计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200700127

基于组织架构的数据权限控制模型研究与实现

程学林　杨小虎　卓崇魁
浙江大学软件学院　浙江 宁波 315103
(cxlin@zju.edu.cn)

摘　要　数据权限控制是软件系统安全性和质量的重要方面,也是 SaaS 多租户软件系统权限管理和授权访问的重要组成部分。数据权限控制的核心需求是不同角色的用户,访问的数据范围不同,如果能够设计出一套通用的数据权限控制方法,降低授权管理的复杂性,提升软件系统安全具有一定的现实意义。在以 RBAC 授权模型为理论的基础上,提出了一种基于组织架构的数据权限控制模型(Organization-Based Data Authority Control,ODAC),ODAC 模型中 SaaS 软件系统提供的各类服务统称为资源,资源分为数据受控资源和数据不受控资源,在将数据受控资源分配给角色时,指定该资源可访问的租户组织架构,用户在访问数据时,系统通过用户角色对应资源的租户组织架构,来实现数据访问控制的目的。在此基础上,基于 Spring MVC、Spring Security 和 MyBatis 框架对 OADC 模型进行了实现。多种实际生产系统使用了该模型,验证了其具有较好的通用性和可行性。

关键词:SaaS;角色;组织架构;数据权限;访问控制;受控资源
中图法分类号　TP311

Research and Implementation of Data Authority Control Model Based on Organization

CHENG Xue-lin,YANG Xiao-hu and ZHUO Chong-kui
School of Software Technology,Zhejiang University,Ningbo,Zhejiang 315103,China

Abstract　Data permission control is an important aspect of software system security and quality,and is also an important part of permission management and authorized access of SaaS multi-tenant software system. The core requirements of data permission management are users set into different roles,which has corresponding data access scopes. If a general set of data permission control methods can be designed to reduce the complexity of authorization management and improve software system security,it has certain practical significance. The common SaaS basically uses the RBAC-based permission control component to meet the needs of user data permission control. However,RBAC is still relatively complicated in configuring of permissions,and the form of ODAC to control data permissions can simplify the configuration of permissions. Based on the theory of the RBAC authorization model,an organization-based data authority control model (Organization-Based Data Authority Control,ODAC) is proposed. In the ODAC model,various services provided by the SaaS multi-tenant software system are collectively called resources. Resources are divided into data-controlled resources and data-uncontrolled resources. When data-controlled resources are assigned to roles,the organizational structure that can access the resources is specified. When users under the SaaS service tenant organization access data,the system uses the organization corresponding to the user role in the resource tenant,to achieve data access control. On this basis,the OADC model is implemented based on Spring MVC,Spring Security and MyBatis framework. Implemented with these mature frameworks,the data authority management system based on the OADC model shows good performance,guarantee for the realization of the data permission system,and reduces the difficulty of logic implementation. The model has been used in a variety of actual production systems,which has been verified to have good versatility and feasibility.

Keywords　SaaS,Role,Organization structure,Data permission,Access control,Controlled resources

1　引言

系统安全性是软件质量的重要方面,软件安全(software security)指使软件在收到恶意攻击的情形下依然能够继续正确运行及确保软件在授权范围内被合法使用的思想[1]。对用户权限加以严格的控制是软件系统安全性的重要保障。美国国家标准与技术研究院(NIST)提出的标准 RBAC 模型是目前通行的权限控制策略[2-3]。

RBAC 模型即基于角色的访问控制(Role-Based Access Control)。在 RBAC 中,权限与角色相关联,用户通过成为适当角色的成员而得到这些角色的权限[4],这就极大地简化了权限的管理。在一个组织中,角色是为了完成各种工作而创造的,用户则依据它的责任和资格来被指派相应的角色,用户可以很容易地从一个角色被指派到另一个角色。角色可依新的需求和系统的合并而赋予新的权限,而权限也可根据需要而从某角色中回收[5]。

