International Journal of Security and Its Applications
Vol.7, No.6 (2013), pp.67-78
http://dx.doi.org/10.14257/ijsia.2013.7.6.07

Data Security Monitoring Platform in Cloud for Enterprise

Geng Yushui and Pang Shunpeng

School of Information, Qi Lu University of Technology
Jinan250353, China
gys@qlu.edu.cn, pangshunpeng@163.com

Abstract

Now most software systems in cloud platform use Multi - Tenancy architecture. A single software system serves for multiple client organization. All customers' data will not be stored in only one node. So the system needs higher data security mechanism. This paper wants to build a data security monitoring model in cloud platform for large enterprises. The model can set the authentication, logging, fine-grained access control, dynamic data filtering strategy and data audit to realize the security protection for enterprises data. The model proposed by this paper uses multi-tenancy SaaS(Software as Service) application architecture, RBAC (Role-Based policies Access Control) model and operation in the context of environmental perception to realize the data access control. It using PMI framework to provide accession management services for enterprise users.

Keywords: *Cloud computing, Data security, RBAC, SaaS, PMI*

1. Introduction

With the depth and width of information construction progressing, information grow into index explosively in information system. Filtering of unwanted data, protecting enterprise existed data and only showing information concerned with the current scene are great necessity. Data safety monitoring is an effective way to meet the above requirements. Not only does it facilitate quick access to personalized information, but it is an important means to ensure the security of data. The technical requirements in the application of emerging SaaS (Software as Service) are more evident in the application. SaaS is a new model of software application adopting online rental service through the network. Due to its unique single-instance and multi-tenant properties, SaaS put forward higher requirements for date security.

2. Cloud Computing

Cloud computing combines a large number of computing resources, storage resources and software resources together. With forms a huge shared virtual IT resource pool and it offers a variety of IT services for remote computer users. In the IT industry, Cloud computing is generally considered to be an important growth point since the Internet economic prosperity which has a huge growing prospect market.

Nevertheless, there are still many companies choosing the traditional software architecture largely of the reasons is most likely that enterprise data security issues are unresolved in cloud computing. Some analysis of the survey results show that data security is one of the biggest obstacles to migrate enterprise applications to the cloud computing. At present, cloud computing security issues have been gotten more and more attention. From the bottom to the

ISSN: 1738-9976 IJSIA

top security issues in the cloud computing environment can be summarized as physical security, network, storage, server security, data security, identity and access security. The paper is only concerned about the security of the application logic, namely the identity and access security, and data security.

In a cloud computing environment, software application modes are based on the SaaS model. SaaS model is a single-instance and multi-user architecture. Mature SaaS application should have three characteristics that are extensibility,multi-user efficiency and configurability. Extensibility is referred, allowing for more function can be increased in the original design when necessary, or to obtain better performance on the basis of extending hardware conditions. Multi-user efficiency requires that SaaS architecture is not only able to maximize the sharing of resources between different users, but also can distinguish data belonging to different customers. Configurability means that in a single application instance serves multiple clients case, each user can configure the respective application appearance and behavior using the metadata [1].

In terms of the SaaS application model and application scenarios of the software in the cloud environment, based on basic authentication and authorization functions, the data security in the SaaS must also have highly configurability. According to the different size of the security strategy of resources, it can achieve precise control of data and operation. Based on the above discussion, the paper provides a monitoring model of enterprise data security in the cloud environment, implementing authentication, authorization, fine-grained access control, dynamic data filtering policies, data auditing and other functions, and the mode is applied in the mode of SaaS application.

3. Critical Idea and Technology

3.1. Principles that Need by the Research

1. The principle of openness. Platform system achieved in this project is the second development of software platforms. For enterprise developing SaaS application it requires that the interface must be provided with the outside world, to achieve organic integration with other applications.

