International Journal of Grid Distribution Computing
Vol.7, No.4 (2014), pp. 123-138
http://dx.doi.org/10.14257/ijgdc.2014.7.4.12

Game-Theoretic Strategy for Personalized Privacy Protection

Chao Yu and Yuliang Shi*

School of Computer Science and Technology, Shandong University, Jinan, China
yingpin_yc@163.com, shiyuliang@sdu.edu.cn

Abstract

With the development of cloud computing, more and more service providers deploy multi-tenant applications to the cloud. Multi-tenant data is stored by non-fully trusted SaaS service providers, and the protection of data privacy attracts more attention. This paper proposes a privacy protection strategy customization framework. This framework considers the privacy protection needs, SaaS application performance, the interests of both tenants and SaaS service providers, and analyzes the whole privacy protection strategy formulation process based on the Nash equilibrium, then establishes the game model of privacy protection, finally obtains the privacy protection strategy by analyzing of the game model. The experiments show that the privacy protection game model has better feasibility and effectiveness.

Keywords: *multi-tenant; Software as a Service (SaaS); Nash equilibrium; privacy protection*

1. Introduction

As a new service model, SaaS has been greatly admired by the academia. Many Internet companies have put forward their own SaaS cloud services. Although SaaS help tenants to reduce expenses, but tenant information will be stored remotely in a non-fully credible service provider platform, which brings many privacy security problems [1]. Malicious cloud service providers may leak tenant's private data, such as ID number, social security number, age, etc. In addition, in order to reduce costs, SaaS service providers usually adopt shared database shared memory mode to store multi-tenant data. Tenant data may be stored in the same data node, which reduces the safety of the tenants' privacy. Therefore, in the cloud environment, how to safely and effectively store data privacy has become the core issue and challenge for both SaaS service providers and tenants.

In order to avoid leakage of private data submitted by tenants, SaaS service providers need to protect privacy data privacy. As tenant data has relevance and semantic features, a single attribute or a combination of several attributes in the privacy data attribute set may leak tenants privacy, so for the protection of data privacy tenant should fully take into account the data attribute correlation between attributes. Literature [2] has adopted SaaS application-oriented combination of data privacy chunking algorithm, and cut the SaaS data attributes into different data blocks, reconstruct and confuse the relationship between attributes to achieve the tenants' data privacy protection. SaaS service providers provide application leases to tenants, at the same time, the reconstruction of logical view of the data object after dividing the data attribute blocks will affect the application performance. SaaS service providers not only complete the protection of tenants' data privacy, but also need to ensure the performance of SaaS applications to

*Corresponding author. Email: shiyuliang@sdu.edu.cn

ISSN: 2005-4262 IJGDC

meet the needs of the tenants. For each tenants, SaaS service providers are required to provide some virtual machine resources to protect tenants' privacy. Then, for multi-tenants, SaaS service providers need to adopt a privacy protection policy based on tenant needs and applications performance privacy, uniformly schedule limited virtual machine resources, balance interests between themselves and multi-tenants.

Multi-tenants and SaaS providers reach the privacy protection strategy after negotiation, due to the limit of SaaS service provider virtual machine total resources, tenants need to compete resources in order to meet their own SaaS application performance and privacy protection needs, and they want to spend less to achieve better privacy protection and application performance. SaaS service providers need to rational allocate resources to ensure their revenue maximization under the premise of the needs of tenants. Therefore, multi-tenants and SaaS service providers compete with each other on performance requirements and resource. We take into account the privacy protection performance and privacy needs of tenants, at the same time, consider the interests of both SaaS providers and tenants, introduce the Nash equilibrium game theory to balance the interests of both sides. The purpose is to achieve the equilibrium that any party cannot only change its own privacy policy to increase revenue or reduce cost. Finally, the ultimate privacy protection policy is determined by the tenants' privacy indicators in the equilibrium state.

This paper constructs Privacy Policy Framework as a whole, analyzes the process of the formulation of strategy between tenants and SaaS service providers, realizes privacy information security of tenants' data; Section 2 gives the related work; Section 3 describes the privacy protection services customized framework; Section 4 gives a detailed method to reach the privacy policy, the evidence and the resolution; Section 5 prove the feasibility and effectiveness of the privacy strategy by experiments; At last, Section6 lay out the summary and the further work.

2. Related Work

For the tenant data privacy protection in cloud environment, researchers tried to address this critical issue at all levels.

Privacy protection technologies have always been a hot research in academia. Literature [3] analyzes three aspects of service which are storage, transmission, processing of privacy information, and then studies privacy protection technology based on traditional data encryption, finally pointed out the need to develop rules to prevent privacy information leaked. Literature [4, 5] try to protect data privacy through the confusion, a privacy manager is designed to reduce the risk of privacy leakage in the cloud and increase privacy-related revenue, but the efficiency of privacy management needs to be improved. Literature [6, 7] propose the concept of privacy constraints, introduce data partition method to achieve the protection of data privacy.

A lot of researches try to further integrate research protocol to achieve the protection of privacy information of tenants. Literature [8] points out that cloud computing service design must take the protection of tenants' privacy into account. The article analyzes the design patterns of privacy information protection and the form of privacy requirements, but ignored a detailed privacy protection programs. Literature [9] analyzes the cloud privacy, security and other issues, and considers privacy considerations as a factor which affects the revenue. The paper presents a contract-based architecture to integrate cloud privacy and security issues, and then introduces a trusted third party to oversee privacy protection services. Literature [10] introduces the concept of privacy constraints, and proposes privacy QoS to ensure privacy security, and then makes a

detailed study and analysis in the PaaS level. Literature [11, 12] also try to establish privacy protection rules in the cloud to protect tenant privacy information, and analyzes further combining the privacy rules and practice. Literature [13] proposes privacy protection as a service. By extending SLA agreement, privacy constraints are considered as an evaluation of QoS. The literature [14] also proposes privacy protection as a service, adopts a series of security protocols to ensure the security of tenants' data privacy, and depends on homomorphic encryption to provide credible and isolated environment to enforce security policies.