按照控制粒度可以将权限管理分为两大类:功能级权限管理和数据级权限管理[6]。目前,Spring Security 框架较好

通信作者:卓崇魁(chonh2020@qq.com)

地解决了功能级权限管理的问题，而数据权限管理问题没有通行的解决方案。本系统设计并实现了基于组织架构的权限控制，较好地满足了组织对数据控制的需求[7]。

在 SaaS 的提供软件服务的平台中，服务产生的数据具有敏感性，由于 SaaS 租户的用户成员职责的差异，要求用户所拥有的数据访问权限也是不同的[8]。SaaS 和传统内部系统、2C 业务权限最大的不同点为 SaaS 是天然多租户，在用户的上层有一层组织的概念，组织只拥有权限的子集，并且组织可以管理部分权限[9-10]。

对于 SaaS 这样庞大的复杂的平台，数据权限的权限系统设计得越全面、精细，后期系统的可维护性就更容易，可扩展性更高[11]。对于 SaaS 租户，本系统的权限控制方式为租户各部门组织对被授权数据资源的操作提供良好的约束。

基于组织架构的数据权限控制是将数据按照组织架构进行划分的一种方式，相比按照层级的权限划分，具有更好的灵活性和更精细的管理粒度[12-13]。

2 关键定义

以下定义限定在单个 SaaS 租户使用的系统服务中。

资源(resource)：软件系统提供的各类服务，包括前端页面、控制器等，具体资源以及它们按照功能聚集形成的集合。在数据结构上，资源为树状结构。

数据受控资源(data controlled resources)：分配时需要对访问数据加以限制的资源。数据受控资源的父资源一定是数据受控资源[14]。

数据不受控资源(data uncontrolled resources)：分配时，不需要对访问数据加以限制的资源。

受控数据(controlled data)：访问权限需根据权限分配加以控制的数据。

功能级资源(functional resources)：完成需求中某一特定功能的具体资源或其集合。此类资源在权限分配时用户系统管理员可见，且可以理解。

业务级资源(business resources)：除功能级需求外，系统内页面、控制器等用户不关心的具体资源。实现中为资源树的叶节点。此类资源在权限分配时用户系统管理员不可见，且用户系统管理员难以理解[15]。

用户级资源(user resources)：数据不受控的功能级资源。

数据级资源(data resources)：数据受控的功能级资源。

资源分类之间的关系如表 1 所列。

表 1 资源关系

Table 1 Resources relationship

	Functional resources	Business resources
Data controlled resources	Data resources	—
Data uncontrolled resources	User resources	—

资源树(resources tree)：系统内各类资源组成的树状结构。系统资源树结构如图 1 所示。

组织架构树(organization tree)：业务领域内，用户按照管理能级构成的树状结构。组织架构树如图 2 所示。

组织架构与成员树(organization and member tree)：规定系统中的业务用户必须隶属于组织架构中的某一部门(节点)，基于这一关系构成的树状机构成为组织架构与成员树。组织架构与成员树如图 3 所示。

