Study on Trust Model for Multi-users in Cloud Computing

Xu Wu[1,2]

(Corresponding author: Xu Wu)

School of Computer, Electronics and Information, Guangxi University[1]
No. 100 East Daxue Road, Nanning, Guangxi 530004, China
Key Laboratory of Multimedia Communication and Network Technology, Guangxi University, Nanning, China[2]
(Email: xrdz2005@163.com)

(Received Mar. 12, 2017; revised and accepted June 20, 2017)

Abstract

Although some trust management approaches are proposed for cloud computing, these approaches only deal with single and simple trust relationship, trust algorithms in these models are one dimensional, and can not accurately measure the trust relationship between multi-users. In the paper we present a cloud trust model based on trust level agreement. The proposed method can assist cloud computing entities to make good interaction decisions. The main contribution of this paper is to provide a hierarchical trust modeling method to user and improve his or her security situational awareness in the cloud computing environments. The experimental results show that the proposed method has a higher trust accurate rate and interaction success rate, and it is qualified to prevent malicious entities attacks while maintaining efficiency. Our analysis shows a significant improvement in comparison to traditional trust management technology. Our work appears to be the first attempt to research the multi-entities trust management method in cloud computing.

Keywords: Cloud Computing; Decision-Making; Trust Management

1 Introduction

With the development of virtualization technology, computers have transited from the real to a virtual machine; people begin to pursue lightweight computing service [12]. As P2P network, grid computing, utility computing and a series of distributed computing technology are constantly emerging, and they create a new kind of distributed computing technology, cloud computing. Cloud computing brings a shift from heavy IT infrastructure invest for limited resources that are internally managed and owned by a customer to pay per use for IT infrastructure owned by a cloud computing service provider. There are many benefits to cloud computing: lower overall cost of IT ownership, increased flexibility, fault tolerance, locality flexibility ability, and to respond to new business requirements quickly and efficiently [13]. However, cloud computing in industry is not as popular as it is in the academia at present, the reason is that user distrust cloud computing environment, and they are not willing to put their private information and data in the computer of a third party. Therefore, the problem of trust in cloud computing environment is becoming more and more serious; a lack of trust between cloud customers and providers has hindered the universal acceptance of clouds as an increasingly popular approach for the processing of large data sets and computationally expensive programs [1].

A good solution is to leverage trust management technology to build trust for cloud computing [3]. Trust management is fundamental to identify malicious, selfish and compromised nodes which have been authenticated. It has been widely studied in many network environments such as peer-to-peer networks, grid and pervasive computing and so on. Trust is an important aspect in the design and analysis of secure distribution systems. It is also one of the most important concepts guiding decision-making. Trust is a critical part of the process by which relationships develop. It is a before-security issue in the ad hoc networks. By clarifying the trust relationship, it will be much easier to take proper security measures, and make correct decision on any security issues. Trust modeling is a technical approach to represent trust for digital processing. Recently, trust modeling is paid more and more attention in cloud computing. Although some trust management approaches are proposed for cloud computing, these approaches only deal with single and simple trust relationship, trust algorithms in these models are one dimensional, and can't accurately measure the trust relationship between multi-users. Therefore, in the paper we propose a cloud trust model based on a trust level agreement (TLABCTM). The major contributions of this paper can be summarized as follows:

1) We present a hierarchical trust management framework. In the framework, trust is divided into three layers: cloud service provider trust layer (CSPTL), cloud component trust layer (CCTL) and cloud user trust layer (CUTL).

2) We propose a trust level agreement which includes two types: User Trust Level Agreement and Provider Trust Level Agreement. The trust level agreement classifies the identity of entity and service type. Users can obtain correspond cloud service according to their trust level agreement.

3) Our work appears to be the first attempt to research the multi-entities trust management method in cloud computing. The experimental results show that the proposed method has a higher trust accurate rate and interaction success rate, and it is qualified to prevent malicious entities' attacks while maintaining efficiency.

This paper is organized as follows. Section 2 describes related work. In Section 3, the proposed cloud trust model (TLABCTM) is discussed. Section 4 presents the experimental setup used to test the mechanism along with the results. Finally, we conclude with a summary of our results and directions for new research in Section 5.

2 Related Work

The issue of establishing trust in different environments has been discussed by many authors. Kumar *et al.* [9] present a novel approach to secure the Mobile Ad Hoc Networks. Error correcting codes are used to assign identification to resource constrained mobile nodes. This assignment helps to create centralized environment with subgroups, groups and hierarchies.

