논문 2007-02-81

Dynamic Access Control for Personalized Environment in Ubiquitous Computing

Yuna Kim*, IlShik Shin, Sung Je Hong, Jong Kim

Abstract : In an ubiquitous environment, for controlling user access according to environment of users, a number of access control models enforcing dynamic environment of users have been proposed. However, they do not support personalized environments of each user and have a run-time overhead of searching active roles. In this paper, we propose a new model, PE-RBAC, that extends the RBAC architecture by addition of a *personalized environment* component as a constraint to accommodate dynamic and mobile users. In this model, a dynamic role activation is presented by using a new role-to-environment structure instead of the conventional role hierarchy, which makes it efficient to find the active roles according to a user's personalized environment.

Keywords : ubiquitous computing, mobile user, role-based access control, dynamic access control, personalized environment

I. Introduction

The advance of small-sized handheld devices enables the users to enjoy various services at anywhere and anytime. The users need to be offered seamless services with maintenance of a connected session, even if their environments are changed. Meanwhile the service providers need to decide whether a permission granted to the user is maintained or not, according to change of the user's environment. For example, if a news service opens from 9 a.m. to 5 p.m. to users, a user should not be able to enjoy the service any more at off time even if the user has an already connected session. Also an outdoor user should not be able to use a home service available only at home.

For supporting the challenges, many extended RBAC (Role-Based Access Control) models have been proposed for enforcing an access control with dynamic environments of users. Most of the models have weaknesses such that it is impossible to search the active roles at once, and personalize the environment constraints associated with the user. Despite of the same service, their users should be able to set their own environment constraints as keeping the rules established by the service. For example, if an education service can be accessed for one hour everyday with a student-role, we can think the following situation that a user A wants to use the

* Corresponding Author
Manuscript received Dec. 19, 2007; accepted : Jan. 24, 2008.
Yuna Kim, Sung Je Hong, Jong Kim: POSTECH
IlShik Shin : Network Technology Lab., KT
※ This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement) (IITA-2007-C1090-0701-0045)

service from 6 p.m. to 7 p.m. but another user B wants from 9 a.m. to 10 a.m. Their time zones are the environment constraints for activating the student-role, and each constraint is personalized to each user. The student-role must be activated to user A (or B) at his/her user-specific environment. We need to support this situation.

In this paper, we propose an extended RBAC model, PE-RBAC, to support dynamic access control for various service provisions for mobile ubiquitous users. We add a new component, personalized environment of a user as a constraint to select relevant roles among the conventional roles. A user-to-environment assignment relation of the proposed model can support the user-specific environment. In addition, the dynamic role activation is presented using a role-to-environment structure instead of the conventional role hierarchy to make it efficient to search the active roles according to the current environment.

1. Related Work

After the RBAC model attracted attention as a next superior alternative to the traditional access control, many researchers have proposed dynamic RBAC models which dynamically change the user's permissions by the user's environmental information. We categorize these dynamic RBAC models into dynamic role permission (DRP) and dynamic role activation (DRA) mechanisms by which component is dynamically changed. DRP means the dynamic change of permissions assigned to roles, and DRA means the dynamic change of active roles assigned to users. This section describes DRP and DRA models.

There are two models in DRP, agent-based RBAC [7] and spatial RBAC (SRBAC) [8]. The agent-based RBAC consists of an abstract role set, a context rule, and an agent that derives actual access rights by context rule and context information. The context information is generated dynamically by the agent. The actual role is decided dynamically by rules and context information, such as user location, time, and so on. SRBAC is an extension of RBAC based on the notion of spatial role, intended as a role that automatically activates its permissions when the user is in a given position. SRBAC adds a new component Locations to RBAC and the component is placed between roles and permissions. The permissions of a role are decided dynamically according to the location information.

As in DRA model, GTRBAC [9,10] and GEO-RBAC [6] share the same idea. GTRBAC uses the duration or time, and GEO-RBAC uses the location as environment constraints. GTRBAC includes periodic enabling of roles and temporal dependencies among roles. GTRBAC is a hybrid model of DRP and DRA, and GEO-RBAC proposed after GTRBAC removes the dynamic role-permission relation. GEO-RBAC uses role schema and role instance. The role schema defines some common properties of a set of spatially aware organizational functions. The properties include a common name, space constraints, and the type of logical locations. The role instance is a role accepting the constraints defined at schema level. DRBAC [11,12] consists of a role and permission state machine. The role state machine travels all roles assigned to a user and selects appropriate roles as active roles. The permission state machine checks the object permissions about which permissions this object can serve at this environment.