2. The principle of structured, hierarchical, modular. Using the object-oriented technology, makes the system highly structured, modular, hierarchical. The whole system is defined by many modules which have good interfaces. Each module has a detailed functional description and design presentation. Each module completes relatively independent function. Interfaces between the modules defined regularly, which makes the changes of module features relative independent, not affecting the features and structure of the whole system and it is easily for the system’s upgrade and maintenance.

3. Excellent portability. Selecting of server software system that supports a variety of operating platforms, such as database server, application server, WEB server. Choosing choice of middleware system is that is a development language with good portability developed and applied by B/S. These can improve the platform portability of the application systems.

3.2. Technical Directions

1. Choice of development tools. As Java inherent with cross-platform, safety, strong network function, enterprise’s solution based on Java has become a fact of current standard. So the selection of development tools is Java.

The overall structural design of the software platform is built on the J2EE platform specification. Many enterprise solutions including the development, deployment and management of the complex issues can be simplified by using the Java 2 technology. It has incomparable advantages compared with the traditional model of Internet applications.

2. Database system selection. The selection of database system is that supports business application systems commonly used in large-scale relational database system MS SQL SERVER and ORACLE.

3. The SaaS overall architecture based on metadata configuration. The SaaS overall architecture based on metadata configuration is shown in Figure 1.

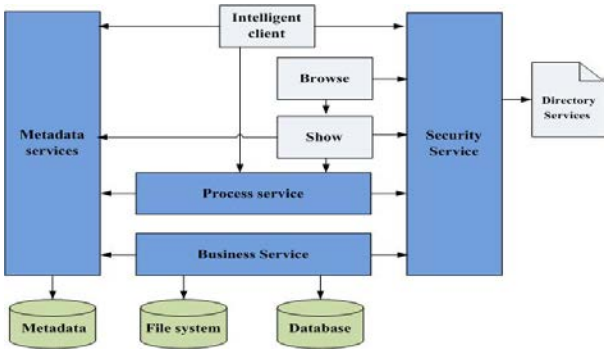


Figure 1. The SaaS Overall Architecture based on Metadata Configuration

Thereinto Process Services give the Smart Client and network interface the supply layer can call and start the synchronization process, or start consuming more of the transaction, to call other Business Services, and throughout the data storage business to interact in order to read and write data.

4. The Basic Structure of the Data Safety Monitoring Model in Cloud Environment

We want to build a large enterprise-level application, monitoring platform for data security in general. In the logic layer of the application, we are to legalize the implements software SaaS application modes of authentication, authorization, fine-grained permissions control, dynamic data filtering strategy, data auditing, and other functional requirements, which are applied to SaaS software system security control areas [9].

4.1. Logical Structure

General logical schema of data safety monitoring model is shown in Figure 2.

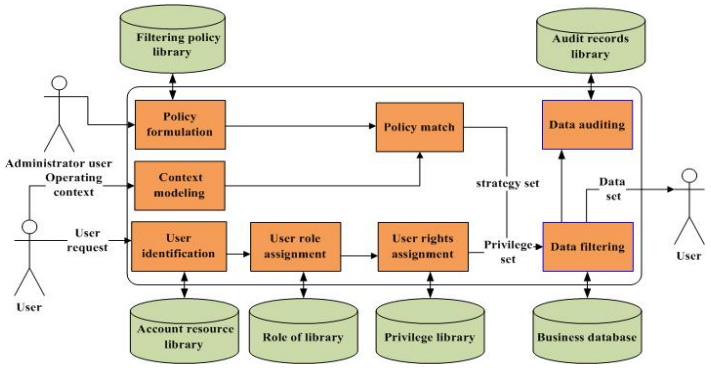


Figure 2. Overall Logic Diagram of Enterprise Data Security Monitoring Platform in Cloud Environment

The data safety monitoring platform is divided into four sub-systems which includes RBAC-based rights management system, Context-based policy management system, Data-filtering system and Data-auditing system.