Deverajan *et al.* [5] propose a new protocol namely Adaptive Fuzzy DoT Threshold Routing Algorithm (AFTRA), which takes into account the Degree of Trust (DoT), connectivity and the energy levels. AFTRA provides all possible routes from the source to the destination. The best route is selected by considering three aspects hop count, trust values and energy. In addition, Deverajan *et al.* [6] also propose a novel trust based system to detect the intrusive behavior. The entire work of this system can be compartmentalized into three phases. They are Pre-eminent node selection, Inter-cluster trust rate computation, Intra-cluster trust rate computation.

Hwang *et al.* [8] distinguish among different service-level agreements (SLAs) by their variable degree of shared responsibility between cloud providers and users. Critical security issues include data integrity, user confidentiality, and trust among providers, individual users, and user groups. The three most popular cloud service models have varying security demands. The work in [17] is a very recently work on trust management in IoT environments. A trusted service platform is established, which provides trust evaluation based on three trust metrics. These metrics include Reputation, Recommendation, and Knowledge. The idea of the proposed method comes from modeling human trust relationship.

In [16] a distributed reputation based trust management system is presented for hybrid cloud computing system. The mechanism can effectively address strategic feedbacks and mitigate unfairness. The performance of the proposed trust management system has been studied in a simulated environment and due to space limitations this information is not fully provided. In order to solve privacy and security problems in the IaaS service layer, a model of trustworthy cloud computing which provides a closed execution environment for the confidential execution of virtual machines was proposed [20]. This work has shown how the problem can be solved using a Trusted Platform Module.

Zhimin *et al.* [19] propose a collaborative trust model for firewalls in cloud computing. The model has three advantages: 1) it uses different security policies for different domains; 2) it considers the transaction contexts, historic data of entities and their influence in the dynamic measurement of the trust value; and 3) the trust model is compatible with the firewall and does not break its local control policies. Hada *et al.* [7] propose a trust model for cloud architecture which uses mobile agent as security agents to acquire useful information from the virtual machine which the user and service provider can utilize to keep track of privacy of their data and virtual machines.

Edna *et al.* [4] presented an overview of the cloud computing paradigm, as well as its main features, architectures and deployment models. Moreover, they identified the main issues related to trust and security in cloud computing environments. Cloud service providers (CSP) should guarantee the services they offer, without violating users' privacy and confidentiality rights. Li *et al.* [11] introduced a multitenancy trusted computing environment model (MTCEM). This model was designed for the IaaS layer with the goal of ensuring a trustworthy cloud computing environment to users.

Pawar *et al.* [15] propose an uncertainty model and define an approach to compute opinion for cloud service providers. Using subjective logic operators along with the computed opinion values, they propose mechanisms to calculate the reputation of cloud service providers. They also evaluate and compare the proposed model with existing reputation models. T-broker [10] presents, a trust-aware service brokering scheme for efficient matching cloud services (or resources) to satisfy various user requests. The experimental results show that, compared with the existing approaches, T-broker yields very good results in many typical cases, and the proposed system is robust to deal with various numbers of dynamic service behavior from multiple cloud sites.

In the paper [14], the author describes the design and implementation of CloudArmor, a reputation-based trust management framework that provides a set of functionalities to deliver Trust as a Service (TaaS). The experimental

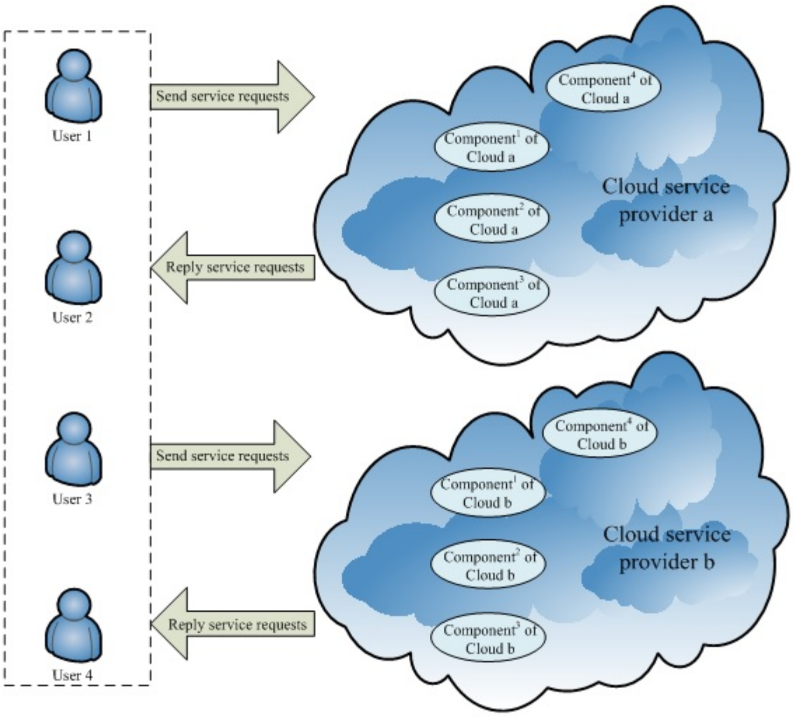


Figure 1: An example of cloud computing environment

results demonstrate the applicability of our approach and show the capability of detecting such malicious behaviors.