For multi-tenant SaaS applications, performance is a key factor that both tenants and SaaS service providers concern, balancing the relationship between the degree and performance of privacy protection is another key point. Game model is often used to develop optimal strategies overall which aims to reach an equilibrium state on revenue among all parties participated in the game. Literature [15] proposes a network situation awareness method based on Markov game, and takes threats, administrators, ordinary users into account, then establishes tripartite participated Markov model to assess the security situation in the game. Literature [16] establishes a Markov queuing network model to balance the load between multiple controllers which has achieved a good effect, while the model hasn't got a good use in a SaaS environment. Literature [14] adopts the Nash equilibrium to allocate resource in the cloud, in every aspect of the auction mechanism, the idea of Nash equilibrium has been fully considered. The service provider determine the desired value they wanted to get based on the workload, each user depends auction price on the remaining time of requirement and the number of resource, and ultimately achieves multi-equilibrium through the game. Literature [17] also introduces the idea of Nash equilibrium into the allocation of cloud resource, and constructs Nash equilibrium model to estimate the potential benefits and take action in IaaS environment. In literature [18], researchers model the service scheduling problem as a general Nash equilibrium problem and propose an efficient algorithm to solve the problem which has been modeled as a Nash equilibrium problem, the paper specifies response time in the SLA as a metric, formulates the strategy in the way of competition between an IaaS provider and SaaS providers, but did not discuss the game between multi-tenants and the SaaS provider.

3. Personalized Privacy Protection Customized Framework

After tenants lease the SaaS application, the multi-tenant SaaS application data is stored on the service provider. For tenants' application data, the SaaS service provider must ensure the security of privacy data. In order to ensure the SaaS application performance requirements, some additional virtual machines resources need to be provided to compensate for performance overhead caused by privacy data reconstruction. If the number of virtual machine resources is fixed, the degree of privacy protection is bound to affect the level of application performance, SaaS service provider need to develop a rational strategy to balance the relationship between the privacy of multi-tenants application performance and the degree of privacy protection, as well as balance the relationship between multiple tenant lease payments and its own income.

Tenants' data information contains the privacy protection information (the information needed to protect privacy). SaaS providers submit tenants' privacy information, and then customize the privacy strategy for each tenant. The privacy protection information submitted by the tenant to the SaaS provider includes the privacy protection content and the privacy protection requirement, which includes the privacy protection level, privacy protection performance as well as privacy constraints.

Definition 1. Privacy protection content refers privacy protection attributes, values of attributes and.

Definition 2. The number of privacy protection data refers to the capacity of user-submitted privacy protection data.

Definition 3. Privacy protection strategy refers to the collection that contains the privacy protection level, the privacy protection performance and the lease expense.

Definition 4. The privacy protection level primarily reflects the degree of privacy protection, which the SaaS provider determines according to the number of blocks.

Definition 5. The privacy protection performance means the number of users' requirements from tenants per unit-time provided by the SaaS provider.

The design of privacy protection service customization framework is showed in Figure 1:

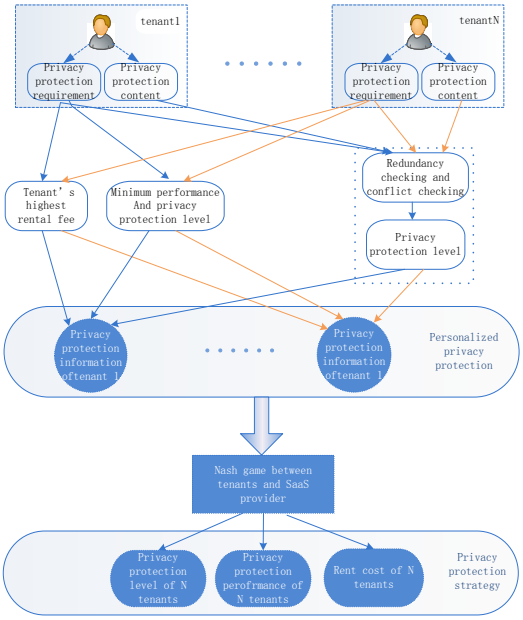

Figure 1. Personalized Privacy Protection Customized Framework

In the SaaS environment, we analyze the privacy protection content and the privacy protection needs, consider the dependency of four aspects of the privacy protection content, the privacy protection level, the privacy protection performance and the lease expense, and finally establish relevant the Nash equilibrium model to determine the appropriate privacy protection strategy.

The types of data privacy protection involve in the single privacy attribute protection and the combined privacy attribute protection. For different tenants, their privacy protection requirements are different. SaaS provider determines the content and privacy protection, etc by analyzing the individual privacy protection requirement, and provides the corresponding semantic constraints, set a threshold on privacy protection level and privacy protection performance, meanwhile, gives a detailed process of privacy strategy formulation.

Aiming at the privacy protection content submitted by tenants, we make for a unified single and combined privacy attribute-targeted validation rules, check the scope of the privacy protection content, finally, check and fix the conflict and redundancy of the privacy protection requirements from tenants, ensure that the expression of final privacy protection requirements is correctness and rationality.

