), which can be either human (U_h) or intelligent device (U_d); (ii) Roles (R), which can be categorized to human role (R_h) and device role (R_d), and (iii) Permissions (P), which can be categorized to human permission (P_h) and device permission (P_d). The set of edges $E = UA \cup PA \cup RH_a \cup RH_u$ constitutes of the following: (i) User-Role Assignment (UA) = $(U_h \times R_h) \cup (U_d \times R_d)$ (ii) Permission-Role Assignment (PA) = $(R_h \times P_h) \cup (R_d \times P_d)$ (iii) Role Hierarchy (RH) = $((R_h \times R_h) \cup (R_d \times R_d)) \times \{a,u\}$ consisting of (i) the activation hierarchy (RH_a) = $\{(r,r') : (r,r',a) \in RH\}$, and (ii) the permission usage hierarchy (RH_u) = $\{(r,r') : (r,r',u) \in RH\}$ [(i)]

Trust value for each user is calculated based on the role he has performed previously. The information about the roles the user has performed peviously is stored in the User Role History. The values of trust can be changed from time to time based on user activities. Negative activities such as, committing the fraud in the can decrease his trustworthiness. The calculation process is described in section 2. The system administrator assigns trust constraints in the form of a *trust interval* to roles, permissions, and other associations between entities based on different characteristics of each model. Trust interval is an interval $[l,1]$, where l is the lowest trust value that each role, permission or association is active. (Basically the minimum trust level is specified for each.)

Users of the senior role can perform the same set of duties as its junior role; hence a user who is assigned to the senior role to be more trustworthy than the user who is assigned to the junior role only. Based on this observation we assume that the trust value of the senior role always dominates the trust value of its junior roles. Figure 1 shows the components in our model.

We define the notion of activation path, usage path and access path as follows. An *activation path* (or *act-path*) between v_1 and v_n is defined to be a sequence of vertices $v_1, \ldots, v_n$ such that $(v_1, v_2) \in UA$ and $(v_{i-1}, v_i) \in RH_a$ for $i = 3, \ldots, n$. A *usage path* (or *u-path*) between v_1 and v_n is defined to be a sequence of vertices $v_1, \ldots, v_n$ such that $(v_i, v_{i+1}) \in RH_u$ for $i = 1, \ldots, n-2$, and $(v_{n-1}, v_n) \in PA$. An *access path* (or *acs-path*)

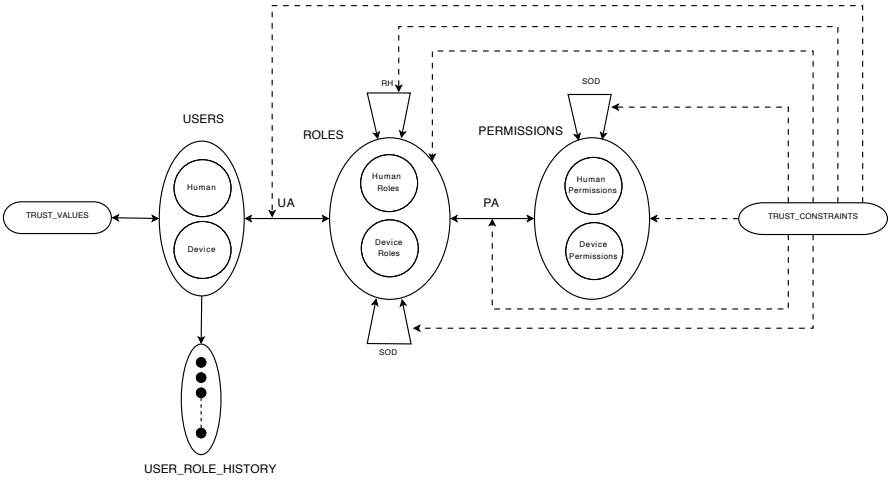

Fig. 1. Trust RBAC Model

between v_1 and v_n is defined to be a sequence of vertices $v_1, \ldots, v_n$, such that (v_1, v_i) is an act-path, and (v_i, v_n) is an u-path. We assume the existence of a trust domain $\mathcal{D}$. The value of trust in the domain can be any real number from zero to one.

