IMPROVED & EXTENDED-RBAC (JV-RBAC) MODEL WITH X.509 AUTHENTICATION

Ms. Jyoti Joshi, Dr. Kunwar Singh Vaisla,

Department of Computer Science & Engineering, BT Kumaon Institute of Technology, Dwarahat – 263653, District – Almora (Uttarakhand), India
joshi.sonjiya.jyoti@gmail.com[1], vaislaks@rediffmail.com[2]

ABSTRACT

Role-based access control models have attracted appreciable research interest in past time due to their providing some flexibility to security management and ability to model organizational structure and their capability to reduce administrative expenses. In this paper, we explains the drawbacks of RBAC96 model in the aspect of the authorization, access rules and fine-grain access controls in the practical application and established a extended-RBAC model, named as JV-RBAC (Joshi-Vaisla RBAC), which integrates the authorization of users and roles, authentication is implemented by X.509 and providing higher security by introducing access rules and audit function.

KEYWORDS

Extended-RBAC, fine-grained access control, audit, access rules, X.509, PKI, PMI;

1. INTRODUCTION

Access control is a core concept in security. This can be done through authentication, authorization, and access control. These three mechanisms are distinctly different but usually generally it can effectively manage all requests for access systems and it can protect the authorized users to access information systems from unauthorized access. Role based access control (RBAC) emerged rapidly in the 1990s as a proven technology for managing and enforcing security in large-scale enterprise wide systems. It can provide more flexibility to security management over the traditional approach of using user and group identifiers. Role-based access control system is divided into the user functions and positions consistent with their roles, according to the role given the authority of the appropriate action in order to reduce the authorization management of complexity, reduce administrative overhead and provide a better environment of complex security policies implementation. Another important technology that can be used for access control is Privilege Management Infrastructure (PMI). The main function of PMI is providing a strong authorization after the authentication has taken place.

We were motivated by the need of using PKI(Public Key Infrastructure), PMI and RBAC concepts to construct an authorization mechanism. RBAC96 model is an extremely important access control model, which is the center of attraction of research and application. X.509 supports RBAC by defining role specifications attribute certificate (ACs) that hold the permissions granted

DOI : 10.5121/cseij.2012.2305

to each role, and role assignment ACS that assign various roles to the users[3]. A public key infrastructure (PKI) aims to authenticate communicating parties, but authentication alone is not enough. Thus, we need an authorization mechanism. A privilege management infrastructure (PMI), which is similar to a PKI, enables authorization after authentication has occurred[3].

This paper analyzes the extended-RBAC model present by YANG Liu and et al. For its shortages, we present a modified and extended-RBAC model, which inherits authorized users and the role of mixed characteristics of extended-RBAC, presents a new expansion program with X.509 certificate that can solve the authorization management in the complex multi-application systems. Finally this making RBAC authorization and authentication more flexible, more secure and more versatile.

2. EXISTING MODELS

2.1. RBAC96 model

The central notion of RBAC: permissions are associated with roles and users are assigned to appropriate roles. This greatly simplifies management of permissions. Roles are created for the various job functions in an organization and users are assigned roles based on their responsibilities and qualifications. Users can be easily reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, and permissions can be revoked from roles as needed. The role's concept was introduced between the users and authority, and the roles and users will be linked, through the role authorization to control the access of system resources.

RBAC96 model is a typical representative of the RBAC model, as shown in Figure 1. It splits traditional RBAC model based on the different needs of nested into four models and were given formal definition has greatly improved system flexibility and availability. In this model, there are three sets of entities called users (U), roles(R), and permissions (P) and also shows a collection of sessions(S). A user in this model is a human being. The concept of a user can be generalized to include intelligent agents such as robots or even network of computers. A role is a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role.

