Adaptive Security Management Model in the Cloud Computing Environment

Youngmin Jung*, Mokdong Chung*

** Department of Computer Engineering, Pukyong National University, 599-1, Daeyeon 3-Dong, Nam-Gu, Busan, Korea*

jym1376@gmail.com, mdchung@pknu.ac.kr

***Abstract*— This paper suggests an adaptive access algorithm to decide the access control to the resources using an improved RBAC technique to solve more complex and difficult problems in the cloud computing environment. And the proposed model determines dynamically security level and access control for the common resources. Therefore, it is supposed to provide appropriate security services according to the dynamic changes of the common resources.**

***Keywords*— Security; Cloud Computing; Access Control; RBAC; DAA protocol**

I. INTRODUCTION

Cloud computing has become a big issue as the IT global companies joined it such as Amazon, Microsoft, Google, IBM, and so on. This technique combines the several computing resources which are in the different places to provide cheaper and easier skills for users with the help of virtualization.

It should provide dynamically the suitable, real-time security services according to the users' demands and environments.

This paper suggests an adaptive access algorithm to decide the access control to the resources based on the contextual information of the environments such as time, location, and security information. And we also suggest an adaptive security management model using an improved RBAC technique to solve more complex and difficult problems in the cloud computing environment.

II. RELATED WORK

A. Cloud Computing

Figure 1 shows the architecture of Cloud computing [1], which provides dynamically scalable and virtualised resources as a service over the Internet on a utility basis. Cloud computing is introduced to deal with the existing computing problems such as limited data capacities, complicated business processes and the scales in the enterprises, the increasing electricity power, and so on. Cloud computing is able to fix these problems by combining the advantage of the main frame computers and that of the distributed computer systems. Cloud computing can extend computing services as the software techniques advance such as network storage technique, virtualized technique, low cost server constructing technique, platform hosting, clustering, multi-tenant architecture, SOA and so on over the Internet.

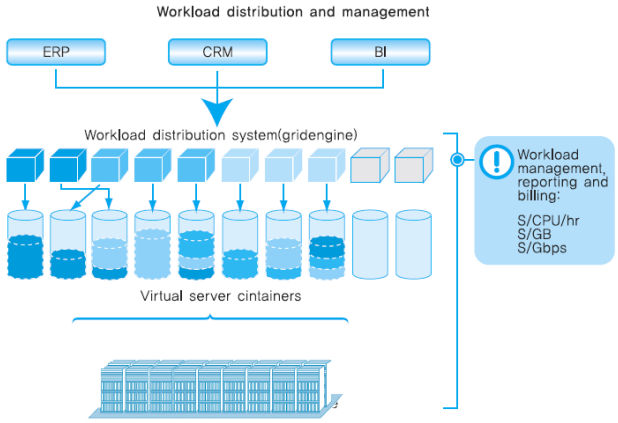

Figure 1. Virtualization architecture for cloud computing

B. Context-Aware Computing

In Dey's definition [2], context may include physical parameters (type of network, physical location, temperature, etc) and human factors (user's preferences, social environment, user's task, etc), and is primarily used to customize a system behavior according to the situation of use and/or users' preferences. Context-aware computing is a mobile computing paradigm in which applications can discover and take advantage of contextual information (such as user location, time of day, nearby people and devices, and user activity) [3]. Thus this paradigm may provide the user with the suitable service which could be appropriate to the user by combining contextual information and the user input.

C. RBAC(Role-Based Access Control)

Figure 2 shows conceptual model of RBAC. RBAC model is defined in terms of three model components—Core RBAC, Hierarchical RBAC and Constraint RBAC Core RBAC

includes sets of six basic data elements called users (U), roles (R), objects (O), operations (Ops), permissions (P) and sessions (Sessions). The basic concept of RBAC is that users are assigned to roles rather than users. The users acquire permissions by acting members of roles.

