A Workflow-based RBAC Model for Web Services in Multiple Autonomous Domains

Zhenwu WANG
School of Mechanical and Information Engineering, China University of Mining and Technology, Beijing, China
Email: wangzhenwu@126.com

Xuejun ZHAO¹, Benting WAN², Jun XIE³, Pengfei BAI¹
¹School of Mechanical and Information Engineering, China University of Mining and Technology, Beijing, China
²Software and Communication Engineering institute, Jiangxi University of Finance and Economics, Nanchang, China
³Information Engineering institute, Capital Normal University, Beijing, China

Abstract—A workflow-based RBAC model for web services (WFRBAC4WS) has been proposed in this paper. The model organizes web services in different autonomous domains through workflow mechanism, and maps RBAC model to tasks of workflow model. The paper details the authorization procedure of WFRBAC4WS model, the lifetime management, the extension of authorization constraint and the formal descriptions of the proposed model. Compared with other RBAC models for web services, this model not only combines RBAC model to workflow, but also describes the interactions between workflow mechanism and RBAC model in web services environment, the authorization work of this model is dynamically and comprehensively.

Index Terms—workflow, RBAC, web services, autonomous domains

I. INTRODUCTION

Service-oriented architecture (SOA) is a framework for distributed systems, which is platform-independence and is constructed by components. Web service becomes the most popular implementation of SOA, and it has many advantages, such as high development efficiency, fast response ability, good reusability, and so on. The service requesters and providers both have the high dynamic because of the heterogeneity of environment and the variousness of operation methods, it is necessary that access control methods of web services need adapt the variety dynamically. The access control of web services must face the following problems [1,2].

(1) Cross-domain

The traditional access control models base on single autonomous domain, the providers and requesters are both in the same domain, and they can “recognize” each other. But web services generally are deployed in different domains and the service requesters and providers cannot “recognize” each other, this problem is called the access control issue among “strangers” [3,4].

(2) Dynamic authorization

The traditional access control models generally assign permissions to subjects according to certain rules, and then save these assignment relations, this procedure is called static authorization. But in web services environment, web services are distributed in multiple autonomous domains and the authorization activities are dynamic. The subjects which make the requests and the objects which provide service resources both have high dynamic characters, the dynamic traits of subjects are caused by the variety of operations and the heterogeneous environment, the dynamic characters of web services represent that the web services composition is dynamic, so we need a dynamic authorization mechanism.

(3) Loose coupling

Web services are deployed in different autonomous domains, the coupling relationships among web services are decided by the workflow of composite web services. The coupling relationships are different in contexts of different composite services because these components can be reused in many applications.

Owing to the above traits, we cannot directly adopt the traditional access control models which usually are used to web services in single domain, and the security problem of web services is the obstruction of web services popularization. Many scholars discussed the security problem of web services in various aspects [5-9]. Literature [5] proposed an attribute-based access control (ABAC) model based on XACML in Web Service, literature [6] designed a cross-domain trust-based access control model for Web service, which based on XACML and WS-Security, and literature [7] discussed the security technologies based on SOAP, which adopted methods of encryption, digital signature and authorization. Recently, some literatures discussed the access control problem of composite web services in multiple domains, literature [8] detailed a dynamic multiple domains access control model based on RBAC and gave a role mining algorithm to find the role set with minimized permissions, literature [9] proposed an UCON enhanced business process dynamic access control model, which unbounded the coupling relationship of organization model and the process model. Business process access control mechanism is a difficult problem in Web services composition application [9], literature [8][9] analyzed the permissions which are needed by services execution according to the procedures of composite web services,
but they only use a little idea of workflow and don’t
detail how to map roles and web services to the tasks of
workflow, which represent the business process. In this
paper, we use workflow mechanism to describe the
business processes, and allocate role instances to the tasks
of workflow instead of the static role in run-time
environment, this method adds flexibility and dynamic to
the permission allocation strategy.

The rest of the papers are organized as follows. Section
II provides an overview of the theory of NIST RBAC
model and the review of workflow mechanism; Section
III details the principle and formal descriptions of the
proposed model, an example illustrates the model’s
application in section IV and at last section V concludes
this paper.

II. THE RELATED WORK

A. The NIST RBAC Model

The RBAC model is the research hotspot in access
control domain, and it appeared a series of models, for
example, RBAC96, ARBAC97 (Administrative
RBAC97), ARBAC99, ARBAC02 and NIST RBAC
(National Institute of Standards and Technology RBAC),
these models have gradually been perfected. 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. Users are assigned appropriate roles
according to their job functions, and hence indirectly
acquire the permissions associated with those roles, and it
is widely used in many information systems.