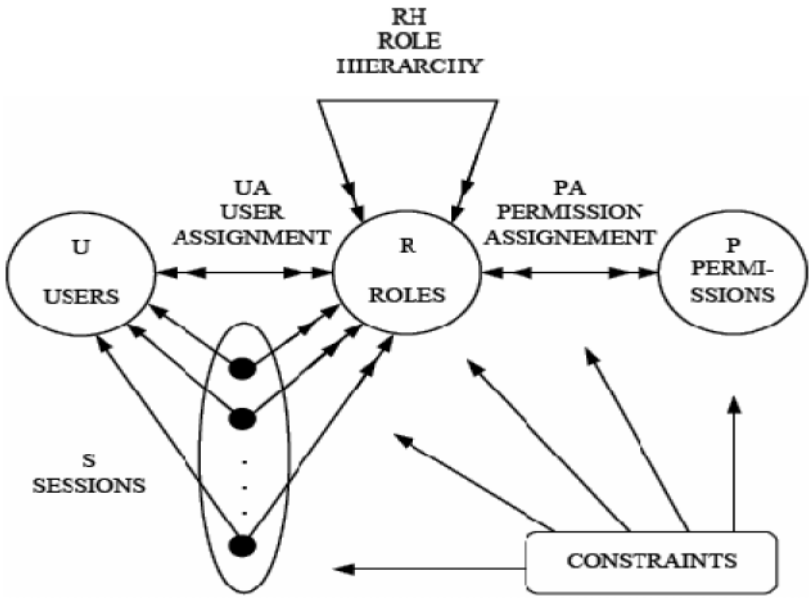

Figure1. RBAC96 model

Permission is an approval of a particular mode of access to one or more objects in the system. The terms authorization, access right, and privilege are also used in the literature to denote permission. The nature of permission depends greatly on the implementation details of a system and the kind of system that it is. Each session is a mapping of one user to possibly many roles, i.e., a user establishes a session during which the user activates some subset of roles. The double-headed arrow from session to R in figure 1 indicates that multiple roles are simultaneously activated. The permissions available to the user are the union of permissions from all roles activated in that session. Each session is associated with a single user, as indicated by the single-headed arrow from the session to U. This association remains constant for the life of a session. Role hierarchies are a natural means for structuring roles to reflect an organization's line of authority and responsibility. Constraints are a powerful mechanism for laying out higher level organizational policy. With respect to RBAC96 model constraints can apply to the UA and PA relations and the user and roles functions for various sessions. Constraints are predicates which, applied to these relations and functions, return a value of "valid" or "not valid".

One of the benefits of RBAC: users with a direct link with access when the organization needs more staff or a change in the functions of the user, there are lots of authorization work. In the RBAC model, the user's authorization is changed the role's authorization, then link the users with a specific role. The major management after RBAC system was to authorize or to cancel user's role. Another benefit of RBAC: system administrators control the access in a more conceptual level which is similar to business management. This authorization is very close to the regular management of the organization rules. Regardless of the RBAC has been widely used, but the traditional RBAC model is still insufficient in the following areas.

1) To make the role as the only authorization way, as the lack of flexibility.
2) The model despite introduced static and dynamic constraints, there are no rules of the restrictions, and the restrictions are often the reality of the management system needed

but not through constraints defined, such as to allow for the access with the role of the conditions. Access rules of the restrictions can be made up for constraints deficiencies [3,4] .

3) The model doesn't provide more security from unauthorized access for e-Governance applications.
4) The model still is coarse-grained access control.
5) Fine-grained access control [5, 6, 7] is a better solution, which is more precise.

2.2. Extended-RBAC model

E-RBAC model, which was presented for RBAC96 model's role authorization shortage, based on RBAC96, and presented the expansion of the role of mixed-type role-based access control model [9], shown in Figure 2.

In addition to the RBAC model authorization, In E-RBAC users can also direct for authorization. For example, when the authority is authorized after a specific request to access the user's system resources, the system at the same time to judge whether the users own role or whether the user has authority access to the module's functions, as long as one was given the authority between the users their role, to allow access, this hybrid approach is not only authorized by an effective solution to the definition of the roles, responsibilities of users, operation functions, such as changes in the dynamics of the system brought about by the problem but also to enhance the user authorization.