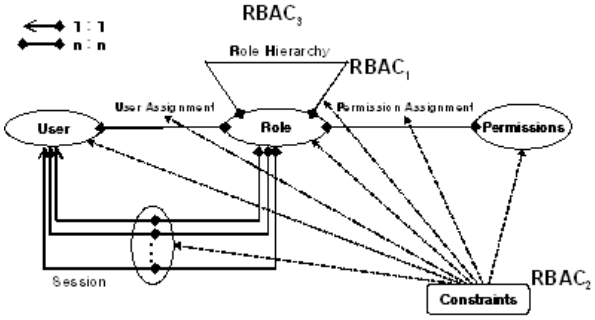

Figure 2. Conceptual model of RBAC

Hierarchical RBAC describes hierarchical relation between roles. It goes beyond simple user and permission role assignment by introducing the concept of a role for authorized users and authorized permissions.

Constrained RBAC is used to set constraint condition in role assignment and activate roles in a session [4]. It is made up of two models: SSD (static separation of duties) and DSD (dynamic separation of duties). SSD defines mutually disjoint user assignments with respect to sets of roles. DSD limits the permissions that are available for a user, its requirements limit the availability of the permissions by placing constraints on the roles that can be activated within or across a user's sessions. Both SSD and DSD is the guarantee for implementation of least privilege principle.

Role-Based Access Control (RBAC), introduced by Ferraiolo and Kuhn, has become the predominant model for advanced access control because it reduces the complexity and cost of security administration in large networked applications [5]. With RBAC, system administrators create roles according to the job functions performed in an organization, grant permissions to those roles, and then assign users to the roles on the basis of their specific job responsibilities and qualifications [6].

If an RBAC framework is established for an organization, the principal administrative actions are the granting and revoking of users. This is in contrast to the more conventional process of attempting to administer lower-level access control. This simplifies the administration and management of privileges. Roles can be updated without updating the privileges for every user on an individual basis. In our model, however, we extend the concept of role to the including of resources as well as users.

D. Improved RBAC

Figure 3, shows Role Switching, where if the requesting cost of service is less than budget limit then role is not changed, but if it is greater than or equal to the budget then role is changed to role for higher version performance.

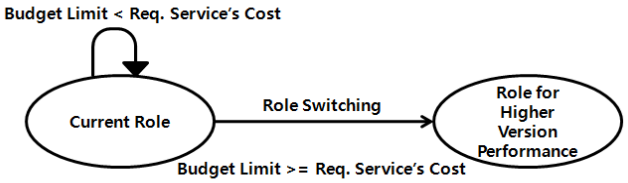

Figure 3. Role switching

If we are able to use the role switching as shown in Fig. 3, we can reduce the users' waste of spending and unnecessary waste of resources. We try to provide a dynamic RBAC in the Cloud computing environment to overcome these problems such as the Context Sensitive Access Control [7] or FCM Algorithms [8]. In addition it can decrease unnecessary purchases of computer resources for service upgrading.

E. MAUT(Muliti-Attribute Utility Theory) and Simple Heuristics

Multi-Attribute Utility Theory is a systematic method that identifies and analyzes multiple variables in order to provide a common basis for arriving at a decision. As a decision making tool to predict security levels depending on the security context (network state, the resource's and user's environments, etc), MAUT suggests how a decision maker should think systematically about identifying and structuring objectives, about vexing value tradeoffs, and about balancing various risks. The decision maker assigns utility values to consequences associated with the paths through the decision tree. This measurement not only reflects the decision maker's ordinal rankings for different consequences, but also indicates him relative preferences for lotteries over these consequences [9].

According to MAUT, the overall evaluation v(x) of an object x is defined as a weighted addition of its evaluation with respect to its relevant value dimensions [10]. The common denominator of all these dimensions is the utility for the evaluator [11]. The utility quantifies the personal degree of satisfaction of an outcome.

The MAUT algorithm allows us to maximize the expected utility in order to become the appropriate criterion for the decision maker's optimal action.

The Center for Adaptive Behavior and Cognition is an interdisciplinary research group founded in 1995 to study the psychology of bounded rationality and how good decisions can be made in an uncertain world. This group studies Simple Heuristics[12]. One of them is Take-The-Best which tries cues in order, searching for a cue that discriminates between the two objects. It serves as the basis for an inference, and all other cues are ignored.