The NIST RBAC model has four components, they are
Core RBAC, Hierarchal RBAC, and two units of
Constraint RBAC. The Core RBAC is the smallest
element set which can build an access control system, it
includes the following conceptions.

(1) \( Users = \{ u_1,u_2, \ldots ,u_n \} \), it is the set of all the users.

(2) \( Roles = \{ r_1,r_2, \ldots ,r_n \} \), it is the set of all the roles.

(3) \( Ops = \{ op_1,op_2, \ldots ,op_k \} \), it describes the set of all
the operations.

(4) \( Objects = \{ ob_1,ob_2, \ldots ,ob_l \} \), it is the set of all
the access objects.

(5) \( Session = \{ s_1,s_2, \ldots ,s_p \} \), it represents the set of all
the sessions.

(6) \( Perms = 2^{Ops \times Objects} \), it is the set of all the
permissions.

(7) \( UA \subseteq Users \times Roles \), it describes a many-to-many
relation from user set to role set, and represents
that users are given to roles.

(8) \( PA \subseteq Perms \times Roles \), it describes a many-to-
many relation from permission set to role set, and represents
that roles are given to permissions

(9) \( assigned\_users : (r : Roles) \rightarrow 2^{Users} \), return the
user set which are given to role \( r \).

\[ assigned\_users(r) = \{ u \in Users | (u, r) \in UA \} \]

(10) \( assigned\_perms : (r : Roles) \rightarrow 2^{Perms} \), return the
permission set which are given to role \( r \).

\[ assigned\_perms(r) = \{ p \in Perms | (p, r) \in PA \} \]

(11) \( assigned\_roles : (u : Users) \rightarrow 2^{Roles} \), return the
role set which are given to user \( u \).

\[ assigned\_roles(u) = \{ r \in Roles | (u, r) \in UA \} \]

(12) \( op(p : Perms) \rightarrow Ops \), return the operations
which are associated with certain permission \( p \).

\[ op(p) = \{ op \in Ops | (op, obj) = p \} \]

(13) \( ob(p : Perms) \rightarrow Objects \), return the objects
which are associated with certain permission \( p \).

\[ ob(p) = \{ ob \in Objects | (op, obj) = p \} \]

(14) \( user\_session(u \in Users) \rightarrow 2^{Sessions} \), return the
sessions which are associated with certain user \( u \).

(15) \( session\_roles(s : sessions) \rightarrow 2^{Roles} \), return the
roles which are associated with certain session \( s \), that
is

\[ session\_roles(s) \subseteq \{ r \in Roles | (session\_users(s), r) \in UA \} \]

(16) \( session\_perms(s : sessions) \rightarrow 2^{Perms} \), return the
permissions which are associated with certain
session \( s \), that is

\[ session\_perms(s) = \bigcap_{r \in session\_users(s)} assigned\_perms(r) \]

Many literatures [10-13] improved the NIST RBAC
model from different aspects and discussed its discrete
form in different application domains. In literature [10],
we introduced visual data muster into RBAC model and
proposed a three-dimensional space RBAC model, which
constrained permissions from three dimensions and
enhanced the ability of access control. Literature
[11] gave a generalized temporal and spatial RBAC model,
which is applied to mobile service applications; Literature
[12] implemented the RBAC model in healthcare ad hoc
network, literature [13] gave a united access control
model for systems in collaborative commerce, which
combined the advantages of RBAC, task-based
authentication control (TBAC), attribute-based access
control (ABAC) and automated trust negotiation (ATN).

Some scholars [16-20] discussed RBAC model in
distributed environment. WANG Hao[16] made all of the
nodes manage the user’s right and adopts different
encryption policy according to the ability of the device
object to realize high level of control and low
consumption of calculation, LANG Bo[17] defined the
semantics properties elements of trust in the context of
access control in distributed systems, ZHAO Jun[18]
designed a distributed secure access control system,
HANG Shuai[19] proposed a role mining algorithm
according to the process structure, and ZHAO Jun[20]
studied the application of distributed multi-channel MAC
protocols in wireless sensor network(WSN).