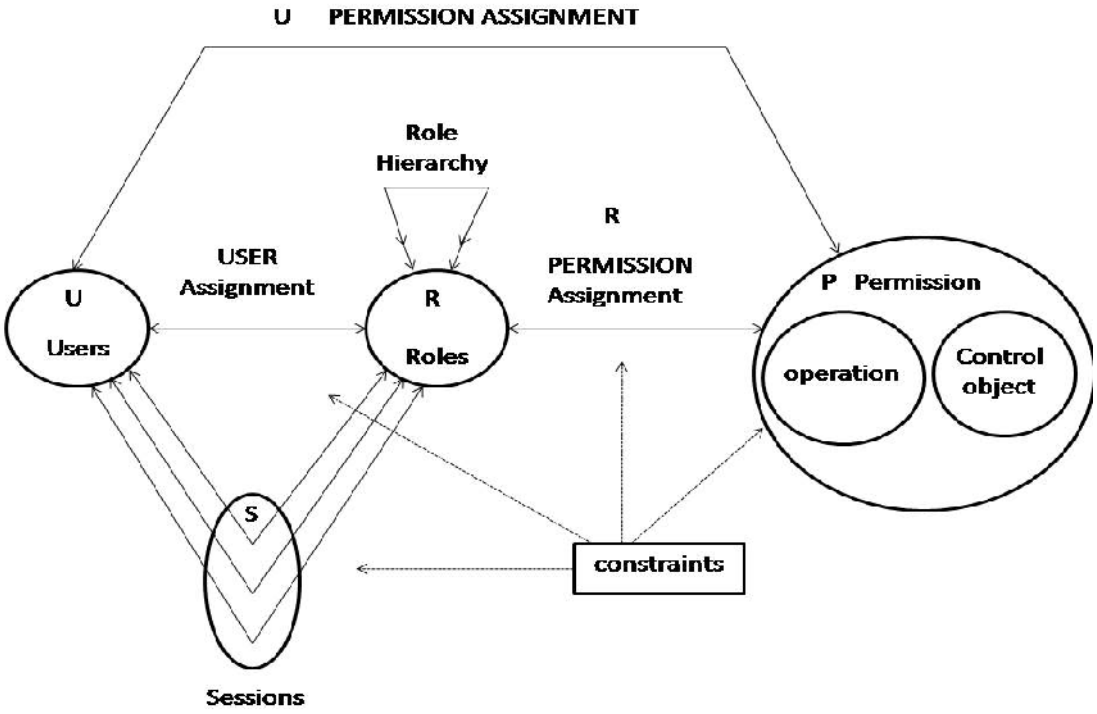

Figure2. E-RBAC model

3. THE IMPROVED AND EXTENDED-RBAC (JV-RBAC) MODEL

The E-RBAC only gives the solution to the user's authorization problem, but for the rules of access, access control granularity, audit there are no corresponding procedure. We presented an improved RBAC model for E-RBAC model shortage whose structure shown in figure 3.
In this model, we presented such aspects as access rules, access control granularity, modules, time constraints, as well as auditing strategy.

The introduction of modules, the objects can be fine-grained, multi-dimensional (at the time, space, timing, etc.) access control. The modules and roles must combine to assign permission, which to combine with user authorization and roles authorization to make a more flexible authorization. In accordance with the special requirements use the corresponding authorization. To refine the role in each of the modules and systems departments, that each role can only operate specific modules.

The introduction of the time constraint, users can lock their own account number while they aren't on line. Even if the illegal use of such accounts, lock-in period of time cannot be used, as to improve security.

The introduction of the audit not only can reproduce the original process and issues, but also to track and record operation of the administrators and illegal users in the log. It is necessary to track responsibility and data recovery.

The introduction of access rules, to make up for shortage of the traditional RBAC static constraint and dynamic constraint. It provides more fine-grained access control for System, but also can reduce the workload of the traditional role of the RBAC model in the distribution of the authorization.