3 Cloud Trust Model Based on Trust Level Agreement (TLABCTM)

In this section, we firstly present an example of cloud computing environment. Secondly, we present the definitions about trust level agreement and the main idea of TLABCTM. Finally, we present the details about how to evaluate the trustworthiness of entities.

3.1 Scenario

Cloud computing is a large-scale and dynamic computing environment, so the types of entity which is involved in are different. The current academic circles widely divide these entities into two categories in the cloud computing: cloud service providers and users. However, with the development of cloud computing technology, in cloud computing appears a new entity identity-”component”. In Figure 1, every cloud has different components which supply corresponding cloud service to users. The component may be an artificial agent or middleware software. Different entities are linked to Internet in cloud dynamically. These entities include cloud service providers, cloud users and cloud’s components as shown in Figure 1, and all entities would dynamically enter or exit the virtual organization in cloud computing environment.

Cloud Service provider (CSP), it provides various cloud service for cloud computing environments, such as Software as a Service, Platform as a Service, Infrastructure as a Service, etc., it is a collection of components.

Cloud User (CU), it is a service entity which uses computing resources, storage resources in Cloud computing environments.

Cloud Component (CC), it is the actual carrier of service in cloud computing environment. Each CSP has j cloud components ($i.e.$, $CC = CC_1, CC_2, \cdots, CC_j$).

A CU sends service request to a CSP. Then the CSP will decide whether the CU is trusted or not. If the CU is trusted, CSP will select certain trusted CC ($i.e.$, component with trust values exceeding a threshold) to supply service.

In Figure 1, there exist multi-trust relationship (trust between CSP and CC, trust between CU and CC, trust between CSP and CU, trust between CU and CU, trust between CSP and CSP, and CC and CC).

3.2 Main idea of TLABCTM

In order to meet multi-entities’ trust requirements, we propose a trust level agreement based cloud trust model (TLABCTM). The Figure 2 presents the main idea.

A hierarchical trust management framework is established. In the framework, trust is divided into three layers: cloud service provider trust layer (CSPTL), cloud component trust layer (CCTL) and cloud user trust layer (CUTL). Entities are divided into three types in TLABCTM: CSP, CU and CC. A cross-layer trust flow is