Some literatures[21-23] applied RBAC model to web
services, which are the new software model of distributed
systems.R.Wonoboesodo [21] introduced RBAC model
to web services and assigned roles to web services through classifying them into single web service and composite web services, literature [22] proposed a role and task-based access control model (RTBAC), which bases on the traditional RBAC, TBAC and security workflow model, literature [23] presented a new dynamic hierarchical RBAC model for web services in order to enhance the flexibility and expansion. All the above papers extended RBAC model from different aspects, but they cannot depict the distributed characters of web services security mechanism. This paper combines RBAC model to workflow in order to describe the security problem more comprehensively.

**B. The Workflow Mechanism**

Workflow is the automation of a business process in whole or part, and the workflow idea has been used to various domains [14,15]. For software development of service-oriented architecture (SOA), the workflow idea provides a promising solution for organizations to achieve their business goals by interactions and collaborations between web services. Some scholars [24-26] introduced workflow mechanism to extend the access control mechanisms, literature [24] formalized web services and workflow processes, and proposed a type system to ensure that the specified TBAC policy is respected during system reductions, literature [25] introduced two notions of services and authorization transfer to describe dynamic service-oriented architecture and proposed a workflow-based and services-oriented role-based access control (WSRBAC) model, literature [26] also gave a access control model which has some workflow concepts to protect web services in business process, literature [16] presented a dynamic hierarchical role-based access control model for web services, which improved the flexibility and independence and expansion of access control for web services.

Although the above papers used workflow ideas to extend their access control models, they did not describe how to control the workflow and who control them, and they did not discuss the detailed interactions of workflow mechanism and RBAC model in web services environment. This paper would detail the configurations of RBAC model in workflow modeling component, the model running theory in workflow executing component and the formal descriptions of the whole model. The workflow-based RBAC model discussed in this paper is based on my paper [27], in that paper, we describe workflow as a groupware of subtasks and actions, and the detailed theory of workflow system can refer literature [27].

**III. THE WORKFLOW-BASED RBAC MODEL FOR COMPOSITE WEB SERVICES**

**A. The Principle of WFRBAC4WS**

Generally, the workflow management system includes two parts, they are workflow modeling component and workflow executing component. The workflow modeling component provides a build-time environment for workflow constructors, and can be used to define, analyze and manage workflow model, the executing component provides a run-time environment to workflow execution. The paper will discuss the procedure of permission definition and allocation during the two parts.

1. **Basic conceptions**

In order to describe the workflow-based RBAC model for composite web services (WFRBAC4WS), we define the following conceptions.

1. **Role instance**: Compared with role which is the key notion in RBAC model, role instance represents an instance of role, and it has the lifetime, that is to say, it is a dynamic conception which is generated when role is activated by users.

2. **FlowStep**: FlowStep is a segment of workflow, and it is composed by a series of FlowSubTasks and FlowActions.

3. **FlowSubTask**: FlowSubTask represents a functional operation, it describes the necessary executing task in workflow process, and it also can contain other FlowSubTask. FlowSubTask can map to one web service, or a series of web services, it relies on the function granularity.

4. **FlowAction**: FlowAction describes the moving conditions of FlowSubTasks, it controls when the workflow is triggered to the next segment.

5. **Global web service**: Global web service (gws) is the web services which can assign permissions to other web services and configure the whole workflow. Compared with common web services, this type web services control the permissions configuration and distribution in the entire distributed environment, and generally, global web service can be bound to FlowAction.

6. **Main Domain**: Main Domain is the domain which runs Global Web Service, and it is in charge of the execution of WFRBAC4WS model.

2. **The principle of WFRBAC4WS model**

The principle of WFRBAC4WS model can be described as figure 1, the authorization work can be divided two parts which are corresponding to workflow definition component and executing component.

1. **authorization definition**

Authorization definition runs in workflow definition step and system manager can do this work. This procedure can be describe by the full line part of figure 1. Users obtain roles and roles can be assigned to FlowStep, then FlowSubTask and FlowAction can be distributed to FlowStep, at last web services are bound to FlowSubTask and global web services are bound to FlowAction.

2. **authorization activation**

In workflow executing step, system will automatically generate a role instance according to certain role which is associated to certain user, and the role instance are bound to FlowSubTask and FlowAction, which are mapped to web services and global web services separately, that is to say, when a web service return computed results to system, the role instance which connected with this web service will be released because the role instance has lifetime.
3. The lifetime of role instance

A FlowSubTask and FlowAction compose the basic workflow step, and the relationship between them can be described as figure 2. From figure 2 we can see that one FlowSubTask or more than one FlowSubTask can trigger a FlowAction, and a FA can trigger one or more than one FlowSubTask. Because role instance has lifetime and it is assigned to FlowSubTask and FlowAction, the lifetime of role instance can be computed by the following situations.