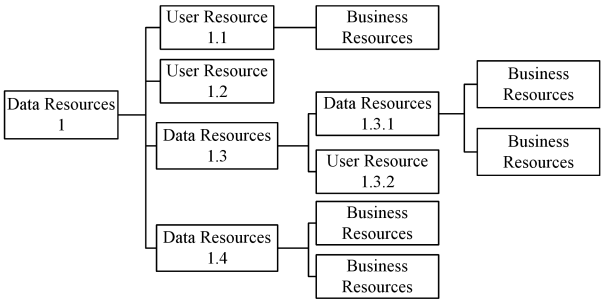

图 1 系统资源树

Fig. 1 Resources tree

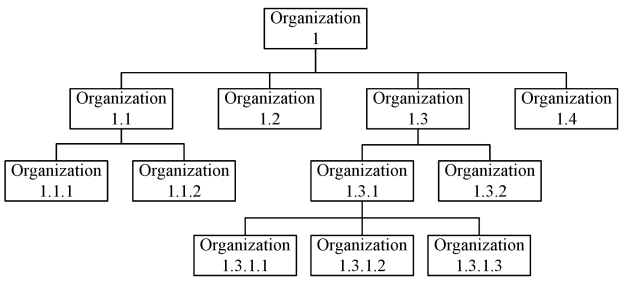

图 2 组织架构树

Fig. 2 Organization tree

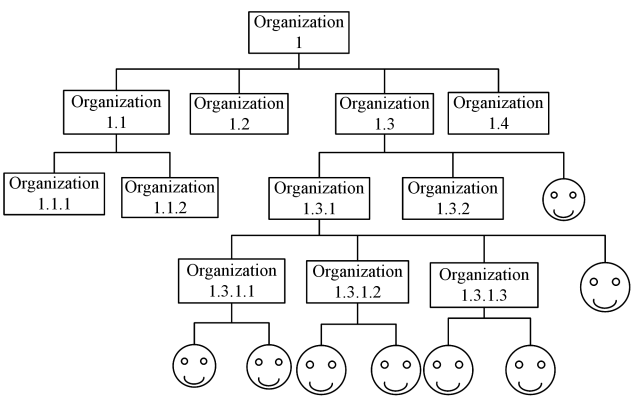

图 3 组织架构与成员树

Fig. 3 Organization and member tree

数据创建者(creator)：数据中某一记录由用户创建(而非系统产生)时，创建记录的用户是该数据的创建者。

数据归属(access_group)：在基于组织架构的数据权限控制中，为使数据可以组织架构划分，规定所有需要被控制的数据记录归属于业务用户或组织架构中的部门。即需要控制的数据表中，每条记录对应于组织架构与成员树中的一个或多个节点。

资源分配：用户系统管理员将资源分配给角色的交互过程。

角色分配：用户系统管理员将角色分配给系统用户的交互过程。

3 ODAC 模型

3.1 数据归属——数据到组织架构与成员树的映射

系统中数据对象 d_1，d_2…，d_m 按其数据类型 D_1，D_2，…，D_n 构成数据集合 D(通常在数据库中，不同类型的数据存储在不同的表中，每个对象保存为一条记录)，组织架构与成员树构成集合 O，数据映射如图 4 所示。

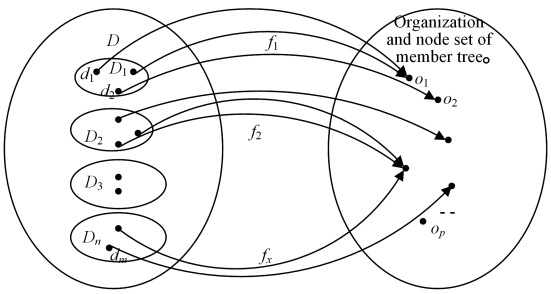

图 4 数据映射

Fig. 4 Data mapping

系统中，部分数据类型需要按照组织架构划分访问权限。为使此类数据可依据组织架构划分，需构建数据集合中受控数据类型的对象到组织架构与成员集合对象的映射 $f_N: D_N \rightarrow O$。映射法则 f_N 取决于不同数据对象对应的业务逻辑。即将系统中需要控制数据访问权限的数据对象，与业务领域的部门或员工建立对应关系，以便按照部门加以分配。

3.2 资源与数据的交叉控制

定义某一角色权限时需要对其可使用的系统资源和数据加以限定，对于不同的系统资源，应当可以独立地制定可以操作的数据集合。

如图 5 所示，需要对数据访问进行控制的系统资源集合 R，包含资源 $r_1, r_2, \cdots, r_m$。系统组织架构与成员集合为 O，包含资源 $o_1, o_2, \cdots, o_n$。系统中的资源与数据构成平面：

$$R \times O = \{\langle r, o\rangle \mid r \in R \wedge o \in O\}$$
$$= \{\langle r_1, o_1\rangle, \langle r_1, o_2\rangle, \cdots, \langle r_1, o_n\rangle, \langle r_2, o_1\rangle, \langle r_2, o_2\rangle, \cdots, \langle r_m, o_n\rangle\}$$