Enterprise data safety monitoring platform that based on RBAC model achieved Role-based access control. PMI (Privilege Management Infrastructure) build rights management services for enterprise-level users to achieve the functions as user authentication, authorization main management, authorization object management, and role-based authorization management [2, 7]. Above the RBAC permissions management system, context-based policy management system for data safety monitoring platform provide flexible and dynamic data filtering policy management and fine-grained data control functions. When users access data, the system captures and matches information of operating context in the current scene to realize further filtering and fine-grained control of user data.

In order to generate the end-user data collection of operations, user-specified data set is influenced by permission set of rights management system based on RBAC and policy set of context-based policy management system.

4.2. Functional Structure

The functional structure of the enterprise data safety monitoring platform in the cloud environment is shown in Figure 3.

Main features are described below:

1. User authentication service. The platform uses the identity mapping database to legalize the user identity authentication service, It can correspond the user's local account to the account of the resources of the host, allow one-to-one, and (also) many-to-one, which increases the flexibility of the platform configuration. The administrator can correspond the users with same access requirements to the same account, provided that the minimum required permissions to the account permissions configured for this group of users.

2. Authority and Subject management. Authorized subject of the platform mainly includes user, role, organization, position, *etc*. It realized the management of the subject as well as the inheritance and transmission of the permission, by establishing a relationship between user and role, organization and position, Led platform to the tree structure of the organization, allows the user demand to add the authorization body as the basis for personnel management. It uses organization tree as a personnel management in a centralized manner, and supplied by post further positioning, which makes it easier to realize access control efficiently.

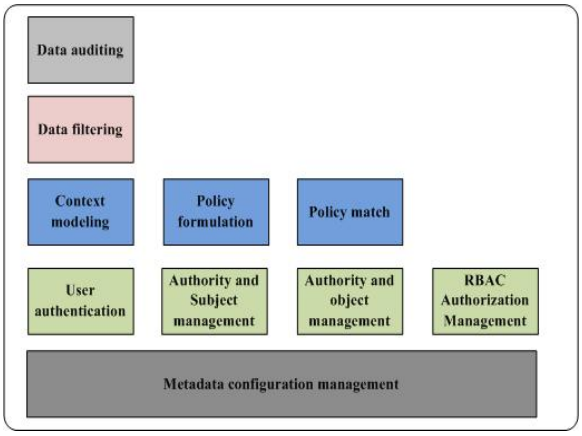


Figure 3. The Functional Structure of the Enterprise Data Safety Monitoring Platform in Cloud Environment

3. Authority and object management. Authorized object refers to all kinds of accessible resources. Platform provided a unified interface for the management and achieved the hierarchical management of the resources, and set up the concept of the resource type and resources set to support the centralized management and operation of multiple resources. Resource type and resources set to resources are many-to-many relationship. On that basis, the definition of the concept of resource operations is for the mutexes that may arise in the process of access to resources, dependence constraint management.

4. RBAC Authorization Management. The platform provides support for distributed RBAC model and on the basis of the original role, it extend the model. By defining the role of inheritance and inclusion relationship, it reduces duplicate management authorization and Improves performance issue of the RBAC model under mass access control environment. The platform defines inclusive, exclusive, inheritance, dependence, the compatibility constraint types for the role, and configures the corresponding set of constraint types at all levels to ensure independent role permissions to avoid permission conflicts from happening.

5. Context modeling. In the process of system designing modeling system perceives the context properties perceivable by the system are modeled, which are be called in the process of policy making process modeling context properties. Then the scene values for those properties delimit, to create an appropriate context judgment, as a basis for policy implementation, which can be regarded, so that for policy modules, based on user action scenes in the actual context of realization of dynamic data filtering [8].

6. Policy formulation. In the policy making process, the system provides a policy template file for the user functioned as the policy rules. Filtering policy of the system is based on XACML syntax expansion, and is stored to the XML file. The filtering rules use the dynamic SQL form, in order to filter easily and efficiently filtering in the database side, so as to improve the performance of data filtering [5][7].