1. The lifetime of role instance equals the executing time of web service which is mapped to the FlowSubTask which is described by situation (a).
2. The lifetime of role instance is the max executing time of web services which are mapped to these FlowSubTasks which are described by situation (b).
3. The lifetime of role instance is the executing time of web service which is mapped to the FlowAction which is described by situation (c) and (d).

Figure 2. The relationship between subtasks and action in workflow

The role instance is a triple and it is described as follow.

Role instance = {userid, roleid, lifetime}

When system calls a web service, global web service will search the role and user which connect with this web service, and generate a responding role instance. Global web service will generate the lifetime according to the execution time of this web service, the execution time is one parameter of quality of service (QoS), we can get it from service provider. Generally, lifetime can be set longer than execution time, for example 1.5 times. When the service results return, the role instance will be released and if the services results cannot return in the lifetime, the role instance also must be released for the security reason.

4. Mapping web services to workflow model

The dotted arrows in figure 3 present the mapping relationships, and the solid arrows describe the called relationships, and web services in different autonomous domains can map to different FlowSubTask in each FlowStep, and one web service can also call another web service in different autonomous domain, so there are the following relationships for role instances.

Figure 3. Mapping web services to workflow model

The status of web service

Web service has lifetime, and it also has some different status and transforms among these status. We introduce the following web service statuses, which are similar to the statuses of process in operation system and are described in Literature [25]. We also adopted these statuses and used them to describe web services.

Figure 4. the status of role instance
Web service has five statuses, they are sleep, ready, run, suspend, and stop.

(1) Sleep status: it represents that one web service is the inactivated status.
(2) Ready status: it represents that the web service has completed the current preparatory work and the running condition has been satisfied.
(3) Run status: it represents that one web service has been activated successfully and is running.
(4) Suspend status: it represents that one web service has been paused to execute for some reasons.
(5) Stop status: it represents that one web service has been terminated, or finish its task.

The status transformation among the above statuses can be described by figure 4, and from it we can see that web services can be called when certain user accesses web services. When one user calls a web service, the system will judge whether user is legal or not according to its role instance.

From figure 3 we can see that web services are mapped to FlowSubTasks and FlowActions, so we need discuss the relations among them.

We can describe the relationship among FlowSubTasks as follows.

(1) Synchronized relationship. It shows that one FlowAction is triggered only if all the web services associated with FlowSubTasks complete their work.
(2) Asynchronous relationship. It shows that one FlowAction is triggered when one web service which is associated with FlowSubTask completes its work.
(3) Mutual exclusion relationship. It shows that two web services which are mapped to FlowSubTasks cannot execute at the same time.

The relationship between FlowSubTask and FlowActions or between two FlowActions can be described by dependency relation.

(4) Dependency relationship. a web service which is mapped to FlowAction is triggered only when FlowSubTasks complete their work, The FlowActions in workflow also satisfy the dependency relationship.

B. The Formal Description and Traits of WFRBAC4WS

1. The formal definition

(1) $FS = \{f_{s1}, f_{s2}, \ldots, f_{sn}\}$, it is the set of all the FlowSteps.
(2) $FST = \{f_{st1}, f_{st2}, \ldots, f_{stn}\}$, it is the set of all the FlowSubTasks.
(3) $FA = \{f_{a1}, f_{a2}, \ldots, f_{an}\}$, it is the set of all the FlowActions.
(4) $WS = \{w_{s1}, w_{s2}, \ldots, w_{sn}\}$, it is the set of all the common web services.
(5) $GWS = \{g_{ws1}, g_{ws2}, \ldots, g_{wsn}\}$, it is the set of all the global web services.
(6) $RI = \{r_{i1}, r_{i2}, \ldots, r_{in}\}$, it is the set of all the role instances.

(7) $UIA \subseteq Users \times RI$, it describes a many-to-many relation from user set to role instance set, and represents users are assigned to role instances.
(8) $FSA \subseteq FS \times Roles$, it is a many-to-many relation from FS to Roles, and it shows the FlowStep set which are assigned to Roles.
(9) $FSTA \subseteq FST \times RI$, it describes a many-to-many relation from FST to RI, and it represents the $f_{st}$ set which are assigned to $RI$.
(10) $FAA \subseteq FA \times RI$, it describes a many-to-many relation from FA to role set, and it represents the $f_{s}$ set which are assigned to $RI$.