其中，向量 $\langle r, o\rangle$ 标示允许通过 r 资源访问数据集合 D_o，D_o 是组织架构与成员节点 o 在数据到组织架构与成员树的映射 f_N 的原像集合。

对某一角色定义其权限，即赋予其集合 $A \subset R \times O$。

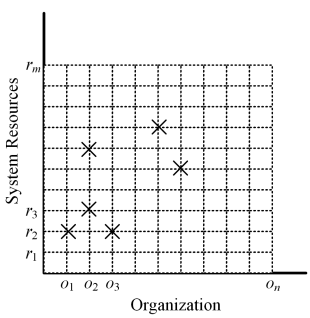

图 5 交叉控制

Fig. 5 Cross control

3.3 基于组织架构的数据权限控制

使用上述控制策略，在角色定义时操作较为复杂。为简化权限定义过程，此处对权限控制策略加以部分限制。

规定 1 在赋予某角色某一特定资源 r_x 的操作权限时，当指定其数据访问权限为组织架构与成员树节点 o_x 时，即对其授权 $\langle r_x, o_x\rangle$ 时，向其数据节点访问集合增加 $\{\langle r_x, o_z\rangle \mid o_z \in o_x$ 在组织架构与成员树中的子树节点集合$\}$ 中的全部元素。即在某一特定资源下，制定角色可以操作上级部门的数据，同时允许其操作该部门全部下辖部门及员工的数据。

规定 2 由于在实际业务场景中，不存在向某一用户指定操作归属于特定用户的数据的需求。规定定义权限时，除可单独定义对自己拥有数据的访问权限（下文简称所有者权限，Owner Authority）外，不允许定义权限限定于某一员工的数据。

规定 3 鉴于实际需求，规定可以授予用户一种特殊的权限：允许数据创建者访问该数据，而不受限于数据归属的权限（下文简称创建者权限，Creator Authority）。

4 分析与实现

针对前文提出的 ODAC 模型，以下采用 Spring MVC 框架、Spring Security 框架和 MyBatis 框架实现对数据权限的控制。

4.1 数据模型

在数据模型设计上，依据对数据权限控制的需求，建立了用于完成认证授权的相关对象，为受控数据增加了权限控制的相关信息。

认证授权概念数据模型（Conceptual Data Model，CDM）如图 6 所示。实体 authorization 为认证实体，包含认证方式、认证信息（如用户名密码、openId 等）以及其他相关记录；实体 user 为系统用户，每个用户可与多个认证实体关联，即拥有多种认证方式；实体 role 为角色，角色与用户为多对多关系（物理数据模型应增加关系表）；实体 organization_structure 为组织架构，实体 resource 为资源，两实体对象均通过记录父节点构成树状结构。