We try to construct multi-tenants and SaaS provider revenue function which is associated with the privacy protection performance and level, and establish the Nash game model. Through the analysis of the model, we get the final tenants' and the SaaS provider's privacy strategy.

The SaaS provider offers virtual machine resources (virtual machines, referred VMs) to response the tenants' privacy protection requirement. Tenants lease SaaS application from SaaS provider, and submit the privacy protection information. After the SaaS provider got the privacy protection information, two tasks should be fulfilled: One is that SaaS provider determine the minimum level of privacy protection, the minimum performance requirements, the amount of data privacy protection, the highest cost based on tenants' privacy protection content; another is that according to the privacy protection constraints, the SaaS provider does redundancy check and conflict check, and determines the ultimate privacy protection constraints, divides the privacy protection level corresponded by the privacy protection constraint. Then the SaaS provider integrates two results to determine each tenant's privacy protection constraints which contents tenants lowest level of privacy protection, the minimum performance requirements for privacy protection, data privacy protection, privacy protection the highest costs.

In order to balance the interest relationship between the SaaS provider and multi-tenants, we introduce Nash equilibrium, consider tenants' privacy protect constraint information as a game theory situation. Then, we use game algorithm to obtain N tenants' privacy protection strategy, leasing costs and the SaaS provider's total revenue.

In the next section, we research and analyze the privacy protection framework, deduce privacy balanced game function associated with tenants and the SaaS provider, construct the game model, and then solve the model to get the specific privacy strategy.

4. Formulation of the Privacy Protection Strategy

4.1. Redundancy Checking and Conflict Checking

During the formulation of tenants' personalized privacy protection strategy, we analyze tenants' privacy protection requirements based on the privacy protection content. The privacy constraints submitted by tenants may have redundancy and conflict. Therefore, SaaS provider needs to check and improve the privacy constraints.

Definition 6. Privacy constraints [2] refer that some certain attitudes of the privacy blocks can not be designated to the same sub-block or one property requires to be divided into a separate block.

Privacy constraint redundancy means that the privacy protection requirements mentioned in tenants' privacy protection constraints expressed as PCi and PCj, the corresponding sets of attributes of privacy protection contents have containment relationship. After redundancy check, for incompatibility constraints (attributes in the set come out at the same time leak the privacy), we need to retain incompatible privacy constraints which have the smaller set of attributes (attributes in the set come out at the same time won't leak the privacy); for compatibility constraint, we need to retain compatible privacy constraints which have the larger set of attributes.

Privacy constraint conflict refers that for incompatible PC_i and compatible PC_j, the set of attributes corresponded by PC_i appears on PC_j. After the check of conflict, we should remove that compatible constraint.

After redundancy check and conflict check, we can complete to divide tenants' privacy constraints, and then, we can complete to divide tenants' privacy protection level.

4.2. Privacy Protection Level

For tenants' privacy protection information, The SaaS provider does analysis in accordance with customized privacy constraints *PPC* in privacy protection requirements of tenants. After redundancy check and conflict check, SaaS provider obtains new privacy constraints *PPC'*, and use *PPC'* for the division of tenants' privacy protection.

We put *PPC'* as the input section, based on the data privacy constraints block algorithm [2], under the premise of not violating the privacy constraints, constantly add new data attributes, or create a new block, ultimately, we can achieve the fewest blocks which meet privacy constraints. Obviously, on the condition of same amount of privacy protection data, the number of blocks will impact the performance. We relate the privacy protection level with blocks, and determine the privacy protection level through the number of blocks to. So, to a certain extent, privacy protection level is a factor that impacts the privacy protection performance.

Assuming that the minimum data block determined in current is $block_{min}$, while the maximum privacy block $block_{max}$ means that each attribute in the privacy protection contents will construct a block. So, the sub-interval of the number of blocks is expressed as L=[$block_{min}$, $block_{max}$]. As tenants' privacy protection requirements will involve the degree of privacy protection, according to which we can determine the minimum number of blocks as L_{min}, the maximum number of blocks as L_{max}, the new sub-sub-block interval is expressed as L'=[L_{min}, L_{max}]. Comparing B and B', we consider five situations as followed:

(1) If L_{min}>$block_{max}$, currently, maximum block cannot meet tenants' privacy protection requirements, SaaS provider will abandon the current selection.

(2) If L_{min}=$block_{max}$, currently,we can only choose the number of blocks as $block_{max}$.

(3) If L_{max}<$block_{min}$, SaaS provider cannot guarantee tenants' privacy without leaked, SaaS provider will abandon the current selection.

(4) If L_{max}=$block_{min}$, then currently,we can only choose the number of blocks as $block_{min}$.

(5) If L_{min}<$block_{max}$ and L_{max}>$block_{min}$, we respectively compare $block_{min}$ and L_{min}, $block_{max}$ and L_{max}, and get a new block range L^*=[L^*_{min}, L^*_{max}].

For the first four cases, privacy protection level is fixed, we mainly concern situation (5), that is to say, sub-block interval L^* has been determined. In order to construct Nash equilibrium model in the following section, we set that for the tenant r, sub-block interval is determined as [L^*_{min}, L^*_{max}], privacy protection level is divided into L^*_{min}-L^*_{max} levels, each of sub-block value represents a level. It should be noted that, for different tenants, because the amount of data privacy protection is different, even the value of level is the same, and the impact on privacy protection performance is different.