7. Policy match. The process of policy match needs to match context value with the context found in the policy set one by one in order to filter out of the filter rules which can be applied in the current scene, and be placed in a filtered space of the current operation, so that the rules can be invoked in the data filter.

8. Data filtering. The system expresses the filtering rules in the form of dynamic SQL, in

phase of filter rules implementation. In the user action scene, it dynamically generates SQL statements with the appropriate filter rules in the user's tacit data set, to filter data control implement dynamic data [3].

9. Data audit. The goal of data audit system is to deploy audit for data needed to auditing. When the data audited is performed, it automatically records the information of the operator, the operation time, the operation target, and the operating behavior, and then provides query and statistical functions of information. What's more, according to data of different security needs, data audit is divided into two audit degrees: recording operation data and non-recording operation data [4].

10. Metadata configuration management. Metadata configuration management is the basis of data security platform. It is main responsible for managing application configuration for different users. It is achieved by providing users with a range of configuration options and features switch, and it is stored in the form of metadata.

5. Solutions of Data Security Monitoring Platform for Enterprise based on SaaS

Solutions of Data security monitoring platform for enterprise based on SaaS includes four Algorithms:

1. RBAC based on artificial neural network.

2. Context-aware modeling based on ontology.

3. Multi-attribute data filtering based on rough se.

4. Multi-dimensional association rule mining algorithm for audit analysis.

5.1.RBAC based on Artificial Neural Network

It uses two layers of cascading algorithm to forward and send them back. It is a supervised learning algorithm on acyclic multistage network. Before system run, through analyzing the relationship between role and its corresponding permission, as well as the relationship between different roles, especially the role permissions inheritance relationship between higher and lower to select appropriate role/permission vector as the training samples of artificial neural network input/output. In training, the role samples as the input vector and the corresponding privilege as the perfect output vector. Input vector and its corresponding output vector form training pairs, and the input and output layers use logistic function as the activation function to train the corresponding artificial neural networks.

Step1. First, user login the system by authentication. According to the information of identity, system resolves the information of its corresponding roles to get role code.

Step2. When user operates the system, it takes the user code as an input vector for artificial neural network to directly calculate corresponding output vector, which user has permissions.

Step3. According to user permissions, allow or deny a user's current system operation.

Step4. Algorithm end

5.2. Context-aware Modeling Based on Ontology

The main idea is to convert concept and relationship of context domain ontology into a collection (ABox) of description logic. The concept and relationship of context ontology is stored in the axiom set (TBox). This consists of two parts:

The concept and relationship of context meta-ontology, inheritance and expansion of domain ontology for meta-ontology. On that basis, so we express the rules set of context ontology by using Semantic Web Rule Language (SWRL).

1. The algorithm is proposed to convert the context model

Step1. Generate TBox

Step1.1. It creates corresponding concepts for kinds of meta-concepts which appear in meta-ontology model in TBox.

Step1.2. Correlation R between concept of A and B for context meta-ontology is created in TBox first, and then axiom $A \subseteq R.B$ is added.

Step1.3. About the inheritance between super-concepts and sub-concepts for context meta-ontology, axiom $B \subseteq A$ should be added in TBox.

Step1.4. For the property X (data type is T) of concept A, axiom $A \subseteq X.T$ should be added in TBox.

Step2. Generate ABox.

Step2.1. It creates corresponding examples for kinds of domain concepts which appear in context meta-ontology in ABox.

Step2.2. For various relationships of context domain ontology, first we map them as exiting in TBox, and then we add them to corresponding relationships in ABox.

Step2.3. For various properties of context domain ontology, corresponding property instance is created in ABox.

Step3. Algorithm end.

2. The algorithm is proposed to convert the SWRL

Input: Rules collection of ontology include Meta-Rules of meta-ontology and specific Dom-Rules of domain.

Output: SWRL rules

Step1. Convert precondition of rules into antecedent of SWRL.

Step1.1. The association R between X and Y is converted to antecedent clause of SMRL in precondition, as the same time relation R is converted to $R(x, y)$. x and y can be variable or constant. They are the instance of X and Y.

