Asian Journal of Information Technology 7 (2): 74-78, 2008
ISSN: 1682-3915

Context-Role Based Access Control Model for Ubiquitous Computing Environment

[1]Lin Li and [2]Tianjie Cao
[1]Department of Computer Science and Technology,
China University of Mining and Technology, Xuzhou 221008, China
[2]State Key Laboratory of Information Security,
Institute of Software of Chinese Academy of Sciences, Beijing 100080, China

Abstract: In ubiquitous computing environment user move frequently and access resources anytime and anywhere. Ubiquity and invisibility are the distinct characteristics of ubiquitous computing. Numerous entities are always unknown to local system. The security service should consider the factor of time, location and trust to enhance the security of interaction between each other. The exiting access control consider just one dimension, temporal, spatial or trust. In this study, we propose a new access control model supporting these three dimensions simultaneously. The proposed access control model can effectively support various ubiquitous computing environments.

Key words: Context, trust, time, location, access control, ubiquitous

INTRODUCTION

Weiser (1991) first introduced the term ubiquitous computing (The abbreviation is Ubicomp or UC). The main idea of ubiquitous computing is to embed computer into the living environment or tool of human, so as to make the computer invisible from users' sight and the users can focus on their task itself, instead of the computer. Pervasive computing embodying the integration of computer, communication and digital media technology makes it possible to integrate the physical world we are living in and the virtual world in the information space together as the whole (Xuan and Nguyen, 2006).

The ability to compute anywhere, anytime in any way is the distinct characteristic of ubiquitous computing. The birth of Pervasive computing brings forth the paradigm of a new computing model as well as security problems: Pervasive computing environment is an open environment in which principals collaborate spontaneously and unforeseeably. High enough trust is essential to ensure security among these principals in which pervasive computing pays more attention to the role of trust than traditional computing. Context (e.g., user's location, user's need, etc) changes dynamically. The authorization of user is required to be based on contextual information.

Lots of works have been done in the field of access control: Role-Based Access Control (RBAC), Context-Based Access Control (CBAC) and Trust-Based Access Control (TBAC). However, each of these approaches considers just one dimension, temporal, spatial or trust. They can not fulfill the security requirements mentioned above independently. Our CRBAC involves three factors simultaneously: we regard location, time and trust as contexts of our proposed access control model. Our CRBAC can effectively support various ubiquitous computing environments.

RELATED WORKS

Role Based Access Control (RBAC): Role Based Access Control (RBAC) (David *et al*., 2003) introduced the concept of role. Users are not directly associated with permissions, they are assigned to roles. Permissions are assigned to roles. Roles in RBAC are inherently subject-centric. So, it cannot be used to capture security relevant information from the environment which could have an impact on access decisions.

Generalized Role Based Access Control (GRBAC): Convington have proposed the Generalized Role Based Access Control (GRBAC) model. GRBAC extended traditional Role-Based Access Control by defining object roles and environment roles. Defining too many roles makes the system hard to maintain. It is not feasible in practice.

Corresponding Author: Li Lin, Department of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221008, China

Temporal RBAC (TRBAC): Roles would be unavailable sometimes. TRBAC (Joshi *et al*., 2002) is an extension of RBAC that has time constraint. It provides temporal dependencies among the enabling and disabling of different roles, expressed by means of role triggers. Role trigger actions may be either immediately executed, or deferred by an explicitly specified amount of time (Bertino *et al*., 2001).

Generalized Temporal RBAC (GTRBAC): GTRBAC is an extension of the TRBAC. TRBAC provides constraints only on role enabling and triggers. The GTRBAC model introduces the separate notions of the enabled and activated states of role and provides constraints and event expressions associated with both these states (Song-Hwa *et al*., 2006). GTRBAC does not consider location constraint.

Spatial RBAC (SRBAC): SRBAC (Hansen and Oleshchuk, 2003) defines the concept of Zone for location. Permissions are assigned to Zone in a role by a Location Permission Assignment List (LPAL) that is presented by matrix. In real environment, there are a lot of locations so that this model should have large matrix. It is also difficult to divide an area into Zones.