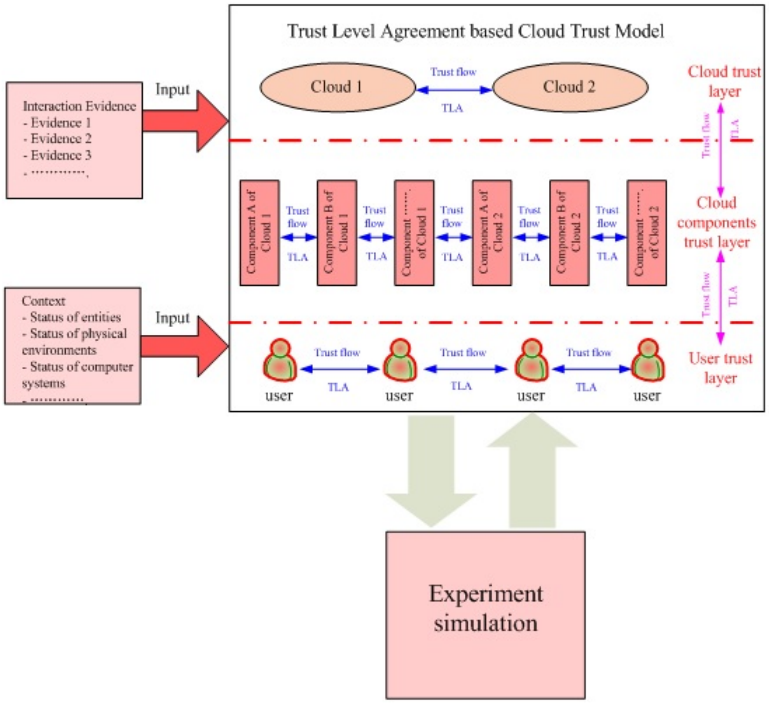


Figure 2: A trust level agreement based cloud trust model

established through the following levels of trust: CSPTL, CCTL and CUTL, so trust evaluation in TLABCTM includes two levels: between the layers and in the same layer. Trust evaluation between the layers involves evaluating the trustworthiness between CSPTL and CCTL and between CCTL and CUTL. Specifically, CSP calculates trust value for other CSP'CC; CC calculates trust value for other CSP; and CC calculates trust value for CU, CU calculates trust value for CC. Trust evaluation in the same layer involves evaluating the trustworthiness in CSPTL and in CCTL and in CUTL. Specifically, CSP calculates trust value for other CSP; CC calculates trust value for other CC; and CU calculates trust value for other CU. We will present the details about how to calculate the trust value in Section 3.3.

In Figure 2, UTLA and PTLA respectively denote User Trust Level Agreement and Provider Trust Level Agreement. UTLA and PTLA are two kinds of TLA agreement. Each CSP and CU respectively maintains UTLA table and PTLA table, which are shown in Tables 1 and 2.

Definition 1. *User Trust Level Agreement (UTLA): It is denoted as $\{IDEE, UST, UT, ET\}$. IDE represents the identity of a CU. UST represents requesting service type (operation type and information type, etc.). UT represents the trust value of CU. ET represents the lowest expected trust value of requesting service. UTLA shows requesting service type, the lowest expected trust value of requesting service and the trust value of CU, when an entity acts as a CU.*

Definition 2. *Provider Trust Level Agreement (PTLA): It is denoted as $\{IDE, CCTY, CCPST, CCT\}$. IDE represents the identity of a CSP. CCTY represents the type of CC. CCPST represents the providing service type of CC. CCT represents the trust value of CC. PTLA shows the providing service type and trust value of CC, when an entity acts as a CSP.*

Table 1: An example of CU_i'$UTLA$ table

IDE	UT	UST	ET
CU_i	0.8	Service Type 1	0.3
		Service Type 2	0.4
		...	...

Table 2: An example of CSP_k'$PTLA$ table

IDE	CCTY	CCT	CCPST
CSP_k	CC_1	0.3	Service Type 1
			...
	CC_2	0.4	Service Type 1
			...
	...	...	...
	CC_j	0.3	...

Consider the situation in Figure 1 where CU_i wants to interact with CSP_k. Based on the proposed TLABCTM, the CU_i will send UTLA to the CSP_k. Then the CSP_k

will estimate whether the CU_i is trusted by checking in UTLA and whether it can provide the service type by checking UST. If the CU_i is trusted and CSP_k'CC_j can provide the service type, CSP_k will send PTLA to the CU_i and select CC_j to provide the service. Then the CU_i will decide whether it should use the service by checking in PTLA. If CCT < ET. the CU_i will refuse to use the service.

3.3 Trust Evaluation of Entity

In this paper, we define trust as an expectation about the behaviors of what an entity, say E_i, expects another entity, say E_j, to perform in a given context. Each entity uses trust values to determine whether it can trust the other entity or not. The trust of entity is represented as a binary value. There are two ways in which to calculate trust value: direct and recommendation. When E_i has enough interaction experience with E_j, E_i uses direct trust to calculate the trust value for E_j. On the other hand, when E_i doesn't have enough interaction experience with E_j, E_i uses recommendation trust to calculate the trust value for E_j. In our paper, an interaction experience threshold is predefines based on the number of interactions in a cloud computing system.

1) Direct trust:
The direct trust value $DT_{E_i}(E_j)$ is defined as:

$$\begin{aligned} DT_{E_i}(E_j) \quad &= \quad \alpha \times \sum_{m=0}^{N(E_j)} (\frac{S(E_i, E_j) \times Z}{N(E_j)} \\ &\quad + pen(m)\frac{1}{1+e^{-n}}) + \beta Risk(E_j). \end{aligned} \tag{1}$$

The computing method of direct trust value is proposed by us in the previous work [18]. α and β are weighting factors that satisfies the condition $\alpha + \beta = 1$. $N(E_j)$ denotes the total number of interactions that E_i has performed with E_j and $S(E_i, E_j)$ denotes the E_i's satisfaction degree of interaction in its ith interaction with E_j which is in the range of (-1, 1). We use Z to denote the time factor. Thus,

$$Z = \mu(t_m, t_{now}) = \frac{1}{t - now - t_m}, Z \in (0, 1) \tag{2}$$

where t_m is the time when the mth interaction occurs and t_{now} is the current time. $pen(m)$ denotes the punishment function and

$$pen(m) \quad = \quad \begin{cases} 1 & \text{if the } m\text{th interaction fails} \\ 0 & \text{if the } m\text{th interaction succeeds} \end{cases}$$

$\frac{1}{1+e^{-n}}$ is the acceleration factor where n denotes the number of failures. It can make trust value drop fast when an interaction fails. As this factor increases with n, it helps avoid heavy penalty simply because of a few unintentional cheats. Finally, $Risk(y)$ is used to express the risk factor.