(11) $f_{s} = \{f_{s1}, f_{s2}, \ldots, f_{sm}\}$, it describes that a FlowStep includes more than one FlowSubTask and one FlowAction.

(12) $assigned\_FS(r : Roles) \rightarrow 2^{FS}$, return the FlowStep set which are assigned to role $r$, and assigned \_FS(r) = $\{f_{s} \in FS| (f_{s}, r) \in FSA\}$.

(13) $generated\_ris\_r : (Roles) \rightarrow 2^{RI}$, return the role instance set which are generated by role $r$.

(14) $assigned\_FST(r_{i} : RI) \rightarrow 2^{FST}$, return the FlowSubTask set which are assigned to role instance $r_{i}$ and assigned \_FST(r_{i}) = $\{f_{st} \in FST| (f_{st}, r_{i}) \in FSTA\}$.

(15) $assigned\_FA(r_{i} : RI) \rightarrow 2^{FA}$, return the FlowAction set which are assigned to role instance $r_{i}$ and assigned \_FA(r_{i}) = $\{f_{a} \in FA| (f_{a}, r_{i}) \in FAA\}$.

(16) $ws(f_{st} \in FST) \rightarrow WS$, return the web services which associated to FlowSubTask $f_{st}$.

(17) $gws(f_{a} \in FA) \rightarrow GWS$, return the global web services which are associated to FlowAction $fa$.

(18) $fst(f_{a} \in FS \cap f_{st} \in FST) \rightarrow FST$, return the FlowSubTask set which are connected with FlowAction $fa$.

(19) $RHI \subseteq RI \times RI$, it represents a partially-ordered set on $RI$, and is marked as $\prec$.

$r_{i1} \prec r_{i2}$ $\Rightarrow$ authorized \_FST(r_{i1}) $\subseteq$ authorized \_FST(r_{i2})

$\cap$ authorized \_users(r_{i1}) $\subseteq$ authorized \_users(r_{i2})

$r_{i1} \prec r_{i2}$ $\Rightarrow$ authorized \_FA(r_{i1}) $\subseteq$ authorized \_FA(r_{i2})

$\cap$ authorized \_users(r_{i1}) $\subseteq$ authorized \_users(r_{i2})

authorized \_users(r_{i} \in RI) $\rightarrow 2^{Users}$, and it can conclude the following formula.

authorized \_users(r_{i}) =

$\{u \in User \mid \exists ri \in RI, ri \geq r_{i} \cap (u, ri) \in UIA\}$

authorized \_fsts(r_{i} \in RI) $\rightarrow 2^{FST}$, and it can conclude the following formula.
authorized_fsts(ri) =
\{ fst \in FST \mid ri \in RI, ri \geq ri \cap (fst, ri) \in FSTA \}

authorized_fas(ri \in RI) \rightarrow 2^{FA}, and it can conclude the following formula.
authorized_fas(ri) =
\{ fst \in FA \mid \exists ri' \in RI, ri' \geq ri \cap (fa, ri') \in FAA \}

2. The extension of authorization constraint

Although the NIST RBAC model has the ability of authorization constraint, it only concerns the conflict when multiple roles assigned to users. This paper extends authorization constraint based on first order logic. We introduce the following set to extend authorization constraint.

(20) Uncertain Function OE and AO
OneElement : \text{OE}(X) = x_1, x_i \in X
AllOther : \text{AO}(X) = X - \{ \text{OE}(X) \}

We further introduce the following sets.
\text{CRI} = \{ cri_1, cri_2, ..., cri_i \}, cri_i \subseteq RI , \text{CRI} is the set of conflicted role instances.
\text{CFST} = \{ cfs_1, cfs_2, ..., cfs_q \}, cfs_i \subseteq FST , \text{CFST} describes the set of conflicted FlowSubTasks.
\text{CFA} = \{ cfa_1, cfa_2, ..., cfa_j \}, cfa_i \subseteq FA, \text{CFA} is the set of conflicted FlowActions
\text{CFS} = \{ cfs_1, cfs_2, ..., cfs_q \}, cfs_i \subseteq FS , \text{CFS} is the set of conflicted FlowSteps.

