Unified Identity Authentication between Heterogeneous Systems Based on LDAP and RBAC

Guowei Wang
School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China
Email: wangguowei@hpu.edu.cn

Guangming Xu
Morden Educational Technology Center, Henan Polytechnic University, Jiaozuo, China
Email: xgm@hpu.edu.cn

Manjun Xue
School of Architectural and Artistic Design, Henan Polytechnic University, Jiaozuo, China
Email: xuemanjun@hpu.edu.cn

Abstract—To solve the problems of unified identity authentication between heterogeneous application systems that emerged in the procedure of informatization construction. The paper presents a 3-tier architecture solution that contains user tier, unified authentication tier and application system tier to realize unified identity authentication between legacy system and new application system. For legacy systems, the paper presents a method of double authentication that users firstly log on unified identity authentication system for unified identity validation, then securely transmit authenticated unified identity information and simulate the authentication progress via the authentication mechanism of legacy systems for permission distributing. For new application systems, the paper presents a method of centralized authentication through a relatively independent Role Based Access Control (RBAC) identity authentication model that consists of organization, user, role and permission entries to realize the logical separation of users and access permission, then designs a Directory Information Tree (DIT) structure based on the RBAC model for permissions distributing and access controlling.

Index Terms—Unified Identity Authentication; LDAP; RBAC; DIT

I. INTRODUCTION

With the rapidly developing of informationization and network scale, many enterprises developed information application systems to improve work efficiency, however, since having no unified consideration and arrangement, these systems were developed by different technologies in different environments at different periods, besides, database management system, operating system and programming language are all different from each other, which brought great inconvenience for business interchange and data synchronization. This kind of operation mode seriously influences the transmission of sharing data. Enclosed information in the separated application systems often causes data inconsistency, forming the isolated information island. To solve the problems of isolated information island and realize data synchronization and data integration between heterogeneous application systems, the popular solution is constructing an integrated platform. While with the constantly growing of the quantity of application systems and their user capacity, it needs to identify personal information and authorize operation authority of different identity on the integrated platform.

In heterogeneous systems, the general method for identity authorization is establishing an independent module that database server stores the user’s password verifier information and identity authorization module checks the username match the password or not. In heterogeneous environment, users must remember multiple user account and password because there exists so many application systems. Usually, users might set a same user account in different application systems, which makes much easier for attackers steal users’ password from low security level to attack system of higher security level [1, 2].

An identity authentication system generally has two main duties: identity verification and recognition. Identity verification verifies whether the user is authorized or not, while recognition makes out the identities of the authorized users [3]. Unified identity authentication means that users only conduct one initiative identity authentication process to access all authorized resources on the network, without the need for participation in different identity identification process in different application systems [4]. At present, the research on unified identity authentication based on different principle has produced different implementation models, gateway, proxy and Lightweight Directory Access


Protocol (LDAP) based unified identity authentication models are commonly used.

LDAP directories have proliferated as the appropriate storage framework for various and heterogeneous data sources, operating under a wide range of applications and services [5]. By using standard protocol of LDAP, LDAP based unified identity authentication systems realize unified authentication through the directory server that stores user account information of multiple application systems and functional directory module that developed for authentication and permission. Because the whole resources access control policy in the achievement of integrated heterogeneous application system is so complicated that if it was deployed in LDAP server it will adds additional burden to LDAP server and produces inevitable problems to normal operation of application systems, and also, it needs to modify parts of the source code of application systems for the purpose of unified identity authentication. Because most application systems were developed by commercial enterprises and the source code was confidential, which bring more difficulties in the progress of directory services based unified identity authentication.

Because isolated information island, the problem of user account information transmitting between different heterogeneous application systems makes unified identity authentication an urgent need for enterprises to develop a unified, secure, easily managed user management systems with good portability and extensibility.

A. Our Contributions

In this paper, we present two methods to solve the problems of unified identity authentication between heterogeneous application systems.

For developed application systems, we present a method of double authentication that users firstly log on unified identity authentication system for unified identity validation, then securely transmit authenticated unified identity information and simulate the authentication progress via the application systems' own authentication mechanism for permission distributing.

For new application systems or undeveloped application systems, we present a method of centralized authentication mechanism by designing an independent RBAC identity authentication model and a DIT structure for unified identity authentication.

B. Outline

Rest of this paper was organized as follows. Section II provides an overview of the theory of LDAP and RBAC, section III describes the functions and architecture of the whole solution, section IV and section V detail the solution for legacy system and new application systems, section VI design a DIT based on the solution, at last concludes this paper.