From Formula (1) we can see that just one time of deception or bad service of trustee may cause its trustor totally distrusts the trustee from then on. If the trustee itself is a just malicious entity, its information or service may not be used and considered by the trustor again; else the trustee does not like to sacrifice the precious and hard established trust value with the trustor.

2) Recommendation trust value
The recommendation trust value $RT_{E_i}(E_j)$ is defined as:

$$RT_{E_i}(E_j) = \sum_{\mu} DT_{E_i}(E_\mu) DT_{E_\mu}(E_j)$$

where $RT_{E_i}(E_j)$ represents the trust that entity E_i places in entity E_j based on asking his friends.

We can write this in matrix notation: If we define DT to be the matrix $[DT_{E_i}(E_\mu)]$ and $\overrightarrow{RT_{E_i}}$ to be vector containing the values RT_{E_i}, then $\overrightarrow{RT_{E_i}} = DT^T \overrightarrow{DT_{E_i}}$. This is a useful way to have each entity gain a view of the cloud computing network that is wider than his own experience. However, the trust values stored by E_i still reflect only the experience of E_i and his acquaintances. In order to get a wider view, E_i may wish to ask his friends' friend $(RT = DT^T)^2 DT_{E_i})$. If he continues in this manner, $(RT = (DT^T)^n DT_{E_i})$, he will have a complete view of the network after n = large iterations. Fortunately, if n is large, the trust vector $\overrightarrow{RT_{E_i}}$ will converge to the same vector for every entity E_i. In other words, $\overrightarrow{RT}$ is a global trust vector in this model. Its elements, RT_{E_μ} quantify how much trust the system as a whole places entity E_μ.

The mechanism of computing recommendation trust allows entities to calculate a recommendation trust for other entities with the recommendation information which is collected by flooding reference trust requests to entities' friends. However, in a large scale cloud computing environment, the mechanism is not scalable due to message overhead problem. From the perspective of sociology, the evidence of trust evaluation between individuals is from direct interaction experience and others' recommendation, but not all others' recommendation information must be collected. According to people's experience of cognitive psychology, old knowledge has less infection and new knowledge has more contribution to trust decision. That is to say, trust value has the attribute of dynamic attenuation over time decay. The trust dynamic nature refers to the trust value of an entity on another entity changes over time due to newer interactions, so the recommendation information should come from newer interactions. The entities in the same layer have the same interaction scenarios and similar interaction requirement, so the recommendation information from the same layer has a higher accuracy than the one from the other layer.

Based on the above description, we defines an Available Recommendation Entity Set (ARES) to decrease the number of recommendation entity.

Definition 3. *Available Recommendation Entity Set (ARES): The recommendation entity in ARES must satisfy the following three conditions: 1) Recommendation information of the recommendation entity should come from newer interactions; 2) The recommendation entity's trust value shall exceed a trust threshold value T ($T = 0.5$ in the paper); 3) The recommendation entity is at the same layer with the requesting entity sending reference trust request.*

4 Experimental Study

In this section, in order to evaluate the effectiveness of TLABCTM, a series of test scenarios are developed. The platform of simulation environment is CloudSim toolkit [2] which is a simulation platform based on Java, which supports modeling and simulation of large scale cloud computing data centers. Therefore, it is feasible to simulate our proposed model of cloud computing environments by CloudSim. Each service provider possesses a set of cloud components and the set sizes of all service providers are uniformly distributed (*i.e.* from 1 to 10). One or more components can be combined to produce a cloud service. We generate 100 service providers and 10000 cloud users and 50 kinds of different cloud components. All entities are divided three types: (1) Virtuous entities P_v that provide honest and accurate recommendation data about the other entity; (2) Random entities P_R that provide random recommendation; (3) Malicious entities P_M that provide malicious and false recommendation. We choose to use four metrics, the accuracy rate, response time of trust computing, interaction success rate and change of trust result, to evaluate the performance of TLABCTM, T-Broker [10] and CloudArmor [14]. All simulations were conducted over 1000 sessions. Table 3 shows the parameters used in our experiments. We assume the scenario: Cloud users consume services offered by the service providers. Occasionally, users may require new services, which need other service providers as support. In this case, cloud service provider may contact other service providers to form a collaborative group to share components to fulfill new service requirements. Note that some service providers may reject the collaboration invitation due to various reasons such as limited profits, etc. If no service providers are willing to collaborate, the collaborative group will not be formed.