The Standard Model Individual entities, namely, users, roles, and permissions are associated with trust values in the *standard model*. The trust values associated with the user describe how much the user is trusted to perform each specific role. The trust interval associated with a role specify the range of trust values with respect to that role which user has to acquire in order to activate the role. The trust interval associated with a permission specify the minimum trust value with respect to the current role of the user that he has to acquire in order to invoke the permission. The standard model requires that if a user u can invoke a permission p, then the trust value of u is in the trust interval associated with all other nodes in the path connecting u to p. The trust values for the user with respect to each role are denoted with a function $\mathcal{T} : ((U_h \times R_h) \cup (U_d \times R_d)) \rightarrow t \in \mathcal{D}$. The trust interval for role and permission are denoted with a function $\mathcal{L} : (R \cup P) \rightarrow [l, 1] \subseteq \mathcal{D}$. For user $u \in U, r \in R$, $\mathcal{T}(u, r)$ denotes the trust value of u with respect to r. For role $r \in R$, $\mathcal{L}(r)$ denotes the trust interval in which r is active. For $p \in P$, $\mathcal{L}(p)$ denotes the trust interval in which p is active. Given a path $v_1, \ldots, v_n$ in the labeled graph $G = (V, E, \mathcal{T}, \mathcal{L})$, where $E = UA \cup PA \cup RH_a \cup RH_u$, we write $\hat{\mathcal{L}}(v_2, \ldots, v_n) = \hat{\mathcal{L}}(v_2, v_n) \subseteq \mathcal{D}$ to denote $\bigcap_{i=2}^{n} \mathcal{L}(v_i)$. In other words, $\hat{\mathcal{L}}(v_2, v_n)$ is the trust interval in which every vertex $v_i \in R \cup P$ is enabled.

Authorization in the Standard Model is specified by the following rules: (i) A user $v_1 \in U$ may activate role $v_n \in R$ if and only if there exists an act-path $v_1, v_2, \ldots, v_n$ and $\mathcal{T}(v_1, v_2) \in \mathcal{L}(v_2)$; (ii) A role $v_1 \in R$ is authorized for permission $v_n \in P$ if and only if there exists an u-path $v_1, v_2, \ldots, v_n$ and $\mathcal{L}(v_1) \subseteq \hat{\mathcal{L}}(v_1, v_n)$; (iii) A user $v_1 \in U$ is authorized for permission $v_n \in P$ if and only if there exists an acs-path $v_1, v_2, \ldots, v_i, \ldots, v_n$

such that $v_i \in R$ for some i, $v_1, \ldots, v_i$ is an act-path, $v_i, \ldots, v_n$ is a u-path, v can activate v_i, and v_i is authorized for v'.

The Strong Model The *strong model* is used when not only the individual entities (users, roles, permissions) involved must satisfy the trust constraints, but the different relationships must also satisfy such constraints. For instance, consider the relation $(r, p) \in PA$. In this case, we not only have to take into account the trust values at which the role r can be activated and the trust values at which the permission p can be invoked, but we also must consider the trust values when r can invoke p. This requires specifying another function in the strong model. The trust constraints are denoted with a function $\mu : E \rightarrow [l, 1] \subseteq \mathcal{D}$. For $e = (v, v') \in E$, $\mu(v, v')$ denotes the trust interval in which the association between v and v' is active. If $(u, r) \in UA$, then $\mu(u, r)$ denotes the trust interval in which u is assigned to r. If $(r', r) \in RH_a$, then $\mu(r', r)$ denotes the trust interval in which r' is senior to r in the activation hierarchy. If $(r', r) \in RH_u$, then $\mu(r', r)$ denotes the trust interval in which r' is senior to r in the permission usage hierarchy. If $(r, p) \in PA$, then $\mu(r, p)$ denotes the trust interval in which p is assigned to r. Given a path $v_1, \ldots, v_n$ in the labeled graph $G = (V, E, \mathcal{T}, \mathcal{L}, \mu)$, where $V = U \cup R \cup P$ and $E = UA \cup PA \cup RH_a \cup RH_u$, we write $\hat{\mu}(v_1, \ldots, v_n) = \hat{\mu}(v_1, v_n) \subseteq \mathcal{D}$ to denote $\bigcap_{i=1}^{n-1} \mu(v_i, v_{i+1})$. Hence, $\hat{\mu}(v_1, v_n)$ is the trust interval in which every edge in the path is enabled.
Authorization in the Strong Model is specified by the following rules: (i) a user $v_1 \in U$ may activate role $v_n \in R$ if and only if there exists an act-path $v_1, v_2, \ldots, v_n$ and $\forall i = 2, \ldots, n \bullet \mathcal{T}(v_1, v_i) \in (\mathcal{L}(v_1) \cap \mathcal{L}(v_i) \cap \hat{\mu}(v_1, v_i))$; (ii) a role $v_1 \in R$ is authorized for permission $v_n \in P$ if and only if there exists an u-path $v_1, v_2, \ldots, v_n$ and $\mathcal{L}(v_1) \subseteq (\hat{\mathcal{L}}(v_1, v_n) \cap \hat{\mu}(v_1, v_n))$; (iii) A user $v_1 \in U$ is authorized for permission $v_n \in P$ if and only if there exists an acs-path $v_1, v_2, \ldots, v_i, \ldots, v_n$ such that $v_i \in R$ for some i, $v_1, \ldots, v_i$ is an act-path, $v_i, \ldots, v_n$ is a u-path, v_1 can activate v_i, and v_i is authorized for v_n.