CONTEXT-ROLE BASED ACCESS CONTROL MODEL

NCBAC definition: We define U, P, S, C, UR, CR that represent the set of users, permissions, sessions, contexts, user roles and context roles. The proposed model consists of the following components:

- U, P, S ,C, R , UR,CR, , UA, CA, PA.
- U: A user is an entity whose access is being controlled.
- P (Permissions): Permission is an approval to perform an operation on one or more CRBAC protected objects.
- S: A role is activated for user during each session.
- C: C represents a set of context information in the system. Context information can be time, location, trust value, etc.
- UR (user roles): UR is equal to ROLE in traditional RBAC (Seon-Ho *et al*., 2006).
- CR (context roles) (Sen-Ho *et al*., 2006): The context role is used to capture security relevant context information about the environment for use in CRBAC. CR⊆UR.
- AR (Activated Role): Activated role is a mapping between user roles and context roles (Seon-Ho *et al*., 2006).

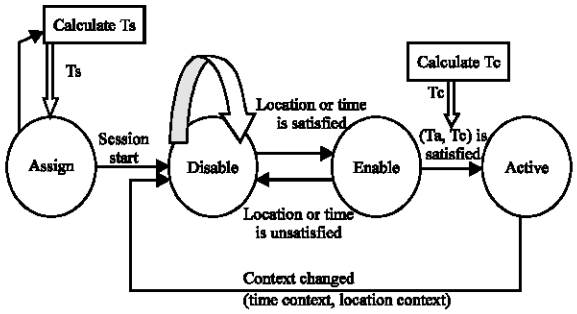

Fig. 1: The role states of the CRBAC

- UA (User_Assignment): UA⊆U×R. UA is the mapping that assigns a user role to a user.
- PA (Permission_Assignment): PA⊆P×R . PA is the mapping that assigns permissions to a role.
- User_Session (u: U) → 2^S the mapping of user u onto a set of sessions.
- Session_Users (s: Sessions) → U
- Session_Roles (s: Sessions) → 2^R. the mapping of session s onto a set of roles.
- Role_hierarchy: RH⊆R×R.
- Role_hierarchy: RH⊆R×R.
- Location_hierarchy: RH⊆R×R.
- CA (Context role_Assignment) CA⊆CR×C is the mapping that assigns a context role to a context.

Role states

Definition 1 CR (Context Role): CR={TiCR, LoCR, TrCR} CR represents a set of context roles. The context role is used to capture security-relevant context information about the environment for use in CRBAC. The context role can contain time-related context role, location-related context role, location-related context role, trust-related context role, etc (Fig. 1).

Definition 2 TiCR (Time-related Context Role): A time-related context role is a pair (r, time), where r is the user role name and time is the current time of the use role.

Definition 3 LoCR (Location-related Context Role): A location-related context role is a pair (r, location),where r is the user role name and location is the current position of the use role.

Definition 4 TrCR (Trust-related Context Role): A trust-related context role is a pair (r, trust value),where r is the user role name and location is the position of the use role.

NCBAC authorization: Location authorizations are policies defining the accesses that the users may own over the locations. Temporal authorizations are to limit

the period during which the authorization is valid. Trust Authorizations are to limit the entity access control whose trust value does not satisfy security policies. Formally, they are defined as follows.

Definition 5 L_Auth (Location Authorization): A location authorization is a pair ((s, r), l) where:

- s is a subject (user) who requests authorizations.
- r is the role which s can gain when he is in location l.
- l is a location.

A location authorization ((s, r), l) means that user s is authorized to gain the role r when he is in the location l. For example, (Alice, Supermarket, Consumer) denotes that Alice is consumer role when she is in the supermarket.

Definition 6 Ti_Auth (Temporal Authorization): A temporal authorization is a quadruple ((s, r), t ((t^i_s, t^i_e], (t^o_s, t^o_e]), n) where

- Time interval [t^i_s, t^i_e] is entry duration during which a subject can gain the role r for access the object o.
- Time interval [t^o_s, t^o_e] is exit duration during which a subject will be revoke the role, -$t^o_s \geq t^i_s$ and $t^o_e \geq t^i_e$.
- n is the number of accesses that the subject can exercise. n$\in$ [1, ∞).