4.1 Evaluation of Trust Accurate Rate

In the first experiment, we evaluate trust accuracy rate which means the rate of obtaining correct trust results through trust management model on the precondition that all the trust management tasks assigned are completely accomplished within its deadline. The trust accurate rate is compared in TLABCTM, T-Broker [10] and CloudArmor [14]. We submit 100 tasks to 10000 cloud users in order to evaluation certain cloud component. As showed in Figure 3, with the increase of the malicious rate (the percentage of Malicious entities P_M), the TLABCTM can ensure trust accuracy rate in a relatively high level, even when malicious rate is up to 55%, trust accuracy rate is still above 83.5%, and thus it proves the advantage of our model on preventing the behavior of associated cheat of users.

Table 3: Default simulations parameters in the experiments

Service providers	100
Cloud users	10000
Cloud components	50
The rate of virtuous entities	0% - 30%
The rate of random entities	0% - 50%
The rate of malicious entities	0% - 50%

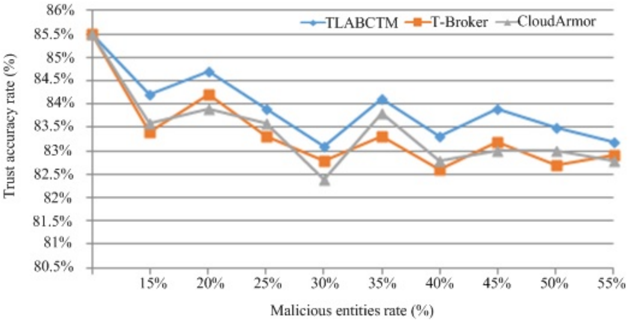
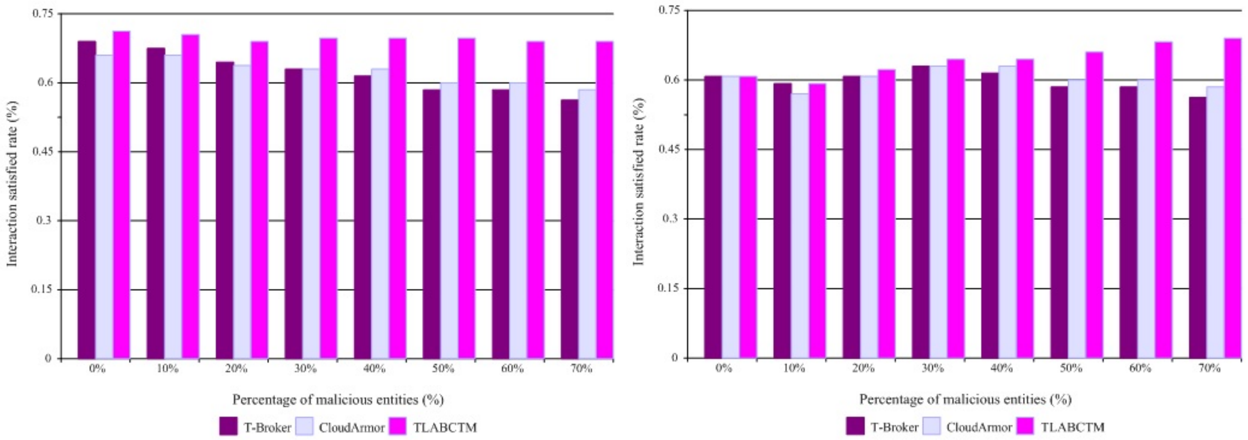


Figure 3: Trust accuracy rate

4.2 Interaction Satisfied Rate

Interaction satisfied rate expresses the satisfied rate in worst case scenario when a cloud user interacts with cloud service providers for getting cloud services. The interaction satisfied rate (ISR): if cloud user has $Number_{total}$ interactions and $Number_{satisfy}$ of them are interactions with satisfied feedback, then $ISR = \frac{Number_{satisfy}}{Number_{total}}$. The interaction satisfied rate is evaluated in the group of experiments. We add a number of malicious servers to the network such that malicious providers make up between 0% and 70% of all servers in the network. For each fraction in steps of 10% we run experiments under two attack models separately and depict the results in in Figure 4a and Figure 4b. We observed a 70% interaction satisfied rate of our mechanism at least in Figure 4a and Figure 4b. For independent cheat and group cheat, our scheme