DRA models are widely accepted among dynamic RBAC models than DRP models due to the simpleness. However, DRA models have a search time overhead for finding active roles. DRA models use <user, role> and <role, environment-constraints> relations. To find the active roles, we need to repeat on all roles of the user to check all of the environment constraints assigned to each role. The process does not stop even after a matched role is found. This makes big overhead in the process of active role search.

2. Motivation

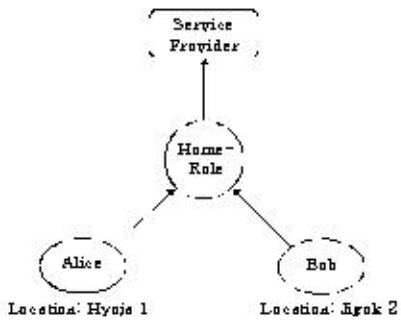

Figure 1. Same role at two different user-specific environments

In an ubiquitous environment, relations between roles and environments cannot be fixed. Some users may want to activate a role when they belong to an environment, but others who are assigned to the same role may want at another environment. Fig. 1 shows an example. A service provider provides a home service that can be consumed by the user who is assigned to a home role. The home role should be activated only when the user is at home. Here, an environment *home* is a user specific information, such as *Hyoja1* and *Igok2*. Also the education service as stated previously can be another example. Therefore, we need to personalize the environment constraints to each user. However, both previous DRP and DRA models use the predefined environment constraints, which are not suitable for a mobile user in ubiquitous environment but for a user in single organization. In other words, they have a weakness that the service time is fixed such as from 6 p.m. to 9 p.m., so all users can use the service for the time.

In addition, most of DRA models have running time overhead when searching active roles appropriate to an environment, except DRBAC model [11,12]. Absence of a direct relation between user and environment-constraint makes running time slower.

Handheld mobile devices always require low power and memory consumption. In DRBAC model, an access control agent on a user device manages his/her own state machines of role hierarchy and permission hierarchy, which increase consumption of power and memory in the device.

Consequently, requirements of our system can be established as follows:

- Personalized environment: The environment constraints need to be personalized to each user.
- Fast search of active roles: Active roles should be found at once for a given environment.
- Low power/memory consumption: We must minimize consumption of power and memory in a handheld device for access control.

II. Proposed Model: PE-RBAC

In this section, we describe the proposed model, PE-RBAC, in detail. We introduce a new component "environment" at first, and then describe the model and an algorithm of searching active roles. Finally, an illustrative scenario example using the model is presented.

1. Environment

We define a new component *"environment"* as environment constraints. The environment constraint personalized to a specific user is written in italic font style.

Using the example of the education service as stated earlier, "from 6 p.m. to 7 p.m." is the *environment* associated with the student role for the user A, "from 9 a.m. to 10 a.m." is the *environment* for the user B. If the current time is 8 p.m., 8 p.m. is just the environments of the user *A* and the user *B*, not a constraint.

In this model, the *environment* is composed of a set of range constraints, such as duration or location boundary position. The following definition formalizes the *environment*.

Definition 1 : *Environment*

- *rc*, *e* : a range constraint *rc* which is an element of an environment set, *e*

- *fully accepted (range)* : if a range of rc_j covers a range of rc_i, we can say that rc_i is fully accepted by rc_j which is represented as $rc_i \preccurlyeq rc_j$.
- *intersection (range)* : the intersected range of rc_i and rc_j is represented as $rc_i \wedge rc_j$.
- *fully accepted (environment)* : for environment sets e_i and e_j, if $\forall rc \in e_i \rightarrow \exists rc' \in e_j$, $rc \preccurlyeq rc'$, then we can say e_i is *fully accepted* by e_j represented as $e_i \preccurlyeq e_j$.
- *intersection (environment)* : for environment sets e_i and e_j if $\exists rc \in e_i$, $\exists rc' \in e_j$, $rc \wedge rc' \neq \perp$, then we can say e_i and e_j have an intersected environment $e_i \wedge e_j$.