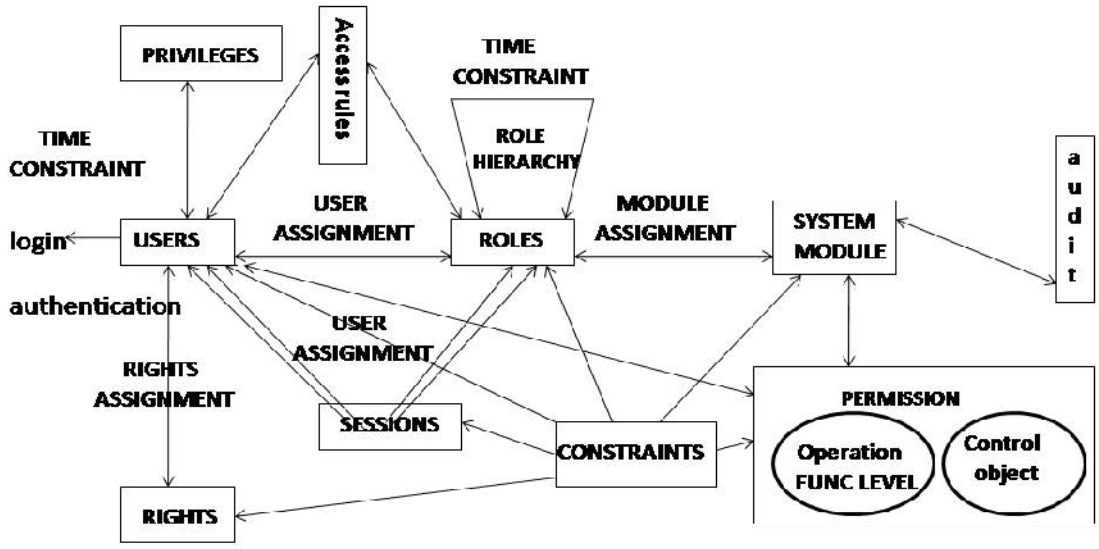

Figure3. JV-RBAC Model

The introduction of authentication, by using X.509 certificate authentication user can identify itself. If identity authentication success. Then system to load the group and individual signature

and role of the time information after that placed in session. Otherwise, user withdraws from the system.

3.1. JV-RBAC model fine-grained permission expand

Shown in Figure 3, E-RBAC model will feature set of permissions (FP) is divided into two parts, one part for the operating rights set (OPS), another part is the object operation of set (OBS). Operator permission set (OPS) is divided into operating function set (FUNC) and operational level set (LEVEL). Object-operation (OBS) is refined data collection object (DATA).

- Operating function set (FUNC) = (func1, func2, ..., funcn) mainly refers to menu in the information system of function, such as recording information, print scores, personnel management and so on.
- Set-level operation (LEVEL) = (1evell, level2, … ,leveln) mainly refers to operation level in the information system of function, such as the disabled, query, modify, delete and so on.
- Operating permission set (OPS) = (ops1, ops2, ... ,opsn) $=2^{FUNCxLEVEL}$ mainly refers to functionality permission in the information system of function.
- Set data objects (DATA) = (data1, data2, ... ,datan) mainly refers to specific data objects in the information system of function.

Through fine-grained extending E-RBAC model, departments, functions, data, and other dimensions of the model are refined and achieve a multi-level system for administrator(s) of large-scale information systems. It makes the system function and control data more flexible, refines the control of the system to the directory, documents, Web pages of a fine-grained access control, and the management of roles and users become more convenient.

3.2. JV-RBAC the access rules

Definition:

- The role of the time (URT): Users have had the role of all time. URT = (urtl, urt2, ..., urtn).
- The role of the operation (UROP): Users have had the role of the operating record of the number of all. UROP = (uorp1, uorp2, ..., uropn).
- User time period (URTP): Users have been used to operate all of the role of the time. URTP = (urtp1, urtp2, ..., urtpn).

3.3. JV-RBAC - Access rules Algorithm: get_decision ( )

Input:

User, role, rights, privilege, permission

Output:

False or true

Stepl:

Input expression // *it includes user's name, role, rights, privilege, permission*

Step2:

expression = get_expression (expression) // *Analysis to get the name of role and user name, and operation were assigned to the get_role and get_per*

Step3:

is1 = check_URT (user-get_role) > RT

(The operation to be met the needs of the role owned time)?

True: false // *To match the role owned time*

Step4:

is2 = check_UROP (user-get_role) > OPN

(The operation to meet the needs that operation have been enough records)?

true: false // *To match the role owned operation*

Step5:

is3 = workt (iget_ Timenow (user-get_role)) // To match the personality signature of role

Step6:

is4=sign(status)

Step7:

is5=authorized_user (user-get_role, user-get_per, user-get_privilege)

// match user is authorized

Step8:

is6=assign-rights(user-get_per, user-get_privilege, user-get_rights)

//assign rights to the authorized user

```
if (is1||is2|| is3|| is4||is5||is6) = true
        Give (OPS)
else
        Drop (OPS)
```

Access rule is another restriction mechanism to check users status. Whether the user is authorized or not. If the user is authorized then assign rights to the user. If the all procedure is true then give the operating permission set (OPS) to it otherwise drop.