The Weak Model The *weak model* is derived from the standard model. Recall that the standard model requires that each entity (users, roles, and permissions) in the authorization path be associated with a trust value and in order to be authorized to access other entities, the requester's trust value must be included in the trust interval of the entity he wants to access, together with other entities along the path. In the weak model, the entity v is authorized for another entity v' if the trust value of v is included in the trust interval of v'. There is no requirement that the intermediate nodes on the path satisfy the trust constraints. Like the standard model, the model is based on the labeled graph $G = (V, E, \mathcal{T}, \mathcal{L})$, where $V = U \cup R \cup P$ and $E = UA \cup PA \cup RH_a \cup RH_u$.
Authorization in the Weak Model is specified by the following rules: (i) A user $v_1 \in U$ may activate role $v_n \in R$ if and only if there exists an act-path $v_1, v_2, \ldots, v_n$ and $\mathcal{T}(v_1, v_n) \in \mathcal{L}(v_n)$; (ii) A role $v_1 \in R$ is authorized for permission $v_n \in P$ if and only if there exists a u-path $v_1, v_2, \ldots, v_n$ and $\mathcal{L}(v_1) \subseteq \mathcal{L}(v_n)$; (iii) A user $v_1 \in U$ is authorized for permission $v_n \in P$ if and only if there exists an acs-path $v_1, v_2, \ldots, v_i, \ldots, v_n$ such that $v_i \in R$ for some i, $v_1, \ldots, v_i$ is an act-path, $v_i, \ldots, v_n$ is a u-path, v_1 can activate v_i, and v_i is authorized for v_n.

Separation of Duties (SoD) Constraints prevent the occurrence of fraud arising out of conflicts of interests in organizations. Separation of duties ensure that conflicting roles

are not assigned to the same user or that conflicting permissions are not assigned to the same role.

Separation of Duty (SoD) comes in two varieties, namely, mutual exclusion relations between two roles and between two permissions, denoted by using SD^R and SD^P edges, respectively. The first variety is in order to guarantee that no user can be assigned to two conflicting roles. The second one is to guarantee that no role can be assigned two conflicting permissions. Since SoD is a symmetric relationship, the SD^R and SD^P edges are bi-directional.

The SoDs defined for the standard and weak models are expressed in terms of the graph $G = (V, E, \mathcal{T}, \mathcal{L})$, where $E = UA \cup PA \cup RH_a \cup RH_u \cup SD^R \cup SD^P$ and $V = U \cup R \cup P$. For these cases, the SoD is similar to the SoD constraints in traditional RBAC. These are given below.

SoD Constraints for the Weak and Standard Model

User-Role Assignment if $(r, r') \in SD^R$ then there are no two edges (u, r) and (u, r') such that $\{(u, r), (u, r')\} \subset UA$

Permission-Role Assignment if $(p, p') \in SD^P$ then there are no two u-paths of the form $r, v_1, v_2, \ldots, p$ and $r, v'_1, v'_2, \ldots, p'$

Sometimes in the organization we want the user who gain very high trust to be able to bypass the SoDs. For this we define the trust constraint for the separation of duties with a function $\delta : E \rightarrow [l, 1] \subseteq \mathcal{D}$. For $e = (v, v') \in SD^R \cup SD^P$, $\delta(v, v')$ denotes the trust interval in which the SoD constraint can be ignored. In particular,

- if $(r, r') \in SD^R$, $\delta(r, r')$ denotes the trust interval in which the role-role separation of duties constraint can be ignored;
- if $(p, p') \in SD^P$, $\delta(p, p')$ denotes the trust interval in which the permission-permission separation of duties constraint can be ignored.

The strong model is defined over the labeled graph $G = (V, E, \mathcal{T}, \mathcal{L}, \mu, \delta)$, where $E = UA \cup PA \cup RH_a \cup RH_u \cup SD^R \cup SD^P$ and $V = U \cup R \cup P$. The strong model allows specification of weaker forms of SoD constraints than those supported by the traditional RBAC. Specifically, it allows one to specify the trust interval in which the SoD constraints can be ignored.