III. ADAPTIVE SECURITY MANAGEMENT MODEL BASED ON RBAC

Figure 4 shows overall architecture of the proposed model. When a user is trying to access to a protected resource, the service provider collects various contextual information from environment and user. The service provider consists of two

modules: the service module which provides users with various types of services such as e-commerce, and DRM (Digital Rights Management) service, etc., and the security module which offers security function of the services.

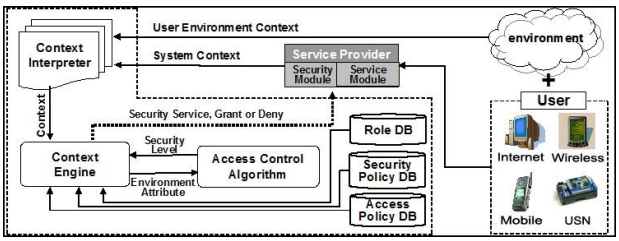

Figure 4. Architecture of the context-based adaptive security management

The context interpreter converts collected contexts to quantitative values. The context engine evaluates security level by using these values and the access control algorithm. According to the security level, role, and access policy, the context engine determines the appropriate security services. In our model, the security service includes granting or denying access. And then, the result of this security service can be delivered to the user who can perform his or her action according to this result.

A. Access Control Management Model Using RBAC

In this paper, we propose a modified RBAC (Role Based Access Control) model which provides an appropriate policy with a specific resource instead of specific role in the traditional RBAC model.

Depending on various characteristics such as priority, worth of the resource, specific policies including security and access policy, may become components of each particular role.

Because access right to each resource can be assigned to the users in terms of the various policies without any change of policy, the appropriate policies can be manipulated due to the dynamic changes of resources.

Figure 5 shows an improved access control management model with RBAC which is proposed in this paper.

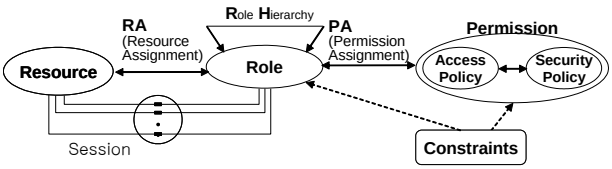

Figure 5. An access control model using RBAC

This proposed access control model consists of Resource, Role, and Permission.

- Resource and Roles: Resource provides service and is protected. A role is classified according to the security requirement in the system. The concept of role is extended to the including of resources as well as users.
- Permission and Constraints: Permission is an approval of a particular mode of access to one or more objects in the system. Constraints are restricting conditions which might be applied to the policies.
- Session: Users establish sessions during which they may activate a set of the roles they belong to. Each session maps one role to possibly many resources.
- Resource Assignment (RA) and Permission Assignment (PA): A role can belong to many resources, and a role can have many permissions.

In the proposed model, resource becomes members of specification role according to the characteristic of resource. Therefore, this model has the advantage that simplifies security policy management and makes security policy flexible as well. Figure 5 shows the access control model using RBAC.

B. Security Policy and Access Policy

Figure 6 shows an example of the security policy and the access policy which may be applied to the protected resources. The security policy that is applied to resource has two components: role and prerequisite, where there are three prerequisites such as constraints, utility function, and user's preference. The access policies have also two components: role and action, where we have two actions; read and download.

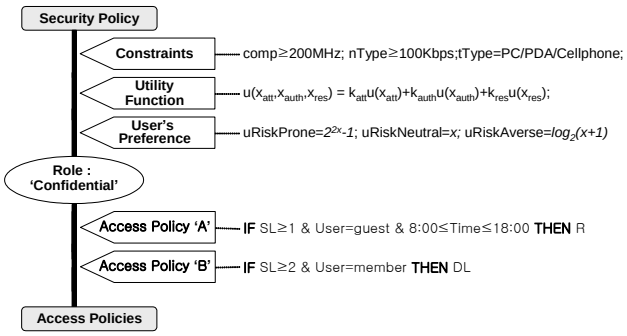

Figure 6. An example of security policy and access policy

In the constraints node of Figure 6, we need to have a terminal equipped with better than 200 MHz CPU and bandwidth over 100 Kbps to access to the protected resource. Also we can use PC, PDA, or Cellular phone. Comp is computing power for message encryption/decryption, nType is network type, and tType is terminal type, respectively.