3.4. JV-RBAC model involves the realization of the main database tables

Improved E-RBAC model main exist entities are: user, role, powers. Main tables such as user information table, role of the information sheet table, information permission table, user rights table, log database table. As follows:

Functional role table (functional role ID, functional role name, department ID, functional role father ID, mutually exclusive role ID, essential role ID, basic restrictions number);

Functional role authority table (functional role ID, Menu function ID, operator-level ID, operation module ID, cycle data ID, data object ID);

Department role table (sector role ID, Department role name, the department ID, Department role ID, mutually exclusive role ID, essential role ID, basic restrictions number);

Role authority departments table (Department role ID, allow access to department ID) ;
User table (user ID, user name, department ID, user-level ID);

User role table (user ID, functional role ID, role department ID, own role time URTIME, role working hours UWORKT, user role signature SIGN);

User Rights table (user ID, allow access unit ID, menu features ID, speak-level ID, data object ID) ;

Log table (log ID, user ID, landing time T, user-level ID, whether the abnormal IS, role of ID, menu features ID) users to record all the operation;

Abnormal operating table (menu features ID, department ID, operation time TI, role of ID, menu features ID) to record all abnormal operations; analyze the administrators in order to operate the high rate of abnormal operation, to take appropriate safety measures.

3.5. JV-RBAC model process chart

Details of the process are as follows (shown in figure 4).