4.3. The Privacy Protection Strategy Model

The process that multi-tenants and SaaS provider determine the privacy protection strategy, in fact, can be described as a game theory between tenants and SaaS provider. On the one hand, each tenant needs to determine the privacy protection performance and privacy protection level to meet the privacy protection requirements. Tenants want to pay lease costs as little as possible, and achieve higher service performance and higher privacy protection level, while, SaaS provider want to improve their profit as much as possible, different tenants will be provided different service performance and privacy protection level. We use game theory equilibrium [22] to build privacy protection strategy model considering multi-tenants and SaaS provider, adopt a Nash equilibrium game method, the Nash equilibrium problem is solved to obtain each tenant's privacy strategy.

Tenants' goal is to determine the actual privacy protection level and performance to reduce the relative cost compared with building their own SaaS applications. For any one tenant *r*, he will submit the lowest privacy protection level $L_{min}(r)$ and the lowest privacy protection performance P_r. Assuming that tenants actually determine privacy protection level as L_r, privacy protection performance as P'_r, if P'_r is lower than P_r, tenants will be compensated by SaaS provider.

The income of tenant *r* can be presented as *cost*(*r*) that *r* builds its own applications itself minus *cost*(*SaaS*-*r*) that *r* rents SaaS application from service provider, that is:

$$Profit(r)=cost(r)-cost(SaaS-r) \quad (1)$$

The cost that tenant build their own applications is mainly buying computing and storage resources to support virtual machines. Assuming that the amount of tenant's data C_r is a fixed value, through experiments, the impact that the amount of data has on application performance *P* is linear and can be expressed as $P=\theta C_r+K$, where *K* and θ are constants, and $\theta<0$; the cost that application system processes unit data and achieves the unit performance is u_r, for the certain tenant r, u_r is constant. In addition, because tenants build applications themselves, data storage and management are achieved by themselves, so there is no privacy protection for data, and no requirements for privacy protection level. Thus, tenant r with the application performance P'_r, *r* needs to pay cost (r), formally expressed as:

$$cost(r)=\frac{K+\theta}{\theta C_r+K}\cdot P'_r \cdot u_r \quad (2)$$

If tenants rent SaaS applications, and SaaS provider supplies virtual machine resources to achieve the tenants' privacy protection information, the cost *cost*(*SaaS*-*r*) relates to performance P_r and P'_r, privacy protection level L_r, the number of tenants' data C_r, the number of virtual machine resources m_r. Tenants sign SLA agreements with SaaS provider on privacy protection, and pay according to privacy protection performance, the specific charge pattern is:

$$cost(SaaS-r)P_r\cdot V_r+(P'_r-P_r)\cdot V_r(+/-) \quad (3)$$

If $P'_r>P_r$, V_r(+/-) values V_r(+), if $P'_r<P_r$, V_r(+/-) values V_r(-), and $V_r(-)>V_r>V_r(+)$, V_r(+) and V_r(-) are considered as a expand and shrink percentage of V_r. The meaning of this charging model means that SLA agreement rules the privacy protection performance as P_r, the charge that SaaS respond to individual user requests is V_r, if actual performance P'_r doesn't reach P_r, the fee that performance values $P'_r - P_r$ is V_r(+),if actual performance P'_r is higher than P_r, the fee that performance values $P_r - P'_r$ is V_r(-).

SaaS provider's goal is to determine V_r for each tenant, SaaS provider's profit formally expressed as:

$$Profit(SaaS)=\sum_{r=1}^{n}\{cost(SaaS-r)-m_r\cdot\gamma\} \quad (4)$$

The number of tenants is *n*, the number of resources that meet tenant *r* 's performance P'_r is m_r, γ represents the cost that purchase each virtual machine.

Assuming that the max number of users of tenant *r* that the unit performance virtual machine resource can respond is U_{max}, m_r virtual machines can respond $m_r \cdot U_{max}$ users at the most. Meanwhile, privacy protection level raises from *i*-1 to *i*-1, the affect on performance is that the number of responded users will reduce the percentage of θ (*i*-1, *i*), where $i = 1,2, ..., L_{max}$. Well, for certain number of virtual machine resources and privacy protection level, the actual privacy protection performance can be formally described as:

$$P_r' = \frac{m_r \cdot U_{\max}}{\frac{K+\theta}{\theta \cdot C_r + K}} \cdot [1 - \sum_{i=1}^{L} \theta(i-1,i)] \tag{5}$$

In summary, combined from formula (1) to (5), the profit of tenant r can be expressed as:

$$Profit(r) = m_r \cdot U_{\max} \cdot [1 - \sum_{i=1}^{L} \theta(i-1,i)] \cdot u_r - \{P_r \cdot V_r + (\frac{m_r \cdot U_{\max}}{\frac{K+\theta}{\theta \cdot C_r + K}} - P_r)[1 - \sum_{i=1}^{L} \theta(i-1,i)] \cdot V_r(+,-)\} \tag{6}$$

Constraints are:

$$Profit(r) > 0 \tag{7}$$

$$P_r \cdot V_r + (\frac{m \cdot U_{\max}}{\frac{K+\theta}{\theta \cdot C_r + K}} - P_r) \cdot V_r(+,-) > 0 \tag{8}$$

$$L_{\min}(r) \le L_r \le L_{\max}(SaaS) \tag{9}$$

The meaning of constraints is: Constraint (7) indicates that the cost that tenants achieve application themselves must be more than the cost that tenants pay for the SaaS service, or tenants will not rent SaaS application; constraint (8) indicates that SaaS provider won't lose money to provide application; constraint (9) shows that actually determined privacy protection level must be no less than the minimum required tenants' privacy protection level and no greater than the highest tenants' privacy protection requirements.