The *environments* of a user should be disjoint each other for simplifying the role activation. The overlapped *environments* should be divided as long as having intersection at the service assignment phase, because they can make overhead. Each divided *environment* fragmentation becomes a new *environment* and is assigned to the overlapped roles. After all, each *environment* determines the candidates of an active role set of a user. In other words, given the *environment* of a user, the active role set of the user is decided at once.

2. Dynamic Access Control for Personalized Environment

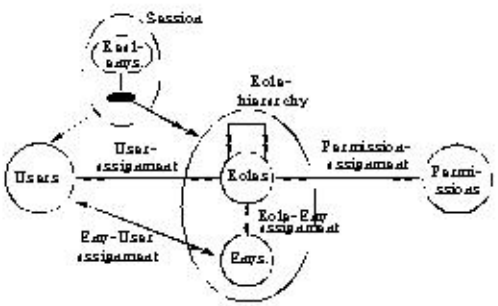

Figure 36. PE-RBAC Model

Our model is based on the RBAC96 model of Sandhu et al. [1-5,13]. We can see the PE-RBAC model at a glance in Fig. 2 where a double-headed arrow means one-to-many relation and the single-headed arrow means one-to-one relation. *Real-envs*: is a physical environment of a user associated with a session, and *Envs*: is logical environments assigned to roles, as well as, to a user. A role can be assigned to many roles and an *environment* can be assigned to many roles (*RA*), which is same as *UA* or *PA* relation. The notable point of this model is the *environment-to-user* relation. As the *environment* component is directly connected with user, the dynamic of a user is enforced more simply and easily.

The following definition formalizes the proposed model.

Definition 2 : PE-RBAC

- *U, R, P, S, PA, UA, RH,* and *user* are the same as in the RBAC96 model.
- *E* : the set of environments
- *RA* : $RA \subseteq R \times E$, a many-to-many role to environment assignment relation
- *user_envs* : $U \rightarrow 2^E$, a function mapping each user u to a set of environments $user\_envs(u)$, $\forall u \in U$ and $\forall e_i$, $\forall e_j \in user\_envs(u)$ $(i \neq j) \rightarrow e_i \wedge e_j = \varnothing$ (disjoint property)
- *env_roles* : $E \rightarrow 2^R$, a function mapping each environment $e \in E$ to a set of roles $env\_roles(e)$
- *real_env* : the current real environment information $real\_env(s)$ of user $user(s)$ with a session s, which changes with time
- *env*: $S \rightarrow E$, a function mapping each session $s \in S$ to the single environment $env(s)$, which change with time, and $real\_env(s) \preccurlyeq env(s)$, that is, $env(s)$ represents the current logical environment information of the user $user(s)$ with the session s for the system.
- *roles*: $S \rightarrow 2^R$, a function mapping each session s_i to the set of roles $roles(s_i) \subseteq \{r \mid r \in env\_roles(env(s_i))\}$, which can change with time, and session s_i has the permission $\cup_{r \in roles(s_i)} \{p \mid (\exists r' \leq r)[(p, r') \in PA]\}$

Each session is a mapping of a user to many roles. Also, each session is a mapping of a user to exactly one *environment*. The roles are assigned to this *environment*. Each session has a real environment of a user, the session owner. The real environment is collected at the authorized user agent in the user's handheld device. A user may have multiple sessions open at the same time with

several services. In this case, each session may have different combination of active roles, but the real-environment of the sessions would be same. When a user creates a session, the user agent sends information of the user's real environment to the system which authorizes the user. The system finds a user's *environment* which fully accepts the user's real environment. Then the user can activate one or more roles assigned to the *environment*. We assumed that each user already registered information of mapping between physical and logical environments into the system. For example, a user *A* registers *home* as *Gangnam, Seoul*.

Role hierarchies are a general means for structuring roles to reflect organization's lines of authority and responsibility [4]. Sandhu et al. [5] proposed the separation of role activation hierarchies and permission usage hierarchies. In PE-RBAC, the role hierarchies are used only as permission usage hierarchies, and the role activation hierarchies are substituted with a role-environment structure with which we can see the candidates of active roles assigned to the environment. If an activated role has a junior role which is not allowed at the same environment, the junior role cannot be activated. The activated senior role inherits only permissions from junior role.