II. RELATED WORK

A. About LDAP

The LDAP is a simplified version of the X.500 directory access protocol, which has so many advanced features that it more easily adapted to meet custom needs, unlike X.500, LDAP supports TCP/IP, which is necessary for internet access [6, 7]. Directory services use a special database that stored all sorts of information resources and provide an appropriate framework for accessing a variety of information to realize information collection and concentrated control. In LDAP, all directories are composed by entries, the stored data are arranged as a tree-like structure which provides basic information about persons or web services, like name, email address, organization, phone number, digital certificate, password hashes, group membership, etc [8]. Entries are composed of attributes, which have a type and one or more values. LDAP retrieval performance is more rapidly, it is optimized for reading, browsing and searching, and the query speed is higher than a normal relational database, which is in accord with the characteristics of higher effective and huge storage [7, 9].

The importance of LDAP does not comes from the ability to look up a user's basic information, but stems from the fact that it can store all kinds of access control information through storing attributes. In other words, LDAP can serve as an alternative privilege management infrastructure by doing away with attribute certificates. The greatest advantage of LDAP is that client program can be readily available on any computer platform and the data distribution scale can be flexibly expanded because of its cross-platform and standard protocols that enable corporate directory entries to be arranged in a hierarchical structure that reflects geographic and organizational boundaries [10]. So users do not have to worry about what server LDAP directory should be deployed.

LDAP architecture is composed of information, naming, functional, and security models. Information model describes the structure of information stored in LDAP directory, naming model describes how data in LDAP directory is organized and identified, functional model describes what operations can be performed in LDAP directory, and security model describes the security mechanism in LDAP directory.

B. About RBAC

The RBAC model is the research hotspot in access control domain to handle authorization for the user through the permission with respect to role [11]. It appeared a series of models, for example, RBAC96, ARBAC97 (AdministrativeRBAC97), ARBAC99, ARBAC02 and NIST RBAC (National Institute of Standards and Technology RBAC) [12].

The earliest mature role based access control model is the RBAC96 model proposed by Sandhu et al [13]. The RBAC model uncouples users and permissions through roles, which provide a bridge between them. It is characterized by the notion that permissions are assigned to roles, and not directly to users, and also, users are assigned appropriate roles according to their job functions. It indirectly acquires the permissions associated with those roles, which greatly simplifies the management of permissions, and widely used in many information systems [12].
Generally, RBAC consists of five components: user, role/group, permission, session and constraint.

**User** is the owner of permission who interacts with the information resources, group is the unit and carrier of permissions distributing. In RBAC, permission is granted to groups other than users. Groups can include other groups, and also can contain users [14]. Users can inherit permissions from the groups.

**Role** is a collection of certain permissions that describes the duties and permissions of the user who is granted the role. In RBAC, roles represent functions within a given organization and authorizations are granted to roles instead of to single user [15]. Role can be arranged in a role hierarchy, where senior-role inherits the permissions from its junior-role [16].

**Permission** is a specific set of actions that impact on one or more objects.

**Session** is one operation that roles used to access resources according to the permissions.

**Constraint** is a set of restrictions of rules that includes static and dynamic constraints.

The RBAC basic model RBAC0 was defined as follows:

- $U$, $R$, $P$, $S$ respectively expresses users, roles, permissions and sessions.
- $PA \subseteq P \times R$ expresses the many-to-many permission-to-role assignment relationship.
- $UA \subseteq U \times R$ expresses the many-to-many user-to-role assignment relationship.
- $RH \subseteq R \times R$ expresses the hierarchical relationship in roles.

$users : R \rightarrow 2^U$ expresses the mapping functions from roles to users [17].

$perms : R \rightarrow 2^P$ expresses the mapping functions from roles to permissions.

$roles : P \rightarrow 2^R$ expresses the mapping functions from permissions to roles.

### III. System Architecture

#### A. Functions Describing

**User management:** User management provides unified authentication patterns and policies for application systems to identify users’ permission and manage users’ information, such as creation, modification, deletion of user account.

**Authorization management:** Authorization management provides authorization management services for users and application system, authorization function from user identity to application system, authorization and access control mechanisms that corresponding to the actual application processing mode but irrelevant to the development and management of application system, access policy inheritance, group members authorization, role-based access control features. All security policy operations such as formulation, modification and deletion are realized by unified platform. In order to achieve the goal of unified management of internal security policies and high availability, the authorized service can be replicated.

**Single Sign On:** Users don’t need to enter different user names and passwords more than one time in different application systems, the authorized resources in different application systems can be available by single sign on.