Profit that SaaS provider gains can be expressed as:

$$Profit(SaaS\text{-}r) = P_r \cdot V_r + (\frac{m_r \cdot U_{\max}}{\frac{K+\theta}{\theta \cdot C_r + K}}[1 - \sum_{i=1}^{L} \theta(i-1,i)] - P_r) \cdot V_r(+,-) - m_r \cdot \gamma \tag{10}$$

Constraints are:

$$Profit(SaaS-r) > 0 \tag{11}$$

$$P_r \cdot V_r + (\frac{m_r \cdot U_{\max}}{\frac{K+\theta}{\theta \cdot C_r + K}} - P_r) \cdot V_r(+,-) > 0 \tag{12}$$

$$L_{\min}(r) \le L_r \le L_{\max}(SaaS) \tag{13}$$

$$V_{SaaS(\min)} < V_r < V_{r(\max)} \tag{14}$$

$$\sum_r m_r \le M \tag{15}$$

The meaning of constraints is: Constraint (11) indicates that the profit that service provider achieve from each tenant's SaaS applications is positive; constraint (13) indicates that the actual the privacy protection level must be no less than the minimum required tenants' privacy protection level and no greater than the highest level of privacy protection that tenants require; constraint (14) shows that the actual charges for single user request cannot exceed the maximum fees pay by tenants and cannot be less than the minimum fees that SaaS provider charges; constraint (15) shows that supposing that the largest amount of virtual machine resource provided by SaaS providers is *M*, the number of virtual machine of all tenants occupied for the privacy protection cannot exceed the total amount of virtual machine resources that SaaS provider provides.

In the profitprivacy protection customized model, SaaS provider and multi-tenants determine at the same time, tenants who make the decision need to consider other tenants and SaaS provider, SaaS provider also need to consider the multi- tenants' decisions, the decision making can be attributed to a game problem. We introduce the Nash equilibrium, tenants and

SaaS provider will eventually reach a strategy that no one can increase their profit through the way that change its own strategy alone. Profit function profit(*r*) and profit(*SaaS*) are continuous functions, the tenant 's strategy vector $x^r=(m_r, L_r)$, said vector x^{-r} represents other tenants' strategy vector set , for a continuous function values in a continuous interval, we argues strategy set X_r = { x^r: constraints(7,8,9) } and X_{SaaS} = {V_r: additional constraints } are closed, while the profit function *profit*(*r*) is a sum of a linear function and concave function, *profit*(*SaaS*) is a linear function , so they are both concave functions. According to the convexity assumption, profit function is a concave function, so the strategy is a convex function. Based on the above analysis, we can get that, the privacy protection model we provided belongs to a generally non-cooperative mixed Nash equilibrium problem.

4.4. The Existence Proof

We use the Nash existence theorem and Kakutani fixed point theorem to prove that equilibrium strategy of privacy protection customized model exists in the constraint range.

For any tenant *r*, its strategy set is X_r, x^{-r} is defined the strategy set of other tenants that is different from *r*, so $X_r(x^{-r})$ must contain strategy set: $X_r^L:=\{x^r \geqslant 0: L=L_{min}\}$, X_r^L means the strategy that makes the *profit*(*r*) minimum.

For any viable strategy V_r of SaaS, we get $profit_r^L(x^r) \le profit_r\{x^r, V_r\} \le profit_r^U(x^r)$. For *profit*(*r*), in the interval constraints will get the maximum value when L_r values $L_{min}(r)$, so:

$$Profit_r^L(x^r) = m \cdot U_{\max} \cdot [1 - \sum_{i=1}^{L_{\min(r)}} \theta(i-1,i)] \cdot u - \{P_r \cdot V_r + (\frac{m \cdot U_{\max}}{K+\theta} - P_r)[1 - \sum_{i=1}^{L_{\min(r)}} \theta(i-1,i)] \cdot V_r(+,-)\}$$

Either V_r (+) or V_r (-) has a maximum value, we choose the greater, there is $profit_r^U(x^r) = \frac{\theta \cdot C_r + K}{K+\theta} \cdot P_r \cdot u_r$. Assuming the tenants' maximum cost is the cost that tenants implement application themselves rather than rent SaaS application. Define $\psi_r = \max_{x^r \in X_r^L} profit_r^L(x^r)$; strategy set $\hat{X}_r := \{x^r : profit_r^U(x^r) \ge \psi^r\}$. The strategy set must be non-empty, convex sets and closed. Thus, for any viable strategy x^{-r} and V_r:

$$\max_{x^r \in X_r(x^{-r})} profit_r(x^r, V_r) \ge \max_{x^r \in X_r^L} profit_r(x^r, V_r) \ge \max_{x^r \in X_r^L} profit_r^L(x^r) = \psi^r$$

So $\arg\max_{x^r \in X_r(x^{-r})} profit_r(x^r, V_r) \subseteq \hat{X}_r$. This means $\hat{X}_r$ is the optimal consistency of the boundary, and then we consider closed convex set $\hat{X} = \hat{X}_1 \times ... \times \hat{X}_n \times X_{SaaS}$,

For the map Map（x, V_r）:

$$\text{Map}(x, V_r) = [\arg\max_{x^1 \in X_1(x^{-1})} profit_1(x^1, V_r)] \times ... \times [\arg\max_{x^n \in X_n(x^{-n})} profit_n(x^n, V_r)] \times \arg\max_{V_r' \in X_{SaaS}} profit_{SaaS}(x, V_r')$$

From the above analysis we can determine that, Map: $\hat{X} \to \hat{X}$ has non-empty convex closed set, so there must be a fixed point under the mapping Map (strategy (x, V_r)), so, the Nash equilibrium exists.