3. Role Activation Mechanism

```
get cval_env(s) from the user agent of user user(s)
with session s
for ∀c ∈ user_envs(user(s)) do
    if cval_env(s) ≪ c then
        env(s) ← c and break
active_roles(s) ← env_roles(env(s))
```

Figure 3. Active role search algorithm in PE-RBAC

When a user's real environment is changed, the user agent in the user's handheld device generates a session-oriented event to inform the access control system of the change. The access control system selects new active roles to be granted to the session with the new *environment* using the active role search (ARS) engine. Fig. 3 shows the pseudo code of the algorithm for searching the active roles in PE-RBAC. Disjoint property of environments makes the search process more easy and simple. The ARS engine needs to find only one *environment* that fully accepts the new real environment. Once the matched *environment* is found, the engine can get candidates of active roles immediately. If no *environment* of the session owner fully accepts the new real environment, the user cannot activate any role or can activate only the basic role which is always able to be activated.

4. Illustrative Example

We show an illustrative example of the way to convert RBAC to PE-RBAC structure. Our proposed model targets an ubiquitous environment where users want seamless services at anywhere and anytime. In this environment, there might be various services such as education, entertainment, health-care, and home services that need to be merged.

Suppose that there is a user, Alice, a high school student. She is provided four services by a service provider (SP). Her family makes a contract with the SP for *home service* and *outdoor-home service*, and her school does for *education service*, and she herself does for *individual service*. The *home service* provides indoor home services including home education, home doctor, home entertainment service, and so on. The *outdoor-home service* provides outdoor services, such as home security, visitor checking and so on. The *individual service* includes entertainment, street information, game, and so on. Each service is assigned to each different role, a *family role* for the home service, an *outdoor-family role* for the outdoor-home service, an *individual role* for the individual service, and a *student role* for the education service.

Each service also has an appropriate *environment* for being served. The home service can be served only at home, and the education service at school only in class. Alice can be served the individual service at anywhere except

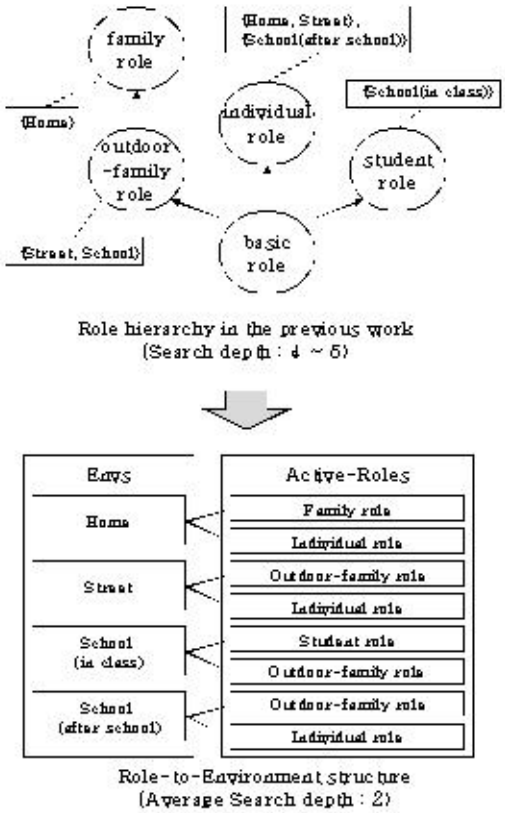

Figure 4. Converting of the role hierarchy in the previous work to Role-to-Environment structure for PE-RBAC

in class. She can use the outdoor-home service anywhere, since an indoor user can use the outdoor-home service with the family role that inherits the outdoor-family role. In this situation, the SP needs to provide a secure connection for the outdoor-home service using the outdoor-family role not the family role. Consequently, the separation of the outdoor-family role and the family role is reasonable.