In the utility function, x_{att} is the strength of the cipher, x_{auth} is the authentication method, and x_{res} is the level of protection of the resource to which the user is trying to access. In the role, the role is connected to the security policy and access policy at the same time, and the role value is provided to the role. The dynamic change of the security policy leads to that of access policy. Therefore, the characteristic of the pro-posed model is adaptively flexible.

C. Context Interpreter and Context Engine

The context interpreter gathers environmental contexts from the service provider and user, and converts these contexts to the range scaled from 0 through 1. And then, the context interpreter

delivers the converted value to the context engine in order to evaluate security level by using adaptive access control algorithm. Table 1 demonstrates a typical conversion table for environmental contexts.

Table 1. Conversion Table for Environmental Contexts

Variable	Value			
	0.2	***0.5***	***0.8***	***1.0***
x_{att}	$\geq 10^{0.5}$	$\geq 10^{3}$	$\geq 10^{7}$	$\geq 10^{11}$
x_{auth}	Password only	Certificate	Biometric	Hybrid
x_{res}	No	Low	Medium	High

The context engine evaluates security level by using contexts, various databases, and the adaptive access control algorithm. According to the determined security level, role, and access policy, the context engine gives the appropriate security services.

D. Adaptive Access Control Algorithm

The overall algorithms for determining adaptive security level and access availability for the protected resource are as followed Table 2.

Table 2. Access Control Algorithm

AccessControl(AccessProblem**)**
// AccessProblem: Grant or deny access for the protected resources according to Access Policy(AP);
// Search Role that related Resource
search Role **by using** context about User's Request
// Determine Security Level using Security Policy(SP)
SL = **SecurityLevel(**securityProblem**) ;**
// Grant or deny access by calculating Role, AP, and SP.
calculate Role, AP, and SP **by using** Constraints, Role, AP, and SP;
if true return grant;
else return deny;

The overall algorithms for determining adaptive security level are as followed Table 3, Table 4, Table 5 and Table 6.

Table 3. Security Level Algorithm

SecurityLevel(*securityProblem***)**
// Determining security level using Security Policy (constraint, utility function, user preference)
// Utilization of domain independent properties
calculate *SL* **by** *I* **end;**
if *SL = 0* **then return;** *SL // No* security system
// Utilization of domain dependent properties
// select a strategy between MAUT and S. Heuristics
if MAUT **then** *SL* = MAUT(*X*);
if Simple Heuristics **then** SL = TakeTheBest(*X*);
return *SL*;
end;

Table 4. MAUT Algorithm

MAUT(X**)**
// Determine total utility function by the interaction
// with the user according to MAUT $u(x_1,x_2,,x_n) = k_1u_1(x_1)+k_2u_2(x_2)+ +k_nu_n(x_n)$
// k_i is a set of scaling constants
// x_i is a domain dependent variable, where $u_i(x^o_i)=0$,
// $u_i(x^*_i)=1$, and k_i is positive scaling const. for all i
ask the user's preference and **decide** k_i;
for i = 1 **to** n
do $u_i(x_i)$ = GetUtilFunction(x_i);
end;
return $u(x_1,x_2,,x_n)$;
end;