4.4. Solution of the Equilibrium Point

Through Add/Move operations, the overloaded data node in the overloaded set can meet the performance requirements of different tenants.

In the previous section, we have analyzed and got that the privacy protection model belongs to Generalized Nash Equilibrium Problem (GNEP). In his section, we introduce

monotone variational inequalities (VI) to solve the problem. For the vector X, with VI (X, F) indicates this monotone VI problem, if X satisfies that the objective function value is 0, and then X is a solution of GNEP, so, X is a convex closed set, which is expressed as

$$F = [\nabla x_r \, profit_r(x,\lambda)_{r=1}^{N}, \nabla x_{SaaS} profit_{SaaS}(x,\lambda)]^{T},$$

For F:

$$F = \begin{bmatrix} \frac{\partial \mathbf{profit}_1(x_1, V_r)}{\partial \boldsymbol{L}_1} \\ \frac{\partial \mathbf{profit}_1(x_1, V_r)}{\partial \boldsymbol{m}_1} \\ \frac{\partial \mathbf{profit}_2(x_2, V_r)}{\partial \boldsymbol{L}_2} \\ \frac{\partial \mathbf{profit}_2(x_2, V_r)}{\partial \boldsymbol{m}_2} \\ \cdots \\ \cdots \\ \frac{\partial \mathbf{profit}_{SaaS}(x_1, V_r)}{\partial V_r} \\ \cdots \end{bmatrix}$$

$$= \begin{bmatrix} \theta \cdot m_1 \cdot U_{\max} \cdot u - \theta \cdot P_r \cdot V_r(+,-) \\ -U_{\max}(1-\theta \cdot L) \cdot u + \frac{U_{\max}}{\frac{K+\theta}{\theta \cdot C_r + K}} \cdot (1-\theta L) \cdot V_r(+,-) \\ \theta \cdot m_2 \cdot U_{\max} \cdot u - \theta \cdot P_r \cdot V_r(+,-) \\ -U_{\max}(1-\theta \cdot L) \cdot u + \frac{U_{\max}}{\frac{K+\theta}{\theta \cdot C_r + K}} \cdot (1-\theta L) \cdot V_r(+,-) \\ \cdots \\ \cdots \\ m_1 \\ \cdots \end{bmatrix}$$

The jacobian matrix is:

$$JF = \begin{bmatrix} \theta \cdot U_{\max} \cdot u & & & & \\ & \theta \cdot U_{\max} \cdot u & & & \\ & & \cdots & & -B^T \\ & & & \cdots & \\ & & & \theta \cdot U_{\max} \cdot u & \\ & & B & & 0 \end{bmatrix}$$

JF is important because it reflects a optimal linear approximation between differentiable equation and the given point. Observing JF, it is easy to get that in the range of feasible solution X, nonzero eigenvalues are positive. Therefore, F is monotone, this VI problem turns to be an monotone VI problem, thereby the solution of the equilibrium point turns to be the solution of the monotone VI problem.

In this paper, we use the continuity iterative method to solve this VI monotone problem, introduce parametric ε, study the relationship between the monotone VI problem and the perturbed problem $P(\varepsilon)$ about ε, where $P=P(0)$. Based on the original VI problem, respectively analyze the property of the perturbed problem $P(\varepsilon)$ when ε is sufficiently large and small. Through solving the sequence perturbed problem $P(\varepsilon)$, we obtain the solution of

the original problem *P*. For the solution of the problem monotone VI *X*, introducing vector λ, satisfying:

$$L(x,\lambda,\varepsilon)=0,\quad 0\le\lambda\le -g(x)\ge 0, \tag{16}$$

And,

$$\lambda=\begin{pmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\\ \cdots\\ \lambda_N\end{pmatrix}\quad g(x)=P_rV_r+(\frac{m_r\cdot U_{\max}}{\frac{K+\theta}{\theta\cdot C_r+K}}-P_r)\cdot V_r(+,-) \tag{17}$$

$L(x,\lambda,\varepsilon)=F(x)+\varepsilon x-\nabla g(x)\cdot\lambda$,

According to the above equation, equations can be obtained:

$F(x)+\varepsilon x-\nabla g(x)\cdot\lambda=0\ \ \forall\varepsilon\ge 0$

Algorithm1: Continuity iterative algorithm.

Input: initial strategy x_0, vector λ

Output: The final strategy *x*

Procedure

Based on this equation L (x, λ, ε) = 0, the formation of continuity algorithms in solving PVIP (x, F, ε) as follows:

1: Select strictly positive monotonically decreasing to zero parameter ε_k (0 <= ε_k <= 1), the initial point w_0 = (x_0, λ), so that k = 0, go to step 2;

2: If L(w_k, ε_k) = 0, stop, (x_k, λ) is the solution; otherwise go to step 3;

3: Find the approximate solution of L (w_k, ε_{k+1}) = 0 w_{k+1}, so that k: = k +*1*, go to step 1;

end

5. Experiments

We built five application servers, the SaaS CRM application is deployed on the servers. At the same time, we set up three data storage servers. All tests are carried out on VMWare virtual machines, and all virtual machines have the same configuration: a 2.33GHz processor, 1G RAM and a 100Mbps Ethernet card. MySQL runs background as the storage engine.