Our model uses a new PE-RBAC structure instead of the role hierarchies for searching candidates of active roles, which leads faster search time. As an example, we show Alice's role hierarchies and the related *environments* on the upper side of Fig. 4. There are five environments and four roles except the basic role. We can convert this role hierarchy into *role-to-environment structure* for PE-RBAC. We apply the disjoint property to the environments for picking unique *environments*. For example, {Street, School} *environment* and {Home, Street} *environment* have an *intersected environment* of {Street}. Hence we divide them into three *environment* of {Street}, {School}, and {Home}. In this way, we can obtain total four disjoint *environment* {Home}, {Street}, {School(in class)}, and {School(after school)}. After that, candidates of active roles are assigned to each disjoint *environment* as shown on the lower side of Fig. 4. When receiving an event related to real environment, the ARS engine selects a pertinent *environment* fully accepting the real environment, and extracts its candidates of active roles immediately. In case of using the previous role hierarchies, searching active roles spends 4 or 5 times of comparisons on the average, but, in PE-RBAC, the average time is just 2.

Ⅲ. Search Time Analysis

PE-RBAC model is proposed for the personalized environment and the low search time overhead. In this section, we analyze the time overhead for searching active roles of our proposed model compared to the conventional DRA models.

For analyzing the active role search depth of PE-RBAC, we need some assumptions. First, PE-RBAC model has the disjoint property of environments, but the conventional DRA models do not. Thus, for a fair comparison, we assume environments of both models are disjoint. Second, each role has at least one environment and every environment has at least one role. Finally, the linear search method is used, that is, when searching one item out of p items, the average search time is $p/2$.

Let a user assign to n roles with m environments. The role to environment relation has an average x roles per environment and y environments per role. Definitely, $1 \leq x \leq n$ and

$1 \leq y \leq m$.

On PE-RBAC, searching active roles is finding only one appropriate environment.

$$PE\text{-}RBAC: \text{ average search depth} = \frac{m}{2} \quad (1)$$

On the conventional DRA models, each assigned role should be checked whether it is appropriate or not. Because the number of average environments per role is y, the average search time is $y/2$ for judging one role in the case of an appropriate role. Otherwise, all environments which are assigned to the role need to be checked. Thus, in this case, the average search time is y. Since the number of average roles per environment is x, we can expect x appropriate roles at any environment and get the total search time of both appropriate and inappropriate roles. The total search time is the sum of both cases of the appropriate and inappropriate roles. The average search time of previous DRA models is as follows.

$$DRA: \text{ average search depth} = \frac{xy}{2} + (n-x)y = (n-\frac{x}{2})y \quad (2)$$

On Equation (2), $xy/2$ is the case of appropriate role, and $(n-x)y$ is the case of inappropriate role. For comparing it with PE-RBAC model, we transform the Equation (2) to the equation of m. We can draw a

Figure 5. Bipartite graph of roles and environments

bipartite graph with roles and environments as shown in Fig. 5.

Nodes are roles and environments, and edges are the relations between roles and environments. Let p be the number of edges of the bipartite graph. Then, $x=p/m$ and $y=p/n$. Therefore, we get $y=mx/n$. Applying this to the Equation (2) is as follows.

$$(n-\frac{x}{2})y = (1-\frac{x}{2n})mx \quad (3)$$

The right hand side of Equation (3) is the quadratic function of x with the maximum value at $x=n$ and the roots on $x=0$ and $x=2n$. The quadratic function is monotonously increasing where $1 \leq x \leq n$, and the bound of Equation (3) is as follows.

$$(2-\frac{1}{n})\frac{m}{2} \leq (1-\frac{x}{2n})mx \leq \frac{mn}{2} \quad (4)$$

$$\frac{m}{2} \leq (2-\frac{1}{n})\frac{m}{2} \leq (1-\frac{x}{2n})mx \quad (5)$$

From the Equation (5), PE-RBAC model has lower search time than the conventional DRA models. When there is only one role ($n=1$, $x=1$), both PE-RBAC and the conventional DRA models have the same active role search time $m/2$. The low search time of our model is originated from user personalized environments and the disjoint property of environments.

IV. Conclusions

In an ubiquitous environment, service providers want to provide various services, and users want services in dynamic environments. The conventional dynamic RBAC models do not support the personalized environment of each user and have a high search time overhead on activating roles.

In this paper, we proposed PE-RBAC model for dynamic users in the ubiquitous environment. PE-RBAC uses a new constraint, *environment*, which restricts the role activation. As adding the user-to-environment assignment relation to RBAC model, the environment constraint can be personalized to each user. Use of both role-to-environment structure and disjoint property of *environment* can reduce overhead of the active role search time. Through the search depth analysis, we showed that PE-RBAC has smaller search time overhead than the conventional dynamic RBAC models. Moreover, power and

memory consumption is limited because what a user device has to do is only sending an event when the user's environment changes.