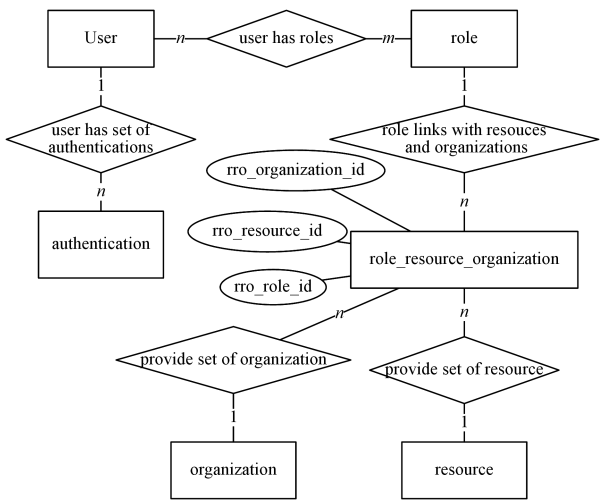

图 6 认证授权概念数据模型

Fig. 6 CDM of authentication and authorization

受控数据的物理数据模型（Physical Data Model，PDM）如表 2 所列。对于受控数据，需要在存储时增加权限控制相关字段以存储数据归属信息。

表 2 受控数据的物理数据模型

Table 2 PDM of controlled data

Field name	Data type
business fields…	…
creator	integer
access_group	varchar

4.2 权限控制流程

数据权限控制的基本流程如图 7 所示。在 Spring Security 完成认证后，在用户 session 中为用户维护对象 Date Au-

thorization Register，负责记录当前请求允许访问的组织架构集合。

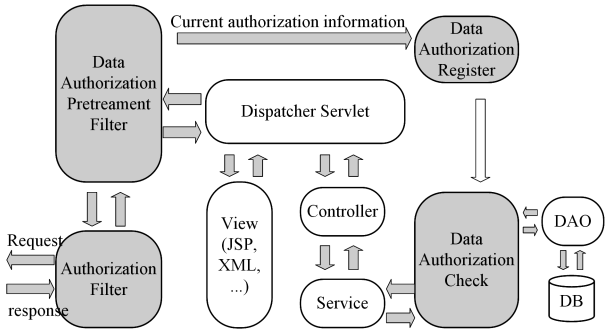

图 7 数据权限控制流程

Fig. 7 Data authority control process

认证完成后用户请求到达服务器端后的处理流程是：

1）通过 Spring Security 授权过滤器，裁决当前用户是否可以访问当前资源。若权限不足，则禁止其访问请求的资源。

2）通过数据授权预处理过滤器，匹配当前资源与角色允许访问的组织架构与成员集合。将获得的数据权限信息在对象 Date Authorization Register 中进行登记。

3）数据库读写前进行数据权限控制，若当前操作为新建数据，则依据当前业务需求与用户信息，完善数据权限控制相关信息，并将其写入数据库；若当前操作为读取、删除、修改操作，则对当前操作数据、允许访问的组织架构与成员节点进行匹配，裁决是否有权访问。

此数据访问控制流程具有以下特点：

1）数据权限控制与具体业务相关，需要根据不同类型的数据增加不同的裁决逻辑。但增加裁决逻辑与业务逻辑分离，不影响原有业务逻辑。

2）数据权限是基于资源授予的，即一次请求具有相同的数据权限，控制器调用多个业务层对象的方法，业务层多次访问数据库，都具有一致的数据权限。

3）由于 Spring 容器内的 Bean 是没有状态的，在裁决数据权限时，仅能依据当前线程判断此次数据访问来源于哪一请求。

4.3 授权模块实现

4.3.1 系统初始化

为提高系统效率，在初始化时需将部分数据由系统一次性加载至 Web 服务器。加载表包括 resource 表、organization_structure 表、role_resource_orgnazation 表，各表对应的对象为 ResourceTree、OrganizationStructureTree、RoleResource-Orgnazation。将上述数据加载至 Web 服务器主要是基于以下考虑：

（1）数据访问频繁。安全控制中每一个用户请求都需要对上述数据进行访问。

（2）数据量较小，系统交付后 resource 表不再增长，organization_structure 表规模基本稳定。

4.3.2 数据授权

数据授权划分为两个阶段，第一阶段为数据预授权阶段，用于获取并保持当前访问资源的可访问组织架构与成员节点集合，第二阶段为数据访问控制阶段，用于决断待操作的数据是否与可操作的数据匹配。