5.1. Test on the Impact that the Amount of Tenants' Data has on Storage Servers

In order to test how the amount of data effects the performance of the storage server and get the spend that tenants build their own applications, and we set the application server load 120 requests/s. Then, we increase the amount of data tenants within the maximum capacity of the storage server, test performance changes that the data storage server responds to the application server which sends the requirement. Figure 2 shows the situation of changes that the requirement response time brings out along with the in growth of the amount of tenants' data from 10W to 300W. In the Figure 2, the x-axis represents the amount of tenant data (/ W), y-axis indicates the response time (/ ms)

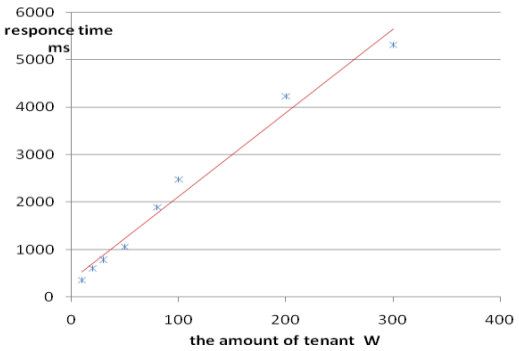

Figure 2. Test on the Performance of the Storage Server

Experimental results show that the impact which expanding the amount of tenants' data has on the performance of the store server can be approximated as linear, i.e. $P = \theta C_r + K$, where K and θ are constants, and $\theta < 0$.

5.2. Analysis of Privacy Protection Strategy

Experiment simulates the privacy protection strategy formulation process of six tenants and the SaaS provider, during the experiment, we set for each tenant that largest tenant number which a single virtual machine is available to respond is U_{max}, coefficient θ、K、$\theta(i\text{-}1,i)$, the value of a single virtual machine resource is γ, u_r which is unit performance unit amount of data is constant; In the strategy, arguments m_r、L、P_r, tenant data volume C_r, tenant lease charges V_r、$V(+)$、$V(-)$ as shown in table 1 have a random value within the range.

Table 1. Values of Experimental Variables

variable	value
U_{max}	7.5$
$\theta(i\text{-}1,i)$	0.6
m_r	[1,8]
C_r	[0.6,1]
L	[1,10]
P_r	[10,20]req/s
θ	-1
K	0.5
u_r	0.5
V_r	[0.5,1]$
γ	3$
$V(+)$	[0.75,1.25]$
$V(-)$	[0.25,0.45]$

Assuming the data storage server is not overloaded. We build equilibrium model according to the values of variables in table 1, then, adopt continuous iterative methods given in 4.3.3 to solve the equilibrium problem, and record the course of the SaaS service provider' and six tenants' income changes, consumption of the solution process time.

Figure 3 describes tenant1's profit function, demonstrates the change of tenant1's profit during the formulation of privacy strategy. The beveled portion represents the change of tenants' profit accompanied by privacy protection level L and virtual machine

resources allocated to tenants m_r. After reaching the equilibrium state, tenants' profit will not be effected by L and m_r.

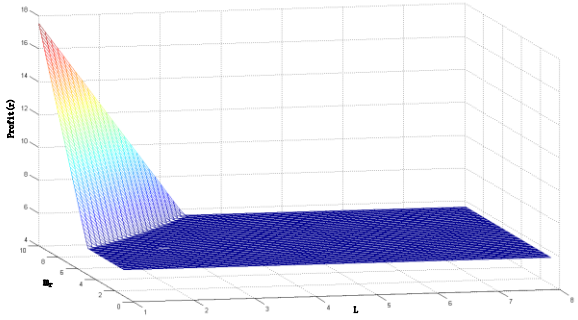

Figure 3. Profit Function of Tenant 1

Figure 4 shows the SaaS provider's profit function for the tenant1, the beveled portion represents the change of the SaaS provider's profit accompanied by privacy protection level L and virtual machine resources allocated to tenants m_r. After reaching the equilibrium state, the SaaS provider's profit will not be effected by L and m_r.

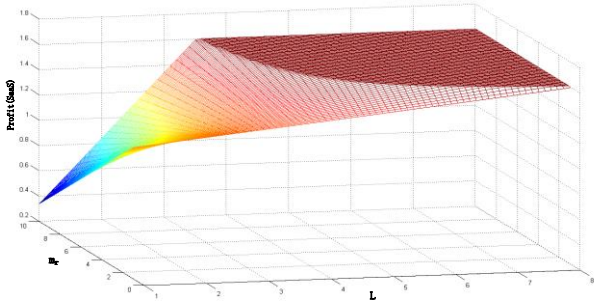

Figure 4. Profit Function of SaaS Provider

Figures 3 and 4 reflect the change of the SaaS provider's profit accompanied by the change of the strategy around the whole process of the game. The other five tenants' profit changes in the course of the game are similar with tenant1, and have the corresponding profit change of the SaaS providers. Ultimately, the SaaS provider's profit is the sum of each SaaS provider's profit competed with six tenants.

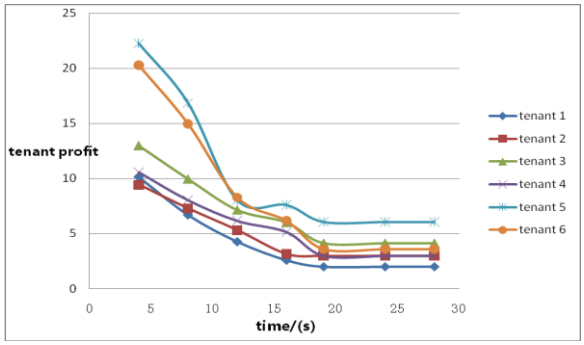

Figure 5. Profits of Multi-tenants Nash Equilibrium

Figure 5 reflects six tenants' profit changes under the proposed equalization algorithm accompanied by the change of time. It takes 19.01s to achieve a equilibrium, the algorithm's execution time is 19.01s. The experiment fully verifies the existence of equilibrium and equalization algorithm's performance.