Step1.2. If the antecedent contains multiple clauses, and then join relationship between them.

Step1.3. Add a predicate like $A(x, a)$, where instance x of concept X is variable and value of property a is constant, meanwhile A is property predicate.

Step2. Convert results of rules into consequent of SWRL.

Step3. Algorithm end.

5.3. Multi-attribute Data Filtering based on Rough Set

Set a condition attribute $c = \{a_0, a_1, \mathrm{L}\ , \mathrm{a}_n\}$

Step1. Given a value of m, m<n;

Step2.The condition attribute set is divided into several subsets which they are disjoint. $Q_0, Q_1, \mathrm{L}\ , Q_S, Q_0 = \{a_0, a_1, \cdots, a_{m-1}\}\ , Q_1 = \{a_m, a_{m+1}, \cdots, a_{2m-1}\}, \cdots, Q_s = \{a_{sm}, a_{sm+1}, \cdots, a_n\}$

Step3. For each $Q \in \{Q_0, Q_1, \cdots, Q_S\}$ to the following:

1. Different value of Q are represented by $q_0, q_1, \cdots, q_{k-1}$.

2. $q_0 = 0, q_1 = 1, \cdots, q_{k-1} = k-1$;

3. $j = 0$;

4. $v = q_j + 1$, $q_{j+1} = q_j$,if there is a $Y_{i0} \in U / IND(D)$ such that $\lfloor q_j \rfloor Q$ and $\lfloor q_{j+1} \rfloor Q$ belong to it, then $q_{j+2} = q_{j+2} - 1, \cdots, q_{k-1} = q_{k-1} - 1$,or $q_{j+1} = v$

5. $j = j + 1$, if $j = k$,then output $q_0, q_1, \cdots, q_{k-1}$, or go to 4.

Multi-attribute filtering can reduce the merging contingency. Here, we get a training set by an information table. Assuming when condition attribute is a=0, decision attribute is d=0, so $(\mathrm{a=0}) \Rightarrow (\mathrm{d=0})$. When numbers of training examples are given insufficient, this rule has a larger contingency. If we add a condition attribute b, we can get d=0 when a=0, b=1, that is $(\mathrm{a=0}) \wedge (\mathrm{b=1}) \Rightarrow (\mathrm{d=0})$.

5.4. Multi-dimensional Association Rule Mining Algorithm for Audit Analysis

The whole algorithm is divided into two steps:

First, the generation of frequent item sets. (This step need minimum support threshold).

Second, to generate strong association rules by the frequent item sets. (This step need minimum confidence threshold).

1. The generation of frequent item sets.

Algorithm character based on Apriori: A set of any non-empty subsets of frequent item sets also must be a frequent item sets, that is, if the model whose length is k is frequent, so the length of k+1 could not be frequent.

Input: audit trail database D and minimum support threshold.

Output: frequent item sets L in D.

Step1. L_1 = {find frequent item sets 1-item sets}

Step2. $\mathrm{for(k=2; L_{k-1} \neq \Phi; k++)}$ do

begin

$C_k = \mathrm{apriori\_gen(L_{k-1,}\ min\_sup)}$

For all candidate

$c \in C_k$ *do*

$L_k = \{c \in C_k \mid c \ \ count \geq \min\_\sup\}$

End

Step3. *return* $L=U_K L_K$;

The function of Apriori_gen includes two options: self-joins and pruning.

Input: L_{k-1} and minimum support threshold

Output: super group of all frequent k – item sets

Step1. insert into C_k

Step2. *select* $p[1], p[2], \cdots, p[k-1], q[k-1]$

from $L_{k-1}\ p, L_{k-1}\ q$

where $p[1]=q[1], \cdots, p[k-2]=q[k-2], p[k-1]<q[k-1]$

Step3. candidate set for each C_k do

$f(s \notin L_{k-1})$ *then*

Delete c from C_k

2. Multi-dimensional association rule generating algorithm for audit analysis.

Input: frequent item sets L from algorithm 1.