#### B. Overall Architecture

The solution of unified identity authentication this paper proposed confronts with the problems that come from two sorts of application systems, first one is developed and working application systems that have their own user authentication systems and identity authentication mechanism that users need to enter a user name and password, only managed resource can be accessed by authorized users. Another is the new application systems that need to be integrated into the unified authentication system, in order not to repeat the development of authentication modules and user management module during development, the unified identity authentication system must provides a complete policy for user authentication. The design of the unified identity authentication system should provides an interface of which the new application system can make use to achieve identity authorization and permission management.

The whole solution is a 3-tier architecture that contains user tier, unified authentication tier and application system tier. User tier mainly includes users from all sorts of application systems. Unified authentication tier realizes unified user management and authorization policy management, in order to realize single sign on, it need to establish a unified basic information table to record information about users and application systems, the key field of the table can be distinguished from the user account from any application systems. Application system tier consists of all sorts of application systems that have their own authentication mechanism and new application systems.

### IV. Solution for Developed Application Systems

User information in every application system should be registered in unified identity authentication system with which the authentication system server can compares the data stored in directory database to retrieve user’s identity, meanwhile, the unified basic information in unified identity authentication system must be distinguished from all application systems for unique identity verification and recognition.

The table structure of unified basic information table is shown in table I.

<table>
<thead>
<tr>
<th>Table I. Unified Basic Information Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>user_name</td>
</tr>
<tr>
<td>password</td>
</tr>
<tr>
<td>role</td>
</tr>
<tr>
<td>application_system</td>
</tr>
</tbody>
</table>

There is a set of access control policies achieved in application systems, each access control policy may be different from others. The unified identity authentication system we presented uses a method of double authentication to solve the problems of unified identity authentication [18].

The main idea is that users firstly log on unified identity authentication system and input unified basic information such as username and password for unified identity validation, if the username and password matches,
unified identity authentication system redirects the users’ browser to a portal page that contains a list of authorized application systems’ address, when users click one address of the list, unified identity authentication system will retrieves the corresponding identity information and sends it to the application system server for login progress simulation and permission distribution through its own authentication model based on user’s identities and roles, at last, application system responds the needed data or business to users’ browser.

The method of double authentication is shown in Fig. 1.

In this procedure, users don’t need to remember different user accounts from different application systems or browser different login pages to click the submit button, they only enter unified basic username and password for identity validation one time, but they can obtain all resources they need.

**Table I. Unified Basic Information Table**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Field value 1</th>
<th>Field value 2</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ua_userid</td>
<td>10460070911</td>
<td>10460070911</td>
<td>user’s identity in UIA</td>
</tr>
<tr>
<td>leg_name</td>
<td>OA</td>
<td>mail</td>
<td>Name of legacy system</td>
</tr>
<tr>
<td>leg_userid</td>
<td>zhang_san</td>
<td>zhangsan</td>
<td>user’s identity in legacy system</td>
</tr>
<tr>
<td>leg_pubkey</td>
<td>30 81 89 02 81…</td>
<td>65 83 55 9d</td>
<td>Pubkey of legacy system</td>
</tr>
<tr>
<td>leg_domain</td>
<td>oa.hpu.edu.cn</td>
<td>mail.hpu.edu.cn</td>
<td>Domain name of legacy system</td>
</tr>
<tr>
<td>leg_ip</td>
<td>192.168.11.71</td>
<td>192.168.11.78</td>
<td>IP address of legacy system</td>
</tr>
</tbody>
</table>

**Figure 1. Procedure of double authentication**

In this procedure, users don’t need to remember different user accounts from different application systems or browser different login pages to click the submit button, they only enter unified basic username and password for identity validation one time, but they can obtain all resources they need.

**A. Procedure of Unified Identity Authentication**

User visits unified login page and sends a request for authentication through the basic username and password that registered in unified identity authentication system.

Unified identity authentication system verifies the received username and password. If the username and password sent by users match the information that registered in authentication server, unified identity authentication system generates a token then responses the token and the list of authorized application systems to user client. Token is a random number generated by using (1). Equation (1) is a mathematical function that contains two parameters, one parameter is a random number and the other is the converted login time.

$$RN_{token} = f(RN, T)$$  \hspace{1cm} (1)

User client sends a service request to one application system of the list by carrying the token.

After received the request, application system sends the received token and other information that contains its own unique identity to unified identity authentication server for validity verification.

Unified identity authentication server verifies the validity of the received token, if the received token match the token that it generated, unified identity authentication server retrieves the correlated user account of the application system that registered in database according to the unique identity information that it received.