The descriptions of first order logic are as follows.
\forall r \in \text{Roles}, \forall cri \in \text{CRI} : \text{generated_fsts}(ri) \cap cri \leq 1
\forall ri \in RI, \forall cfs \in \text{CFST} : \text{assigned_fsts}(ri) \cap cfs \leq 1
\forall ri \in RI, \forall cfa \in \text{CFA} : \text{assigned_fas}(ri) \cap cfa \leq 1
\forall r \in \text{Roles}, \forall cfs \in \text{CFS} : \text{assigned_fas}(FS(r)) \cap cfs \leq 1

2. The characters of WFRBAC4WS model

Compared with the traditional RBAC model, the WFRBAC4WS model has the following characters.
(1) To refine and depict the workflow control information through introducing the conceptions of FlowStep and FlowAction. Web services provided by different providers cannot understand the whole workflow steps and it is necessary to configure and run these control information by certain global services. We use FlowStep to map common web services provided by different providers, and use FlowAction to map the global services which are in charge of workflow shift.

(2) To configure and control the access control of distributed web services through introducing the conception of global service. It is unrealistic that each web service can control the workflow shift, because different providers only provide the web services which are like “blackboxes” and have standard interfaces, and they have not duty to do these “extra work”. So the workflow control work must be done by a global service which can configure and run all the steps of workflow shift.

(3) To manage roles dynamically through introducing the conception of role instance. Compared with role, role instance is dynamical and has lifetime. When web services are called, the corresponding role instance will start and control the access permissions.

IV. EXAMPLE ILLUSTRATION

We will use an example to demonstrate how the WFRBAC4WS model works. In this example, there are three autonomous domains, and the domain 1 is the Main Domain, and it defines the workflow processes. Each FlowSubtask in workflow processes can be mapped to certain web service which is provided by different domains. In authorization definition step, Global web services will allocate roles to FlowSubtasks, and when system calls the concrete web services, the authorization task will be activated, and system will generate role instances which associate to roles, the dotted arrowheads in figure 5 represent that Global web services assign role instances to the concrete web services and manage them in their lifetimes.

V. CONCLUSION

With the rapid development of service-oriented architecture, web services bring high productiveness and low cost for software development, but the security problem is the obstruction of web services popularization. Recently, the access control of web services becomes the hot research issue and scholars discussed this problem in various aspects according to protocol stack of web services. This paper proposed a workflow-based RBAC model for composite web services, the model configures permissions according to the workflow steps which include FlowSubTask and FlowAction, and manage the authorization task dynamically in multiple autonomous domains, so this method not only can satisfy the authorization requirement in distributed environment, but also can take back the permissions dynamically.

REFERENCES


[16] WANG Hao,WU Bo,GE Jin Wen,and WANG Ping, "Distributed Access Control Scheme Based on Controlled Object in the Internet of Things",Journal of University of Electronic Science and Technology of China,v141.,n6,pp 893-898,2012


Zhenwu WANG was born in 1978, Qingdao, Shandong province, China. His main research interests include web services selection and composition, network security, and intelligent optimization algorithm. He received his M.S.and Ph.D. degrees from China University of Petroleum, Beijing, China in 2005 and 2008 respectively. He has authored/coauthored more than 20 scientific papers, and he is a fellow of Chinese Computer Federation (CCF), and Association for Computing Machinery (ACM).

Xuejun ZHAO was born in 1962,Beijing,China. Her main research interests include network security, graphics and image processing, and remote sensing technology. She received her M.S. degree from China University of Mining and Technology, Beijing, China in 1990. Now she is an associate professor in China University of Mining and Technology, she presides a number of national fund projects, and is a fellow of Chinese Computer Federation (CCF), and Association for Computing Machinery (ACM).

Benting WAN was born in 1976, Nanchang, Jiangxi province, China. His main research interests include mobile computing, parallel and distributed computing and Internet of things technology. He received his M.S. and Ph.D. degrees from China University of Petroleum, Beijing, China in 2003 and 2006 respectively. Now he is an associate professor in Jiangxi University of Finance and Economics, Nanchang, China. He is in charge of more than five fund projects of Jiangxi province, and he is a fellow of Chinese Computer Federation (CCF), and Association for Computing Machinery (ACM).
Jun XIE was born in 1982, Ji’an, Jiangxi province, China. His main research interests include parallel and distributed computing and Internet of things technology. He received his M.S degree from Jilin University, Jilin, China in 2006, and received his Ph.D degree from China University of Petroleum, Beijing, China in 2009. Now he is an associate professor in Capital Normal University, Beijing, China, he is a fellow of Chinese Computer Federation (CCF), and Association for Computing Machinery (ACM).

Pengfei BAI was born in 1991, his main research interests include Role-based Access Control (RBAC) model and now he is a student in China University of Mining and Technology, Beijing, China.