Output: Multi-dimensional association rule.

For each frequent item sets L do

Find all of non-empty subsets from L.

For each non-empty subset subL do

if $(support\_count(L) / support\_count(subL) \geq \min\_conf)$

then output multi-dimensional association rule.

" $subL \Rightarrow (L - subL)$ "

6. Algorithm Implementation

The overall structural design of the software platform is built on the J2EE platform specification. The selection of database system is that supports business application systems commonly used in large-scale relational database system MS SQL SERVER and ORACLE. Algorithm 1to 4 of this scheme mainly involve to realize role-based access control, context modeling, data filtering and data audit. As the importance of the role-based access control, as well as the combination with others, here we mainly introduce the implementation process of RBAC. Role relationships are shown in Figure 4:

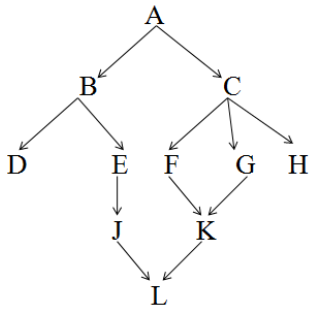


Figure 4. Role Relations

Table 1. Code Rule of User and Role

role	A	B	C	D	E	F	G	H	I	J	K	L
R	11	10	9	8	7	6	5	4	3	2	1	0
B	0	0	0	0	0	0	0	0	1	1	0	0

Superior role inherits all permissions of subordinate role. Each role privilege is shown in Table 2. Role and others combine as input vector in table, as well as the corresponding permission as output vector. After artificial neural network completes training, we have 6 users with different roles.

$u_1 : \{I, J\}$ (000000001100)

$u_2 : \{G\}$ (000000100000)

$u_3 : \{B, F\}$ (000001000000)

$u_4 : \{E, H\}$ (000010010000)

$u_5 : \{E, G, I\}$ (000010101000)

$u_6 : \{A\}$ (100000000000)

Table 2. Training Sample of Artificial Neural Network

Input(role)		Output (permission)
role	role code	P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11} P_{12} P_{13} P_{14}
A	1 0 0 0 0 0 0 0 0 0 0 0	0 1 1 1 1 1 1 1 1 1 1 1 1 1
B	0 1 0 0 0 0 0 0 0 0 0 0	1 1 0 1 1 1 0 0 0 1 0 0 1 0
C	0 0 1 0 0 0 0 0 0 0 0 0	0 0 1 0 1 0 1 1 1 0 1 0 0 1
D	0 0 0 1 0 0 0 0 0 0 0 0	1 0 0 1 0 0 0 0 0 0 0 0 1 0
E	0 0 0 0 1 0 0 0 0 0 0 0	0 1 0 0 1 1 0 0 0 0 0 0 0 0
F	0 0 0 0 0 1 0 0 0 0 0 0	0 0 1 0 1 0 1 0 0 0 0 0 0 1
G	0 0 0 0 0 0 1 0 0 0 0 0	0 0 1 0 1 0 0 1 0 0 0 0 0 1
H	0 0 0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 1 0 0 0 0 0
I	0 0 0 0 0 0 0 0 1 0 0 0	1 0 0 0 0 0 0 0 0 0 0 0 0 0
J	0 0 0 0 0 0 0 0 0 1 0 0	0 1 0 0 1 0 0 0 0 0 0 0 0 0
K	0 0 0 0 0 0 0 0 0 0 1 0	0 0 1 0 1 0 0 0 0 0 0 0 0 1
L	0 0 0 0 0 0 0 0 0 0 0 1	0 0 0 0 1 0 0 0 0 0 0 0 0 0
BK	0 1 0 0 0 0 0 0 0 0 1 0	1 1 1 1 1 1 0 0 0 1 0 0 1 1
CD	0 0 1 1 0 0 0 0 0 0 0 0	1 0 1 1 1 0 1 1 1 0 1 0 1 1

Six role codes of users as the input vector for neural network to train corresponding output vector, at the same time the training accuracy control in 10^{-6}.The output values we get are consistent with expected output permissions in Table 3, so this algorithm is effective. Although we only verify 6 users, but the algorithm can be extended to any number of users based on specific information system. This complies with the multi-user extensibility of SaaS.