Example: ((Alice, consumer), (8, 18], (18,8], 100).

Alice is allowed to get the role "consumer" during the period (8, 18] and will be revoked the role during the period (18,8], for a maximum number of one hundred times.

Definition 7 Tr_Auth (Trust Authorization): A trust authorization is a pair ((s, r), T) where

- s is a subject (user) who requests authorizations.
- r is the role which should be activated when the T satisfies the security policy.
- T is the trust value of a subject. Trust value is up to the subject attributes themselves and changes according to the context. T = F (Ts, Tc) = F (f_1 (A_1, A_2, …, A_n), f_2 (C_1, C_2,…, C_3)) where
- Ts is the trust value result from subject attributes. It is a function of subject attributes. Ts = f_1 (A_1, A_2, …, A_n), where A_1,A_2,A_3,… A_n are subject attributes. Ts is static. This value is mapping to the User Role (UR). Ts→ User Roles.
- Tc is the trust value result from contexts. It is a function of contexts. Tc = f_2 (C_1,C_2,C_3…,C_n), C_1, C_2, C_3, … C_n are context information such as C_1 = TC: (Time Context), C_2 = LC: (Location Context), C_3= TC: (Trust Context). This value is mapping to the Context Role (CR). Tc→ Context Roles. Tc is dynamic. Tc is evaluated newly once contexts changed.

Definition 8 CA (CRBAC Authorization): The CRBAC authorization is defined as C_Aut (Tr_Auth (L_Auth, Ti_Auth))

- L_Auth is a location authorization.
- Ti_Auth is a temporal authorization.
- Tr_Auth is a trust authorization.
- C_Auth is a context authorization.

The CRBAC Authorization Algorithm may be written as follows, where now time. Entry is the current time when the user entry a location, NowTime.exit is the current time when the user exit a location, L is set of locations.

Algorithm 1 activated. role

Input: A_1,A_2,…,A_n, Location information, time
Output: activated. role
1: Ts = f1 (A_1,A_2,…,A_n)
2: **if** Ts>=T_{min} **then**
3: assign: UR={r_1,r_2,… r_n}
4: session start
5: **else**
6: Ts = 0
7: UR = NULL
8: no session would be activated
9: **end if**
10: **if** Ts≠0 and UR≠NULL **do**
11: **for** each location∈L
12: **for** each NowTime.entry ⊆ (t^i_s, t^i_e] and NowTime.exit⊆ (t^o_s, t^o_e] **do**

Algorithm 1 Continue

```
13:          C1 = 1
14:          C2 = NowTime.entry
15:          C3 = NowTime.exit
16:          Tc = f2 (C1,C2, C3)
17:          Assign: CR = {r1,r2,... rn}
18:          Enable CR
19 :         T =  F (Ts, Tc)
20:         if T satisfy the security policy
21:             CR .r is activated
22:             activated.role = CR .r
23:         else
24:             activated.role = 0
25:             no role would be activated
26:         endif
27:       When locationor NowTime.entry or NowTime.exit changed
28:       end for
29:     end for
30:   end while
31: end if
32: Return {activated.role| activated.role∈ CR}
```

Authorization rule

Definition 9 (CRBAC authorization rule): The CRBAC authorization Rule is defined by the tetrad (s, C_Auth,o, m) where

- s is a subject (user) who requests authorizations.
- C_Auth is CRBAC Authorization.
- o is a authorized object.
- m is access mode which can be read or write.

A role can be associated with a variety of contexts, but rights of these context roles would be different from each other. For example:

There are three roles, they are: consumer, sales manager, human resource manager. There are three locations, they are: supermarket hall, Sale Manager's Office, Human Resource department. Different roles in the same location would own different rights. Meanwhile same role in different locations would be authorized different rights.

Role 1: Consumer. In a supermarket hall, all consumers are allowed to access the resources that reserved for public use, such as brochures.

Role 2: Sales Manager. Sales Managers take charge of making sales planning, buying and selling, price adjustment, contract negotiations and so on.