Table 5. Get Utility Function Algorithm

GetUtilFunction(x_i**)**
// Determine utility function due to users' preferences
// x_i is one of domain dependent variables
uRiskProne : user is risk prone for x_i // convex
uRiskNeutral : user is risk neutral for x_i // linear
uRiskAverse : user is risk averse for x_i //concave
x : arbitrary chosen from x_i
h : arbitrary chosen amount
$<x+h, x-h>$: lottery from x+h to x-h
// where the lottery (x^*, p, x^o) yields a p chance at x^*
// and a $(1-p)$ chance at xo
ask user to prefer $<x+h, x-h>$ or x; // interaction
if user prefer $<x+h, x-h>$ **then**
return uRiskProne; // e.g. $u = b(2^{cx}-1)$
else if user prefer x **then**
return uRiskAverse; // e.g. $u = blog_2(x+1)$
else
return uRiskNeutral;
end; // e.g. $u = b$

Table 6. Take the Best Algorithm

TakeTheBest($u(x_1,x_2,,x_n)$**)**
// Take the best, ignore the rest $u(x_1,x_2,,x_n)$: user's basic preferences
// if the most important preference is xi, then only *xi*
// is considered to calculate *SL*
// The other properties except *xi* are ignored
$u(x_1,x_2,,x_n)$ is **calculated by** only considering x_i;
SL is **calculated by** the value of $u(x_1,x_2,..,x_n)$;
return *SL;*
end;

E. Service Provider

Service providers offer special services for users to apply access policy and security policy depending on each environmental factor such as type and importance of service, current status of the system, and so on.

And the proposed model offers contexts of the users and service providers to the context interpreter in the access control management model to provide effective security policy and security service for the various users.

F. Security Service

This model determines the adaptive security level to meet the dynamic changes of environmental attributes in the ubiquitous environments. Based on this security level, this

model adaptively adjusts the values of the environmental contexts of a security system such as algorithm type, key size, authentication method, and/or protocol type as shown in (1).

$$U = \sum_{i=1}^{n} k_i u_i(x_i),\ (0 \le U \le 1)$$
$$SL = \lceil U * 10 \rceil / 2,\ (SL = 0, 1, \ldots, 5)$$
$$P_{SL} = \{p_0, p_1, \ldots, p_5\}$$
$$p_i = \{S_j, A_l, R_m\} \quad (1)$$

U is the total utility value, u_i is a utility value of environmental attributes in the heterogeneous networks, and k_i is a scaling constant of the environ-mental attributes. SL represents security level scaled from 0 through 5, where value 0 means we cannot utilize the security system. The larger the number is, the stronger the strength of security is. Table 1, Table 2 shows protocol types, and authentication methods, respectively, Table 3 shows several algorithm types.

IV. Application under the Cloud Computing Environment

In this paper, we try to test the node authentication and communication with DAA(Direct Anonymous Attestation) protocol.[13]

The test bed uses Cloud Computing resources in the DAA protocol environment which includes the experimental Trusted Platform Module (TPM) module.

For the availability test of the proposed model which is shown in Figure 7, we implement the access control based on RBAC adaptive security model. The proposed model handles the access control of the resources and determines security level throughout the service provider. Security level and access control are also chosen dynamically by the various environmental factors.

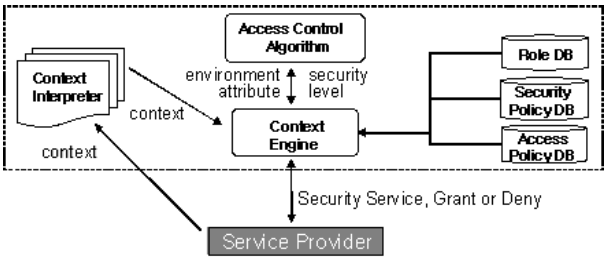

Figure 7. Adaptive security management model

Suggested model is implemented by using the Java5.0 on IBM compatible PC with Intel Pentium-IV 3.0 GHz CPU for checking learning and performance of the proposed model.

A. Using RBAC in the Cloud Computing Environment

Figure 8 shows Cloud Computing Access Control model, which has rule-based roles to issue bills for the variable usage and includes specific roles for users' preferences. Role switching solves these problems according to the context information in Figure 3 in the section 2.