（1）数据权限登记

数据权限记录通过 DataAuthorityRegister 类的对象进行登记保持。DataAuthorityRegister 的结构如图 8 所示。

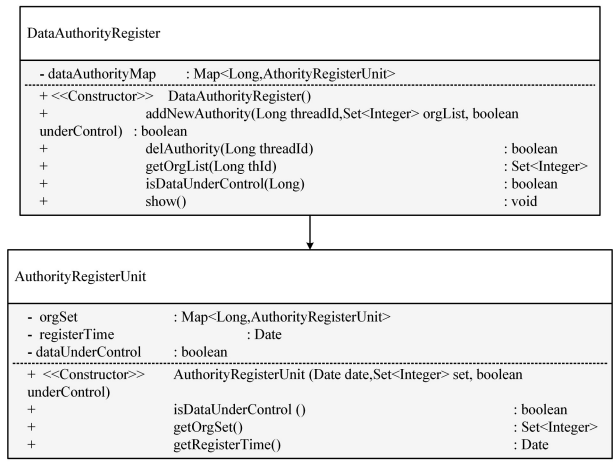

图 8 数据授权登记类图

Fig. 8 Class diagram of data authorization registration

AuthorityRegisterUnit 类用于保存一次资源访问的权限集合，DataAuthorityRegister 用于保存当前用户的授权信息。

DataAuthorityRegister 以当前线程 Id 为键值，以散列表形式存储单次访问的 AuthorityR-egisterUnit，并对其进行动态维护，即时删除结束访问资源的授权信息和过期授权。

（2）数据预授权阶段

数据预授权阶段的流程如图 9 所示。

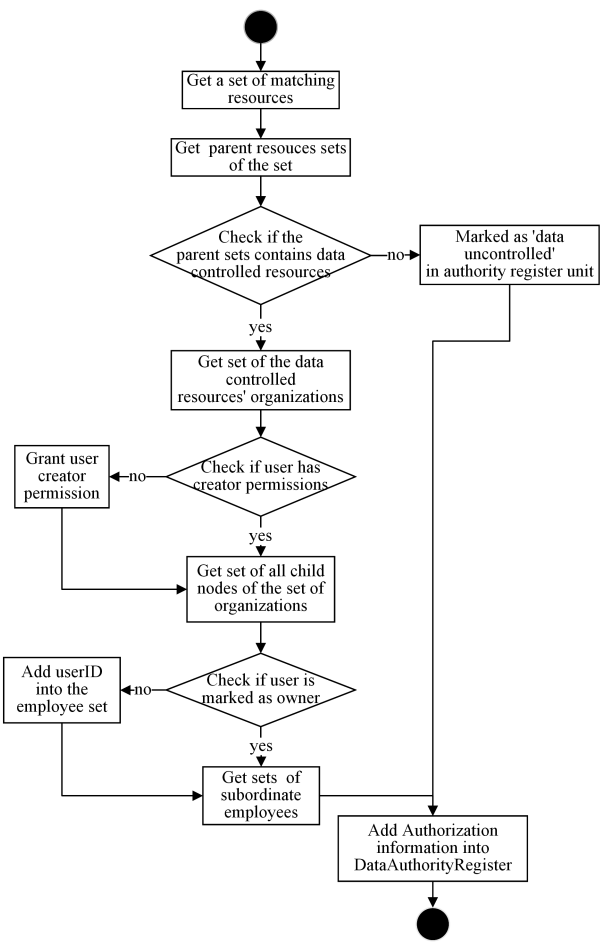

图 9 数据预授权流程

Fig. 9 Data pre authorization process

(3)数据访问控制阶段

数据访问控制阶段主要完成对访问数据是否归属于预授部门集合或员工集合的裁决。控制逻辑由具体的业务逻辑决定。MyBatis 的实现方式为:创建继承于 Mapper 的接口 MapperEx,并创建 MapperImpl 类实现 MapperEx 接口。MapperImpl 类在实现接口方法时,对插入操作添加数据授权相关字段信息,然后调用 Mapper 接口插入方法来完成;其他方法先对权限进行检验,然后调用 Mapp-er 接口的方法来完成数据操作。