Unified identity authentication system responds the encrypted user account to application system.

Application system executes the following three operations:

1. Decrypts the received encrypted user account information.
2. Checks the user’s role and permission according to its own authentication mechanism.
3. Responses the resources that user needed to user’s client.

The whole procedure is shown in Fig. 2.

In this procedure, the secure transmission of user account from unified identity authentication system to application system is very important. The confidentiality and integrity of user account can be protected through encryption, using a combination of asymmetric and symmetric cryptographic can offer the efficiency of symmetric cryptography while maintaining the security of asymmetric cryptography [19, 20].

**B. Secure Transmission of User Account**

To realize simultaneous protection of user account confidentiality and integrity the paper presents a method that unified identity authentication system encrypts user account through symmetric cryptography and legacy system decrypts it. The session key is securely distributed through asymmetric cryptography, and each communication use one-time pad mod.

Suppose $E_p$ and $D_p$ respectively denotes encryption algorithm and decryption algorithm of asymmetric cryptography, $E_s$ and $D_s$ respectively denotes encryption algorithm and decryption algorithm of symmetric cryptography, $A$ and $B$ respectively denotes unified authentication system and legacy system, $K_{es}$ and $K_{ds}$ respectively denotes the public-key and private-key of unified authentication system, $K_{as}$ and $K_{pa}$ respectively denotes the public-key and private-key of legacy system, $K_s$ denotes session-key.

The whole process of secure transmission of user account as follows:

1. A gets $K_{pa}$ of $B$ by querying the unified information table.
A generates a $K_S$ at random and encrypts it by using (2), then sends the encrypted $K_S$ to $B$.

$$C_{KS} = E_p(K_S, K_{db})$$  

(2)

$B$ gets $K_S$ through decryption (3).

$$K_S = D_p(C_{KS}, D_{db})$$  

(3)

$A$ and $B$ conduct secure communication through (4) and (5) for user account transmitting.

$$C_{userid} = E_s(K_S, userid)$$  

(4)

$$userid = D_s(K_S, C_{userid})$$  

(5)

$A$ and $B$ destroy $K_S$ when the communication finished.

V. SOLUTION FOR NEW APPLICATION SYSTEMS

New application systems don’t have to develop user identity authentication module. In our solutions, users log on unified identity authentication system for identity validation, new application systems call the interface that unified identity authentication system provided to verify user’s legal identity. This need the administrator of unified identity authentication system adds the information about new application system, including roles, permissions, and other information to LDAP directory server.

The integration of user identity authentication and access control is the key function in information security infrastructure of the new application system. The unified identity authentication module concerns the ways according to which users can access resources and specifies which subjects can access which objects for performing which actions. In practice, the commonly used access control is RBAC that provides user authentication to identify the user’s authorities for assigning different permissions [21].

A. Procedure of Unified Identity Authentication

Users visit unified login page and send a request for authentication through the username and password that registered in unified identity authentication system. If the username and password sent by users match the information from the authentication server, the unified identity authentication system redirects users’ client to the portal page that contains a list of authorized resources through interface that provides portable and other functions according to the users’ role, permission and operation.

User client sends a service request to one application system of the list.

Application system responses the resources that user needed to the client.

The whole procedure is shown in Fig. 3.

© 2014 ACADEMY PUBLISHER
B. Design of RBAC model
The RBAC model provides a flexible and efficient mechanism in security of the solution [22]. In order to realize the logical separation of users and access permission and reduce authorization management complexity and improve the performance of reusability in a variety of environments, by integrating the original concept of RBAC and introducing new elements and mechanism the solution designs a relatively independent and universal model that shown in Fig. 4.

\[ U = \text{User} = \text{Human beings, machines, networks, or intelligent agents that can perform some activities} \ [23]. \]

\[ R = \text{Role} = \text{A set of permissions necessary to access resources.} \]

\[ G = \text{Organization} = \text{User’s affiliation that can be divided into top level organizations and sub level organizations.} \]

\[ P = \text{Permissions} = \text{An approval of a mode of access to a resource.} \]

\[ O = \text{Operation} = \text{Executions of a program specific function that invoked by a user, usually include addition, modification, deletion.} \]

\[ A = \text{Resources} = \text{Entities that contain or receive information, include legacy systems and new application systems.} \]

When users access the application systems, after the identity validation, the unified identity authentication system automatically retrieves the user’s entire roles information according to their logon information then combines the resources of all roles and resources of organization into two different sets, at last obtains user’s final resources through the intersection operation of the two resources set.