For the future work, we need to consider how to scale the role-to-environment structure when a new *environment* or role is added, and how to manage a mapping information between physical and logical environment.

References

[1] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli "Proposed NIST Standard for Role-Based Access Control", ACM Transactions on Information and System Security, Vol. 4, Issue 3, Pages 224-274, 2001.

[2] Ravi Sandhu and Edward J. Coyne, "Role-Based Access Control Models", IEEE Computer, Vol. 29, Issue 2, Pages 38-47, 1996.

[3] David F. Ferraiolo, Janet A. Cugini, and D. Richard Kuhn, "Role-Based Access Control (RBAC) : Features and Motivations", Proc. of Computer Security Applications Conference, 1995.

[4] Ravi Sandhu, "Role-Based Access Control", In Advances in Computers, Eds. Academic, Vol. 46, Pages 237-286, 1997.

[5] Ravi Sandhu, "Role Activation Hierarchies", Proc. of the 3rd ACM workshop on Role-based access control, Pages 33-40, 1998.

[6] E. Bertino, B. Catania, M.L. Damiani, and P. Perlasca, "GEO-RBAC: A Spatially Aware RBAC", Proc. of the 10th ACM symposium on access control models, 2005.

[7] Wataru Yamazaki, Hironori Hiraishi, and Fumio Mizoguchi, "Design an Agent Based RBAC System for Dynamic Security Policy", Proc. of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004.

[8] F. Hansen and V. Oleshchuk, "Spatial Role-based Access Control Model for Wireless Networks", Proc. of the IEEE Vehicular Technology Conference, 2003.

[9] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari, "TRBAC: A Temporal Role-Based Access Control Model", ACM Transactions on Information and System Security, Vol. 4, Issue 3, Pages 191-223, 2001.

[10] J. Joshi, E. Bertino, U. Latif and A.Ghafoor, "A Generalized Temporal Role-Based Access Control Model", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, Issue 1, Pages 4-23, 2005.

[11] Guangsen Zhang and Manish Parashar, "Dynamic Context-aware Access Control for Grid Applications", Proc. of the 4th International Workshop on Grid Computing, 2003.

[12] Guangsen Zhang and Manish Parashar, "Context-aware Dynamic Access Control for Pervasive Applications", Proceeding of the Communication Networks and Distributed Systems Modeling and Simulation Conference, 2004.

[13] Gail-Joon Ahn and Ravi Sandhu, "Role-Based Authorization Constraints Specification", ACM Transactions on Information and System Security, Vol. 3, Issue 4, Pages 207-226, 2000.

Biography

Yuna Kim

Feb., 2001 : B.S. degree in Dept. of CSE, POSTECH
Feb., 2003 : M.S. degree in Dept. of CSE, POSTECH
2004 ~ : Ph.D. candidate in Dept. of CSE, POSTECH

Research Area : service-oriented computing, information & system security
Email : existion@postech.ac.kr

Il Shik Shin

Feb., 2004 : B.S. degree in Dept. of Computer Engineering, Yonsei University
Feb., 2006 : M.S. degree in Dept. of CSE, POSTECH
2006 ~ : Assistant Research Engineer, Network Technology Laboratory, KT
Research Area : network solution, information security, ubiquitous computing
Email : lostmyth@kt.co.kr

Sung Je Hong

1973 : B.S. degree in Dept. of EE, Seoul National University
1979 : M.S. degree in Dept. of CS, Iowa State University
1983 : Ph. D. degree in University of Illinois at Urbana-Champaign
1989 ~ : Professor, Dept. of CSE, POSTECH
Research Area : parallel processing, VLSI testing, fault tolerance
Email : sjhong@postech.ac.kr

Jong Kim

1981 : B.S. degree in Dept. of EE, Hanyang University
1983 : M.S. degree in Dept. of CS, KAST
1991 : Ph. D. degree in Pennsylvania State University
1992 : Professor, Dept. of CSE, POSTECH
Research Area: dependable computing, information & system security, embedded computing
Email : jkim@postech.ac.kr