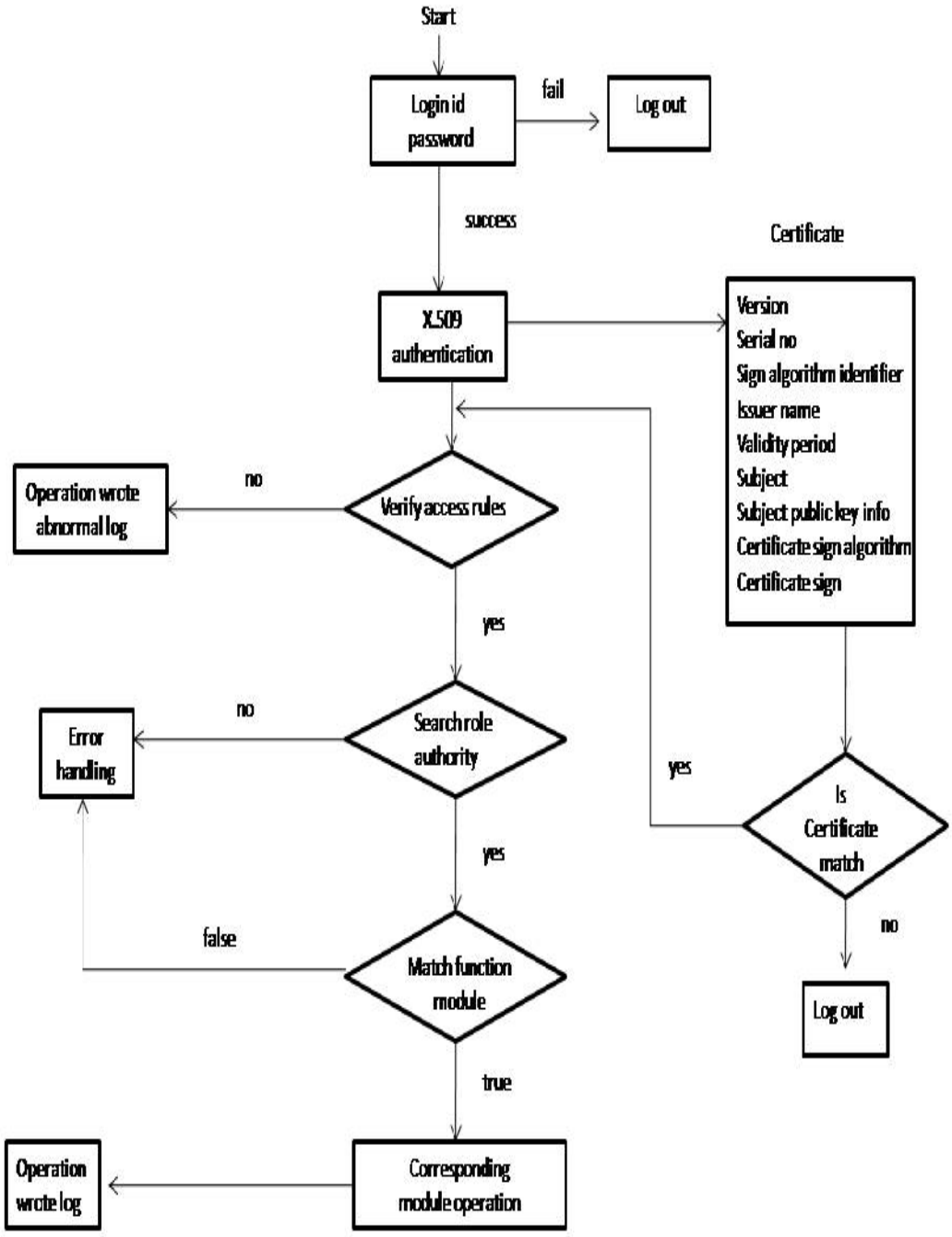

Figure4. JV-RBAC Process Chart

1) **User login:** If login id and password is match with the user then identity authentication success. Then system loads the group and the individual signature and the role of time information after that placed in session. Otherwise, user withdraws from the system.

2) After that further authentication is start with X.509 authentication. If the user's detail match with the certificate then authentication success. Otherwise, user logout from the system.
3) Verify the user whether accord with access rules (role of the time, role of the operation, user time period) of the module, then to call the access rules method get_decision ( ). If users meet the rules, the system check the corresponding table whether there is a license to operate, and then to verify whether there is a permission to operate in the functional competence of the functional authority table. Otherwise, return error handling.
4) To mismatch the rules, its operation writes into the abnormal operation log.
5) Finally, the day-to-day operation wrote into the log.
6) Audit administrator audits abnormal and day-to-day operation of the audit log.

4. CONCLUSION

Improved and extended-RBAC model not only to integrate the authorization of users and roles but also enhance its flexibility and maintainability of user authentication by using X.509 certificate. At the same time, time and authorization constraints as well as access rules, making the system security much higher. Improved RBAC through the expansion of fine-grained model, departments, functions, and other dimensions of the model were refined to achieve a large-scale information system of fine-grained access control. By using audit function, the system administrator can track the abnormal behavior of illegal users, to enhance further security.

5. REFERENCES

[1] Dr. K. S. Vaisla, J.Joshi, "Modified and Improved Extended –Role-Based Access Control Model", Proc. of National Conference on Business Intelligence and Data Warehousing, by Narosa Publication pp.206-213, March 2012.
[2] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E.Youman, "Role-based access control models", IEEE Computer, vol.29, pp.38-47, February 1996.
[3] D. W. Chadwick, A. Otenko, and E.Ball, "Implementing role based access controls using X.509 attribute certificates", IEEE Internet Computer Society, March-April 2003.
[4] XUE Wei and HUAI Jin-peng, "Research on extension and implementation mechanism for rule-based access control", Journal of Computer Research and Development, vol.40, pp. 1635-1642, November 2003 (in Chinese)
[5] Aneta Ponizewska-Marada, "Role engineering of information system using extended RBAC model", Proc. of 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE05), IEEE Computer Society Press. Jun. 2005, pp.154-159, DOI: 10.1109/WETICE.2005.50
[6] YANG Ya-ping, LI Wei-qin, and LIU Huai-yu, "Research and Implementation of Role-Based Access Control System", Journal of Beijing University of Aeronautics and Astronautics, vol.27, pp. 178- 181, February 2001 (in Chinese).
[7] SI Wei, ZENG Guang-zhou, SHENG Qi, and LI Ying-jun, "Fine Grain Extension and Application of the RBAC Model", Computer Science, vol.33, pp. 87-89, April 2006 (in Chinese).
[8] ZHAI Zheng-de, FENG Deng-guo, and XU Zhen, "Fine-grained controllable delegation authorization model based on trustworthiness", Journal of Software, vol.18, pp.2002-2015, August 2007 (in Chinese).
[9] P. Zhang, W. Yanzhang, "Study on the Improved RBAC Model in E-Government", IEEE, 978-1-4244-5326-9/10, 2010.
[10] S.Farrell, R.Housley, An Internet Attribute Certificate Profile for Authorization, Internet-draft April 2002, http://www.ietf.org/rfc/rfc3281.txt.