Table 3. User Role and Output Permission

Input(role)		Output (permission)
role	role code	p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} p_{12} p_{13} p_{14}
U	0 0 0 0 0 0 0 0 1 1 0 0	1 1 0 0 1 0 0 0 0 0 0 0 0 0
G	0 0 0 0 0 0 0 1 0 0 0 0 0	0 0 1 0 1 0 0 1 0 0 0 0 0 1
BF	0 1 0 0 0 1 0 0 0 0 0 0	1 1 1 1 1 1 1 0 0 1 0 0 1 1
EH	0 0 0 0 1 0 0 1 0 0 0 0	0 1 0 0 1 1 0 0 1 0 0 0 0 0
EGI	0 0 0 0 1 0 1 0 1 0 0 0	1 1 1 0 1 1 0 1 0 0 0 0 0 1
A	1 0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1 1 1 1

Algorithm uses the pre-trained artificial neural network to compute user's access rights, eliminating a lot of look-up table operation, can effectively improve the efficiency of access control.

7. Summary

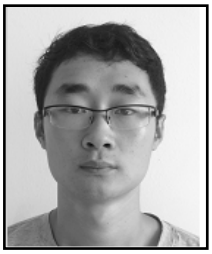
Dynamic data filtering policies of the present study is based on role-based control mechanism about access. It filters the set of operating data of users in the layer of applied and logical implementation. It provides a flexible and dynamic data security and fine-grained control of data for multi-user shared data in SaaS applications. It provides centralized authentication, authorization and access control for multiple services and users in the cloud environment to ensure the consistency of the data security policy enforcement. The data security monitoring platform uses the SaaS application model to provide users with fast, low-cost security services. By controlling fine-grained role-based authorization privileges and the data security monitoring platform increasing the concept of resource permissions, it manages users in a hierarchical authorization way. By defining the role of the relation of inheritance and inclusion relations, it improves performance of the RBAC model under environment of mass access control used in. The filtering rules in the platform use the dynamic SQL form which is stored in the form XML file so as to filter data easily and efficiently in the database side, and the to improve the performance of data filtering.

References

[1] G. Reese, “Cloud Application Architecture: Building Applications and Infrastructure in the Cloud”, O'Reilly Media Inc, Sebastopol, **(2009)**.
[2] X. D. Wu, “Data security research of cloud computing”, 26th National Computer Security Conference, wuyishan, China,**(2011)** September 16-19.
[3] W. Ren and Z. Q. Fu, University of science and technology of China, vol. 40, no. 1203, **(2011)**.
[4] J. Wei and Z. R. Yang, China management information, vol. 14, no. 29, **(2011)**.
[5] M. Lynch, “The Cloud Wars:100+billion at stake. Merrill Lynch research note, **(2008)** May.
[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, “Above the clouds: A Berkeley view of cloud computing”, Tech. Rep. UCB/EECS-2009-28, EECS Department, U.C. Berkeley, San Francisco, **(2009)**.
[7] G. Y. Hu, J. Computer measurement and control, vol. 19, no. 2539, **(2011)**.
[8] D. K. Wang, J. Journal of Beijing University of technology, vol. 32, no. 497, **(2006)**.
[9] N. Smart and F. Vercauteren, “Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes”, Springer, Beijing, **(2010)**.

Authors

Geng Yushui, is a Professor of School of Information, QiLu University of Technology. Currently he is serving as the departmental Chairmen. His research interests are cloud computing, Business Process Management, information integration. He has led more than 10 research projects.



Pang Shunpeng, is born on Nov 26, 1988, and currently a student of QiLu University of Technology. His research interest are cloud computing, data security