6. Summarize

This paper proposes a privacy protection customization framework that comprehensively considers privacy protection requirements, SaaS application performance, the interests of both tenants and SaaS provider, introduces Nash equilibrium game model to balance relationships between privacy protection level and performance, and then makes a detailed analysis of the process of formulating the privacy protection strategy of tenants and the SaaS provider. By solving the game model, we determine the privacy protection strategy and achieve tenants' privacy data information security.

In this paper, the Nash equilibrium theory is adopted to formulate privacy protection strategy, due to the stringent constraints of the Nash equilibrium, in the complex cloud environment, when the same application response tenant's requirements in different time periods, we need to adjust the privacy protection strategy customization model. In the next step, according to this problem, we will be perfect for this framework.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 61272241), Independent Innovation Foundation of Shandong University (No. 2012TS074).

References

[1] H. Chang and E. Choi, "Challenges and Security in Cloud Computing [M]", Communication and Networking. Springer Berlin Heidelberg, **(2010),** pp. 214-217.
[2] K. Zhang, Q. Z. Li and Y. L. Shi, "Research on data combination privacy preservation mechanism for SaaS [J]", Jisuanji Xuebao (Chinese Journal of Computers), vol. 33, no. 11, **(2010)**, pp. 2044-2054.
[3] W. Pieters, "Security and privacy in the clouds: a bird's eye view [M]", Computers, privacy and data protection: An element of choice, Springer Netherlands, **(2011)**, pp. 445-457.
[4] M. Mowbray and S. Pearson, "A client-based privacy manager for cloud computing [C]", Proceedings of the Fourth International ICST Conference on Communication System software and middleware, ACM, (2**009),** p. 5.
[5] S. Pearson, Y. Shen and M. Mowbray, "A privacy manager for cloud computing [M]", Cloud Computing, Springer Berlin Heidelberg, (2**00),** pp. 90-106.
[6] G. Aggarwal, M. Bawa and P. Ganesan, "Two can keep a secret: A distributed architecture for secure database services [J]", CIDR (**2005)**.
[7] Y. Shen, W. Cui and Q. Li, "Hybrid Fragmentation to Preserve Data Privacy for SaaS [C]", Web Information Systems and Applications Conference (WISA), 2011 Eighth. IEEE, **(2011)**, pp. 3-6.
[8] S. Pearson, "Taking account of privacy when designing cloud computing services [C]", Software Engineering Challenges of Cloud Computing, CLOUD'09. ICSE Workshop on. IEEE, **(2009)**, pp. 44-52.
[9] M. Schnjakin, R. Alnem and C. Meinel, "Contract-based cloud architecture", In CloudDB '10 Proceedings of the second international workshop on Cloud data management.
[10] D. S. Allison and M. A. M. Capretz, "Furthering the growth of cloud computing by providing privacy as a service [M]", Information and Communication on Technology for the Fight against Global Warming. Springer Berlin Heidelberg, **(2011),** pp. 64-78.
[11] J. Ruiter and M. Warnier, "Privacy regulations for cloud computing: Compliance and implementation in theory and practice [M]", Computers, privacy and data protection: an element of choice. Springer Netherlands, (**2011**), pp. 361-376.
[12] S. Porwal, S. K. Nair and T. Dimitrakos, "Regulatory Impact of Data Protection and Privacy in the Cloud [M]", Trust Management V. Springer Berlin Heidelberg, **(2011)**, pp. 290-299.

[13] D. Ardagna, B. Panicucci and M. Passacantando, "A game theoretic formulation of the service provisioning problem in cloud systems [C]", Proceedings of the 20th international conference on World Wide Web, ACM, **(2011),** pp. 177-186.
[14] D. Sun, G. Chang and C. Wang, *et al.* "Efficient Nash equilibrium based cloud resource allocation by using a continuous double auction [C]", Computer Design and Applications (ICCDA), 2010 International Conference on. IEEE, (**2010)**, vol. 1, V1-94-V1-99.
[15] Y. Zhang, X. B. Tan and X. L. Cui, "Network security situation awareness approach based on Markov Game model [J]", Journal of Software, **(2011)**, vol. 22, no. 3, pp. 495-508.
[16] M. T. T. Hsiao and A. A. Lazar, "Optimal decentralized flow control of Markovian queuing networks with multiple controllers [J]", Performance Evaluation, vol. 13, no.3, **(1991)**, pp. 181-204.
[17] G. Wei, A. V. Vasilakos and Y. Zheng, "A game-theoretic method of fair resource allocation for cloud computing services [J]", The Journal of Supercomputing, vol. 54, no. 2, **(2010)**, pp. 252-269.
[18] D. Ardagna, B. Panicucci and M. Passacantando, "A game theoretic formulation of the service provisioning problem in cloud systems [C]", Proceedings of the 20th international conference on World Wide Web, ACM, (**2011**), pp. 177-186.
[19] F. Facchinei and C. Kanzow, "Generalized Nash equilibrium problems [J]", 4OR, vol. 5, no. 3, **(2007)**, pp. 173-210.

Authors

Chao Yu, he was born in 1988. M. S. candidate. His research interests include cloud computing and privacy preserving.

Yuliang Shi, he was born in 1978. Ph. D. and associate professor. His research interests include service computing, cloud computing and database. He is a membership of China Computer Federation (E200011607M).