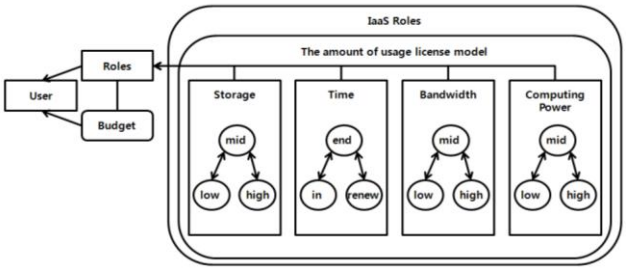

Figure 8. Cloud computing access control

B. DAA(Direct Anonymous Attestation) Protocol

Figure 9 structure of a device with DAA module and Figure 10 show and Trusted Platform Module(TPM). DAA is one of group signature scheme based on Zero Knowledge Proof [14] which is designed by TCG. It provides a remote authentication of TPM hardware while protects privacy of user in Platform [15]. Features of DAA are as follows.

- DAA authenticates without TTP (Trusted Third Party).
- DAA provides anonymity.
- DAA has ability to find rogue TPM.
- DAA is secure in random oracle model because it is based on Strong RSA and Decisional Diffie-Hellman Assumption [16].

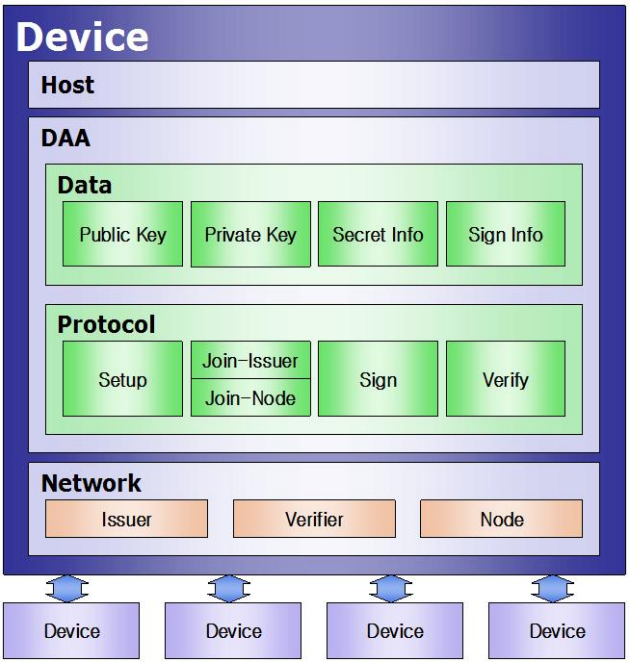

Figure 9. Structure of a device with DAA module

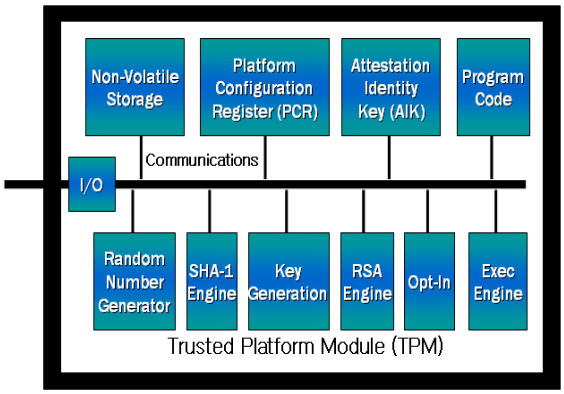

Figure 10. Trusted platform module(TPM)

DAA protocol handles a TPM user who wants to be verified, an Issuer for an issue certificate, and a Verifier for verification of a TPM user. DAA protocol has setup, join, sign, and verify procedures. These procedures are as follows.

1) Setup: It makes public key and private key of issuer using Fiat-Shamir Heuristic [17].

2) Join: TPM sends information of N1f to DAA Issuer and proves that it has private information f. Then TPM issues private information (f0, f1, v) for creating certificate by DAA Issuer.

3) Sign: TPM signs message using issued certificate by Join protocol and the received N2f from Verifier.