结束语 本文对软件系统安全展开了讨论,在对现有数据授权或访问控制方法进行分析的基础上,提出了一种基于组织架构的数据权限控制模型,并基于 Spring MVC, Spring Security 和 MyBatis 框架来实现该模型。该模型的实现方法已在多种实际生产系统中使用,具有较好的通用性和可行性。ODAC 模型对数据权限控制粒度属于行级别,不能实现对数据的列(字段)级的控制,后续将在此基础上进一步开展研究工作,实现对列级别的数据访问控制。

参 考 文 献

[1] 赵静,杨蕊,姜滦生. Web 信息系统中的资源访问控制[J]. 计算机工程与设计,2010,31(15):3353-3389.

[2] 林伟炬,刘列根,张宇. 一个通用的权限管理模型的设计方案[J]. 微计算机信息,2009,22(15):1-3.

[3] NAZERIAN F,MOTAMENI H,NEMATZADEH H. Emergency role-based access control (E-RBAC) and analysis of model specifications with alloy[J]. Journal of Information Security and Applications,2019,45:131-142.

[4] GHAFOORIAN M,ABBASINEZHAD-MOOD D,SHAKERI H. A thorough trust and reputation based RBAC model for secure data storage in the cloud[J]. IEEE Transactions on Parallel and Distributed Systems,2018,30(4):778-788.

[5] JIN X,KRISHNAN R,SANDHUR. A unified attribute-based access control model covering DAC,MAC and RBAC[C]// IFIP Annual Conference on Data and Applications Security and Privacy. Springer,Berlin,Heidelberg,2012:41-55.

[6] MUDDIN M,ISLAM S,AL-NEMRAT A. A dynamic access control model using authorising workflow and task-role-based access control[J]. IEEE Access,2019,7:166676-166689.

[7] QIANG Z,DONG C. Enhance the user data privacy for SAAS by separation of data[C]// 2009 International Conference on Information Management,Innovation Management and Industrial Engineering. IEEE,2009,3:130-132.

[8] TIWARI P K,JOSHI S. Data security for software as a service [M]//Web-based services:Concepts,methodologies,tools,and applications. IGI Global,2016:864-880.

[9] JOHA A,JANSSEN M. Design choices underlying the software as a service (SaaS) business model from the user perspective:Exploring the fourth wave of outsourcing[J]. Journal of Universal Computer Science,2012,18(11).

[10] TSAI W T,ZHONG P. Multi-tenancy and sub-tenancy architecture in software-as-a-service (SaaS)[C]// 2014 IEEE 8th International Symposium on Service Oriented System Engineering. IEEE,2014:128-139.

[11] LOMOTEY R K,DETERS R. SaaS authentication middleware for mobile consumers of iaas cloud[C]// 2013 IEEE Ninth World Congress on Services. IEEE,2013:448-455.

[12] BELIM S V,BOGACHENKO N F,KABANOV A N. Severity Level of Permissions in Role-Based Access Control[C]// 2018 Dynamics of Systems,Mechanisms and Machines (Dynamics). IEEE,2018:1-5.

[13] PERMANA R I,SUROSO J S. Data Governance Maturity Assessment at PT. XYZ. Case Study:Data Management Division [C]// 2018 International Conference on Information Management and Technology (ICIMTech). IEEE,2018:15-20.

[14] FERRISJ M. Providing access control to user-controlled resources in a cloud computing environment:U. S. Patent 8,984,505[P]. 2015-3-17.

[15] THOMPSON W J J,VAN DER WALT J S. Business intelligence in the cloud[J]. South African Journal of Information Management,2010,12(1):1-15.

CHENG Xue-lin, born in 1976, Ph. D, senior engineer, master supervisor, is a member of China Computer Federation. His main research interests include data mining and analysis, software engineering.

ZHUO Chong-kui, born in 1996, postgraduate. His main research intersts include software engineering and data analysis.