Role 3: Human Resource manager: human resource managers manager all clerks and determine employing or laying off them.

There are three temporal interval, they are: 8:00 a.m.-18:00 p.m., 18:00-20:00 and 20:00-8:00 a.m. Different roles in the same temporal interval would own different rights. Meanwhile same role in different temporal intervals would be authorized different rights.

Consumers may go shopping during 8:00-18:00 a.m. They should leave the shopping mall before 18:00. All clerks work during 8:00-20:00. They should leave the shopping mall before 20:00.

CR1 (consumer1, supermarket hall, 8:00-18:00 a.m.): This role owns the rights to go shopping, enjoy all the activities held by the company.

CR2 (consumer1, Human Resource department, 8:00-18:00 a.m.): This role would not have certain rights which are owned by the human resources manger when he is in the department at the same time. For example, consumer1 can not be allowed to determine employ or lay off someone.

CR3 (consumer1, Human Resource department, 18:00-20:00 a.m.): This role will not be activated. Because role consumer should leave the shopping mall before 18:00.

CR4 (Sales Manager, Sale Manager's Office, 18:00-20:00 a.m.): This role gets all rights that a sale manager owns.

CR5 (Sales Manager, Human Resource department, 18:00-20:00 a.m.): This role may not own the right mentioned above, but just some common rights such as look at some shopping brochures. And he also does not own the right such as determining employing or laying off clerks.

When one person (such as a sales Manager named Alice) come into a supermarket, we can calculate the Trust value (Ts) according to the subject characteristics. And he will be assigned a role set, such as {consumer, sales manager, human resource manager, ...}. When he reach the Sale Manager's Office, we calculate the Trust value (Tc) according to contexts. Once the Trust value (Ts, Tc) satisfies the security policy, Sales Manager role is activated. The person (Alice) will get the Sales Manager's rights. Whenever the contexts changed, Tc is calculated once more. The user (Alice) will get more services or loose some service.

CONCLUSION

In this study, we introduced a Context Role Based Access Control (CRBAC) model for ubiquitous computing environment. There are three distributions of our paper: firstly, calculating trust value as one kind of context. Secondly, combining trust with location and temporal dimensions organically. Thirdly extending the traditional role based access control by context role to support various ubiquitous computing environments effectively. In future work, we plan to extend our CRBAC to support constraints, such as separation of duty constrains, location constraints, temporal constraints and so on. We also plan to consider trust level and hierarchy and CRBAC hierarchy.

ACKNOWLEDGEMENT

This work is supported by the Jiangsu Provincial Natural Science Foundation of China (BK2007035) and the Science and Technology Foundation of CUMT (0D0601624287).

REFERENCES

Bertino, E., P.A. Bonatti and E. Ferrari, 2001. TRBAC: A Temporal Role-Based Access Control Model. ACM. Trans. Inform. Sys. Sec., 4: 191-223.

David, F., D. Ferraiolo, R. Kuhn and R. Chandramouli, 2003. Role-Based Access Control, Artech House.

Hansen, F. and V. Oleshchuk, 2003. SRBAC: A Spatial Role-based Access Control Model for Mobile Systems. In: Proceedings of Nordec, Gjovik, Norway.

Joshi, J.B.D., E. Bertino and A. Ghafoor, 2002. Hybrid Role Hierarchy for Generalized Temporal Role Based Access Control Model. In: Proceedings of the 26th Annual International Computer Software and Application Conference.

Le Xuan Hung and Nguyen Ngoc Diep, 2006. A Flexible and Scalable Access Control for Ubiquitous. Computing Environments. ISI 2006, LNCS 3975, Springer- Verlag Berlin Heidelberg, pp: 688-689.

Seon-Ho Park, Young-Ju Han and Tai-Myoung Chung, 2006. Context-role based access control for context-aware application. Springer-Verlag. Berlin. Heidelberg, pp: 572-580.

Song-hwa Chae and Wonil Kim *et al.*, 2006. Role-based access control model for ubiquitous computing environment. Springer-Verlag. Berlin, pp: 354-363.

Weiser, M., 1991. The Computer for the 21st Century, Scientific American, pp: 94-100.