4) Verify: TPM verifies the signature through DAA verifier.

V. CONCLUSIONS

In this paper, we implement DAA protocol for the security of the individual devices under the cloud computing environment. It also defines TPM module using JAVA. And we have tested authentication and communication of each node through DAA protocol.

And the proposed model determines dynamically security level and access control for the common resources. Therefore, it is supposed to provide appropriate security services according to the dynamic changes of the common resources.

We can expect our proposed model to solve the security problems in the static environment using MAUT and simple heuristics. The static security service system cannot deal with dynamic changes of the multiple environmental variables. In this paper, however, adaptive security management in the proposed model based on RBAC may be expected to solve these problems effectively.

Also the proposed model solved problems of requiring unreasonable system resources and longer waiting time. And it protects resources safely from mal-intentional users.

REFERENCES

[1] James Staten, “Is Cloud Computing Ready for the Enterprise?,” Forrester Research, March 2008.(http://www.forrester.com/rb/research)

[2] A. K. Dey, "Providing Architectural Support for Building Context-Aware Applications," Ph. D. Dissertation, Georgia Institute of Technology, 2000.

[3] Guanling Chen, "A survey of context-aware mobile computing research," Dartmouth Univ.TR2000-38

[4] Kong Guangqian, Li Jianshi. “Research on RBAC-based separation of duty constraints,” Journal of Information and Computing Science, Vol.2, pp. 235-240, 2007.

[5] NIST, (http://csrc.nist.gov/rbac/)

[6] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E.Youman., "Role-based access control models," IEEE Computer, Vol. 29, No. 2, pp. 38-47, February 1996.

[7] R.J. Hulsebosch, "Context Sensitive Access Control,” Symposium on Access Control Models and Technologies, pp. 111-119, June 2005.

[8] Seokhwan Yang, Mokdong Chung, “Context-Aware Security Service for Healthcare System in RFID/USN Environments using Fuzzy C-Means Algorithm,” 3rd International Conference on Ubiquitous Information Technologies & Applications, Vietnam, pp. 448-452, December 2008.

[9] R.L.Keeney and H.Raiffa, “Decisions with Multiple Objectives: Preferences and Value Tradeoffs,” John Wiley & Sons, New York, NY, 1976.

[10] D. Winterfeld, von and W. Edwards, “Decision Analysis and Behavioral Research,” Cambridge, England: Cambridge University Press.

[11] “Rules for Using Multi-Attribute Utility Theory for Estimating a User’s Interests,” Proceedings of the ninth GI-Workshop. ABIS-Adaptivität und Benutzermodellierung in interaktiven softwaresystemen, Dortmund, Germany, 2001.

[12] L. Martignon and U. Hoffrage, “Why Does One-Reason Decision Making Work? In Simple Heuristics That Make Us Smart,” Oxford University Press, New York, pp. 119-140, 1999.

[13] Brickell, E., Camenisch, J., Chen, L., "Direct anonymous attestation," CCS’04 11th ACM conference on Computer and communications security, New York, United States of America, pp. 132–145. ACM Press, New York, 2004.

[14] Quisquater Jean-Jacques, "How to Explain Zero-Knowledge Protocols to Your Children,” Advances in Cryptology — CRYPTO'89 Proceedings, Lecture Notes In Computer Science,Vol.435, pp. 628-631, 1989.

[15] E. Brickell, J. Camenisch, and L. Chen, “Direct anonymous attestation,” In Proceedings of 11th ACM Conference on Computer and Communications Security, ACM Press, 2004 Practical Solutions to Identification and Signature Problems, ACPC 86, LNCS, 1987.

[16] M.J.Convington, “Generalized Role-Based Access Control for Securing Future Applications,” Proc of 23rd National Information Systems Security Conference (NISSC), Baltimore, pp. 115-125, 2000.

[17] M.J. Moyer, M.Ahamad, “Generalized Role-Based Access Control,” Proc of IEEE Int’l Conf. on Distributed Computing Systems (ICDSC2001), Mesa, pp. 391-398, 2001.