(a) Independent cheat

(b) Group cheat

Figure 4: Simulation results of entities under cheat

performs well even if a majority of malicious providers is present in the network at a prominent place. Even if no malicious providers are present in the system, providers are evaluated as malicious in 3%-5% of all cases - this accounts for mistakes providers make when providing a service, e.g., by providing the wrong meta-data or creating and sharing an unreadable file. As Figure 4(a) and Figure 4(b) shows, comparing with T-Broker [10] and CloudArmor [14], TLABCTM gets more efficient. The main reason is that TLABCTM use trust level agreement to adapt cloud user's service requirement. Before interacting, a cloud user will give out the lowest expected trust value of requesting service. Only when the trust value of component of cloud providers is higher than the lowest expected trust value, the cloud user will decide to use the cloud service provided by cloud providers.

4.3 Response Time

In the third experiment, we evaluate response time which means the time of obtaining correct trust results through trust management model on the precondition that all the trust management tasks assigned are completely accomplished within its deadline. We submit more than 100 tasks to 10000 cloud users in order to evaluation certain cloud component. The response time is compared in TLABCTM, T-Broker [10] and CloudArmor [14]. From Figure 5 we can see that when T-Broker and CloudArmor are used, the response time is about 750ms and 800ms, while the TLABCTM has the lowest response time 500ms. This is because, by introducing ARES, the service provider is capable of decreasing the number of recommendation entity and eliminating untrustworthy recommendation entity, thus reducing the trust computing time, while there aren't any methods provided to selecting recommendation information in T-Broker and CloudArmor.

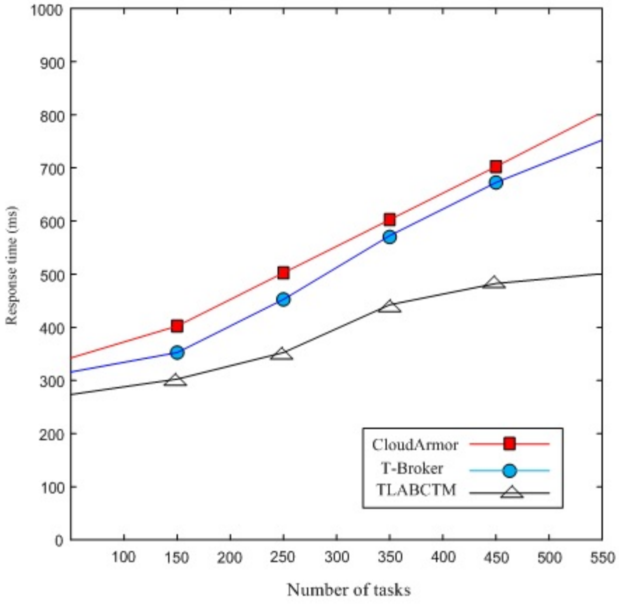


Figure 5: Response time

5 Conclusions and Future

In the paper we present a cloud trust model based on trust level agreement (TLABCTM), the proposed method not only improves the accuracy of the trust management, and satisfies the trust evaluation requirement of multi-entities in cloud computing environment. In addition, the proposed method can reduce the complexity of trust computing and management and assist cloud computing participants to make good trust decisions. In reality, trust is a social problem, not a purely technical issue. However, we believe that technology can provide the cloud customers with a way of measuring the claims of the cloud service provider as to how trustworthy their clouds are. The benefits of TLABCTM are threefold: First, it presents a hierarchical trust management framework. It divides trust into three layers: cloud trust layer, cloud component trust layer and user trust layer. Trust relationship between multi-entities is evaluated in TLABCTM. Second, it deals with the dynamic of trust evaluation. TLABCTM classifies the identity of entity, service type, and users can obtain correspond cloud service according to their service requests; and third, it gives a better understanding of cloud components.

Acknowledgements

The work in this paper has been supported by National Natural Science Foundation of China (Program No. 71501156 and No.61373116) and China Postdoctoral Science Foundation (Program No.2014M560796) and special funding for key discipline construction of general institutions of higher learning from Shanxi province.

References

[1] D. S. AbdElminaam, "Improving the security of cloud computing by building new hybrid cryptography algorithms," *International Journal of Electronics and Information Engineering*, vol. 8, no. 1, pp. 40-48, 2018.

[2] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities", in *Proceedings of the International Conference on High Performance Computing & Simulation*, pp. 1–11, June 2009.

[3] P. S. Chung, C. W. Liu, and M. S. Hwang, "A study of attribute-based proxy re-encryption scheme in cloud environments", *International Journal of Network Security*, vol. 16, no. 1, pp. 1-13, 2014.