SoD Constraints for the Strong Model

User-Role Assignment: if $(r, r') \in SD^R$ then there are no two edges (u, r) and (u, r'), corresponding to some user u, where $\mathcal{T}(u, r) \notin (\mathcal{L}(u) \cap \mathcal{L}(r) \cap \mu(u, r) \cap \delta(r, r'))$ and $\mathcal{T}(u, r') \notin (\mathcal{L}(u) \cap \mathcal{L}(r') \cap \mu(u, r') \cap \delta(r, r'))$;

Permission-Role Assignment: if $(p, p') \in SD^P$ then there are no two u-paths $r, v_1, v_2, \ldots, p$ and $r, v'_1, v'_2, \ldots, p'$, where $\mathcal{L}(r) \nsubseteq (\hat{\mathcal{L}}(r, p) \cap \hat{\mu}(r, p) \cap \delta(p, p'))$ and $\mathcal{L}(r) \nsubseteq (\hat{\mathcal{L}}(r, p') \cap \hat{\mu}(r, p') \cap \delta(p, p'))$.

4 Conclusion and Future Work

Traditional access control models are mostly not be suitable for pervasive computing applications. Towards this end, we propose a trust based access control model as an

extension of RBAC. We use the context-sensitive model of trust proposed earlier as the underlysing trust model. We investigate the dependence of various entities and relations in RBAC on trust. This dependency necessitates changes in the invariants and the operations of RBAC. The configuration of the new model is formalized using graph-theoretic notation. In future, we plan to incorporate other environmental contexts, such as space and time, to our model. We also plan to investigate conflicts and redundancies among the constraint specification. Such analysis is needed before our model can be used for real world applications.

References

1. E. Bertino, P. Bonatti, and E. Ferrari. TRBAC: A Temporal Role-Based Access Control Model. In *Proceedings of the 5th ACM Workshop on Role-Based Access Control*, pages 21–30, Berlin, Germany, 2000.
2. E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca. GEO-RBAC: A Spatially Aware RBAC. In *Proceedings of the 10th ACM Symposium on Access Control Models and Technologies*, Stockholm, Sweden, 2005.
3. S. Chakraborty and I. Ray. TrustBAC: Integrating Trust Relationships into the RBAC Model for Access Control in Open Systems. In *Proceedings of the 11th ACM Symposium on Access Control Models and Technologies*, Lake Tahoe, CA, June 2006.
4. S. M. Chandran and J. B. D. Joshi. *LoT-RBAC*: A Location and Time-Based RBAC Model. In *Proceedings of the 6th International Conference on Web Information Systems Engineering*, New York, NY, November 2005.
5. L. Chen and J. Crampton. On Spatio-Temporal Constraints and Inheritance in Role-Based Access Control. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security and Communications Security*, Tokyo, Japan, March 2008.
6. J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor. A Generalized Temporal Role-Based Access Control Model. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):4–23, January 2005.
7. I. Ray, M. Kumar, and L. Yu. LRBAC: A Location-Aware Role-Based Access Control Model. In *Proceedings of the 2nd International Conference on Information Systems Security*, Kolkata, India, December 2006.
8. I. Ray, I. Ray, and S. Chakraborty. An Interoperable Context Sensitive Model of Trust. *Journal of Intelligent Information Systems*, 32(1):75–104, February 2009.
9. I. Ray and M. Toahchoodee. A Spatio-Temporal Access Control Model Supporting Delegation for Pervasive Computing Applications. In *Proceedings of the 5th International Conference on Trust, Privacy & Security in Digital Business*, Turin, Italy, September 2008.
10. G. Sampemane, P. Naldurg, and R. H. Campbell. Access Control for Active Spaces. In *Proceedings of the Annual Computer Security Applications Conference* , Las Vegas, NV, USA, December 2002.
11. A. Samuel, A. Ghafoor, and E. Bertino. A Framework for Specification and Verification of Generalized Spatio-Temporal Role Based Access Control Model. Technical Report CERIAS TR 2007-08, Purdue University, February 2007.
12. M. Toahchoodee and I. Ray. On the Formal Analysis of a Spatio-Temporal Role-Based Access Control Model. In *Proceedings of the 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, number 5094 in LNCS, London, U.K., July 2008.
13. G. Ya-Jun, H. Fan, Z. Qing-Guo, and L. Rong. An Access Control Model for Ubiquitous Computing Application. In *Proceedings of the 2nd International Conference on Mobile Technology, Applications and Systems*, Guangzhou, China, November 2005.