Resource of user roles is a union set which is \( R_s = A_1 \cup A_2 \cup \ldots \cup A_n \), Resource of user organization is \( R_o \), user’s final resources is an intersection set \( R = R_s \cap R_o \).

C. Design of Roles DIT
In directory service, users and roles are directly linked because various application systems independent from each other. Coarse roles are equivalent to an administrator of application system or a roles group that responsible for the user addition, user approval and authorization management within the system. Fine roles are parts of coarse roles, just like a member of a group that has certain permissions, for example, some users have modification permissions, while others have addition permission.

Roles DIT is shown in Fig. 7.

VI. DESIGN OF DIT
The key point of unified identity authentication is how to design a reasonable DIT structure [24]. Each entry of DIT contains an object class that describes a particular object and one or more attributes that describe the entry. So, our realization is begin with the design of attributes and object class, firstly, we define the object class that corresponding to the entry then design what attributes the object should contains.

A. Design of Root Directory Entry
According to RBAC model, it needs to set the root directory entry firstly. The root directory entry of the LDAP directory can be set to the following Fig. 5.

\[ \text{DIT} = \text{DIT} = \text{DIT} = \text{DIT} = \text{DIT} \]

B. Design of Users DIT
Users of the directory contain two categories: individual users of identity authentication system and application systems users. Design of user object class needs to store users’ basic information, such as telephone NO, e-mail address and information about application system users, such as domain names, IP addresses, and so on.

Based on the above analysis, the user directory can be divided into two branches which are individual users of identity authentication systems and application systems users.

Users DIT is shown in Fig. 6.

\[ \text{Users DIT} = \text{Users DIT} = \text{Users DIT} = \text{Users DIT} \]

C. Design of Roles DIT
In directory service, users and roles are directly linked [25]. Roles can be divided into coarse roles and fine roles because various application systems independent from each other. Coarse roles are equivalent to an administrator of application system or a roles group that responsible for the user addition, user approval and authorization management within the system. Fine roles are parts of coarse roles, just like a member of a group that has certain permissions, for example, some users have modification permissions, while others have addition permission.

Roles DIT is shown in Fig. 7.
In the roles entry, the following three attributes must be implied to realize role constraint:

**Capacity:** The maximum number of users authorized by this role.

**Conflict:** If role A conflicts with role B, the same user cannot simultaneously possess both roles.

**Depend:** If role A depends on role B, role B must be granted to user before role A.

### D. Design of Permissions DIT

Permissions define what actions the users can perform in the application system. Different users have different permissions. In order to avoid overlapping, whole permissions should be divided into unit permissions. For convenient permissions management, each application system is defined as top level and permissions are confined in different sub systems.

Permissions DIT is shown in Fig. 8.

![Permissions directory](image)

**Figure 8.** Permissions directory

### E. Design of Organizations DIT

For convenient operation of large number of organizations, each organization is defined as top level and all departments in organization are defined as sub level.

Organizations DIT is shown in Fig. 9.

![Organizations DIT](image)

**Figure 9.** Organizations DIT

### VII. CONCLUSIONS

Unified identity authentication is an important point of constructing integrated platform to realize data integration between heterogeneous application systems. The solution this paper presented has practical value to solve the problems of identity authentication in legacy application systems and new application systems, which is helpful for users to reduce operating frequency and remember multiple user accounts and passwords. The solution can avoids the redundant development of authentication module and does not modifies the source code of the legacy system, which is beneficial to enterprises for resources-saving. However, parts user information from different systems need to be registered in unified identity authentication server, if this procedure was achieved by manual operation, it will bring much inconvenience because the data format, operating system, development technologies and environment, database and database management systems, programming language are all different from each other, so, future work will be focused on the research of automatic user information synchronization and data migration under distributed heterogeneous environment.

### ACKNOWLEDGMENTS

The research of this paper was supported by soft science research program of Henan province (1024004500064) and information technology in education research program of Henan province (ITE12103). The authors thank the research team for the valuable discussions and suggestions. Information technology education and research projects in Henan province.

### REFERENCES


Guowei Wang, male, was born at Pingdingshan, China in June 1979. He is a Ph.D. student in Henan Polytechnic University. The major field of study is system architecture of computer. He engages in teaching and research in computer security, computer networks and database applications.

Guangming Xu earned her master degree from Henan Polytechnic University, China, in 2012. Her research fields mainly include geographic information system.

Manjun Xue earned her master degree from Henan Normal University, China, in 2012. Her research fields mainly include computer-aided design and website optimization.