[4] D. C. Edna, O. A. Robson and T. S. J. Rafael, "Trust model for file sharing in cloud computing," in *Proceedings of the Second International Conference on Cloud Computing, GRIDs, and Virtualization*, pp. 66–73, 2011.

[5] D. G. Gopaland R. Saravanan, "Fuzzy based energy aware routing protocol with trustworthiness for MANET," *International Journal of Electronics and Information Engineering*, vol. 3, no. 2, pp. 67–80, 2015.

[6] D. G. Gopaland R. Saravanan, "A novel trust based system to detect the intrusive behavior in MANET," *International Journal of Electronics and Information Engineering*, vol. 3, no. 1, pp. 31–43, 2015.

[7] P. S. Hada, R. Singh and M. M. Meghwal, "Security agents: A mobile agent based trust model for cloud computing," *International Journal of Computer Applications*, pp. 12–15, 2011.

[8] K. Hwang, D. Li, "Trusted cloud computing with secure resources and data coloring," *IEEE Internet Computing*, vol. 14, no. 5, pp. 14–22, 2010.

[9] A. Kumar, K. Gopal and A. Aggarwa, "Design and analysis of lightweight trust mechanism for secret data using lightweight cryptographic primitives in MANETs," *International Journal of Network Security*, vol. 18, no. 1, pp. 1–18, 2016.

[10] X. Y. Li, H. D. Ma, F. Zhou, W. B. Yao, "T-Broker: A trust-aware service brokering scheme for multiple cloud collaborative services," *Information Forensics and Security*, vol. 10, no. 7, pp. 1402–1415, July 2015.

[11] X. Y. Li, L. T. Zhou, Y. Shi, and Y. Guo, "A trusted computing environment model in cloud architecture," in *Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, pp. 11–14, 2010.

[12] G. Y. Lin, D. R. Wang, Y. Y. Bie, M. Lei, "A mutual trust based access control model in cloud computing," *China Communications*, vol. 11, no. 4, pp. 154–162, 2014.

[13] A. Mosa, H. M. El-Bakry, S. M. Abd El-Razek, S. Q. Hasan, "A proposed E-government framework based on cloud service architecture," *International Journal of Electronics and Information Engineering*, vol. 5, no. 2, pp. 93–104, 2016.

[14] T. H. Noor, Q. Sheng, L. Yao, S. Dustdar, A. H. H. Ngu, "CloudArmor: Supporting reputation-based trust management for cloud services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 367–380, 2016.

[15] P. S. Pawar, M. Rajarajan, S. K. Nair, and A. Zisman, "Trust model for optimized cloud services," in *Proceedings of the 6th IFTP International Conference on Trust Management*, pp. 99–112, 2012.

[16] N. Santos, K. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proceedings of Conference on Hot Topics in Cloud Computing (HotCloud'09)*, June 2009.

[17] N. B. Truong, T. W. Um, G. M. Lee , "A reputation and knowledge based trust service platform for trustworthy social internet of things," in *Proceedings of the 19th International ICIN Conference on Innovations in Clouds, Internet and Networks*, pp. 104–111, 2016.

[18] X. Wu, J. He, F. Xu, "An enhanced trust model based on reputation for P2P networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'08)*, pp. 67–73, 2008.

[19] Z. M. Yang, L. X. Qiao, C. Liu, C. Yang, and G. M. Wan, "A collaborative trust model of firewall through based on cloud computing," in *Proceedings of the 14th International Conference on Computer Supported Cooperative Work in Design*, pp. 329–334, 2010.

[20] W. H. Zhang and H. L. Sheng, "An improved trusted cloud computing platform model based on DAA and privacy CA scheme," in *Proceedings of the IEEE International Conference on Computer Application and System Modeling (ICCASM'10)*, 2010.

Biography

Xu Wu biography. received her Ph.D. degree in Computer Science, from the Beijing University of Technology, in 2010. She was out of post-doctoral stations of the MOEKLINNS Lab, Department of Computer Science and Technology of Xian Jiaotong University in 2016. She is an associate professor of Xi'an University of Posts and Telecommunications. She is working as a visiting scholar in School of Engineering and Technology, Indiana University-Purdue University Indianapolis, Indianapolis, USA, when working on this paper. Her research interests include trusted computing, pervasive computing, mobile computing, and software engineering. She has published more than 50 technical papers and books/chapters in the above areas. Her research is supported by National Natural Science Foundation of China (Program No. 71501156 and No.61373116) and China Postdoctoral Science Foundation (Program No.2014M560796) and Shaanxi Postdoctoral Science Foundation and special funding for key discipline construction of general institutions of higher learning from Shanxi province.