PROCEEDINGS OF SCIENCE

A Novel Trust Management Framework for Mobile Cloud Computing Environment

Chaoyang zhu[1]
Information & Communication Department, China Electrical Power Research Institute
Beijing, 100192, China
E-mail: zhucy@epri.sgcc.com.cn

Qing Ding
School of Software Engineering, University of Science and Technology of China
Hefei, 230051, China
E-mail: dingqing@ustc.edu.cn

Xiangzhou Chen
Information & Communication Department, China Electrical Power Research Institute
Beijing, 100192, China
E-mail: chenxiangzhou@epri.sgcc.com.cn

Zhu Shi
School of Software Engineering, University of Science and Technology of China
Hefei, 230051, China
E-mail: shizhu@ustc.edu.cn

With the trend of Mobile Cloud Computing, the establishment of trust between mobile consumers and cloud service providers become the primary issue as it relates to the cloud platform security. Subjective or objective trust model are typically adopted in the Cloud environment; however, none of these models can easily go for MCC because they only cover a few aspects of trust establishment and do not support all the essential features. In this paper, a Service Level Agreements (SLAs) server assisted distributed trust management framework is designed for such networks that enables mobile users to quickly adapt itself to change local conditions. Based on this architecture, an innovative trust model is presented which integrates objective and subjective trust aggregated in a distributed manner. The simulation results demonstrate that this model can be effective to differentiate trustworthy and untrustworthy Cloud Service Providers (CSP) and the service users in the presence of few untrustworthy nodes. In particular, it provides a more effective method to meet the mobile users' individualization requirement.

CENet2017
22-23 July 2017
Shanghai, China

[1]This paper was supported in part by a grant from the Science and Technology Project of State Grid Corporation of China with number XX71-15-036

PoS(CENet2017)070

1. Introduction

Mobile Cloud Computing (MCC) is the extension of the conventional Cloud Computing which introduces mobile clients such as PDA, sensors, etc. In recent years, it is witnessed that the business applications are increasingly moving into MCC platforms [1-3]. In this open environment, security has become the primary concern. Basically, the trust between users in cloud environment is usually based on a trusted third party such as SLAs and certification, etc. to take a set of assertion on a given target. But the establishment of trust between mobile consumers and cloud service providers is more challenging because of the complexity of introducing mobile users.

In 1996 Blaze et al. proposed the concept of trust management, then adopted it into the security mechanism of the distributed system [4]. A Survey of Trust and Trust Management in Cloud Computing can be found in one paper[5]. Some researchers concentrated on the design of trust management framework suitable for Cloud environment. In the paper[6], Noor et al. proposed a framework for analyzing trust management systems in Cloud environment to help researchers evaluate various solutions. Bharathi et al. proposed an extended trust management scheme in the Cloud which takes advantage of the user profiles, especially the location information of the user to avoid services from targeting by malicious users[7]. Fan et al. addressed the problem of trust management in multi-cloud environments[8]. Kim presented a trust management approach by analyzing users' telephone call data. This inter-user trust relationship was integrated in MCC [9]. A trust management system was proposed by Hammam, which considered resource availability, neighbors' evaluation and response quality and task completeness in calculating the trust value based on the EigenTrust algorithm [10]. Commonly, the trust model can be classified as objective and subjective based on the measuring method. According to the way that the trust information is gathered, it can be classified as local trust and global trust. Xia et al. built a subjective trust model called AFStrust in mobile ad hoc network [11]. The salient features of this model is multiple decision factors, including direct trust, recommendation trust, etc. A trust management protocol called SQTrust was presented to yield peer-to-peer subjective trust evaluation [12]. There's been comparatively few researches concerning the combination of objective and subjective trust. Tan et al. proposed a mixed model to evaluate E-Learning services[13]. Tong et al. studied the relationship between objective trust and subjective trust [14]. Fan also proposed a mixed model in Cloud environment [8].

However, considering the complexity of the MCC, the simple integration of objective and subjective trust cannot meet various types of the users' demand, such as the roaming user. In this paper, a mixed trust model is presented which includes the subjective trust and objective trust to evaluate the trustworthiness of a CSP and a Mobile User (MU). The SLAs server (SLAS) are regarded as the infrastructure to achieve a global trust system. The subjective trust model is mainly based on feedback information received from the MUs and CSPs' administrators, and the objective trust model is based on the record from SLAs.

The paper is organized as follows. The SLAS-based federal trust management framework is presented in Section 2. The trust computing model is proposed in Section 3. Section4 shows the simulation results. The paper is concluded in Section 5.

2.A SLAS-based Federal Trust Management Framework

In this paper, a multi-Cloud environment with roaming users is considered. There are three important roles in the system: SLAS, CSP, and MU as shown in Fig. 1. We give the detailed information about core components respectively below.

SLAS has five core components: SLA Register, SLA Templete Library, SLA Runtime Monitor, Trust Register and Trust Calculator.

a. SLA Register: CSPs register their SLA parameters in it, and MU query it to get SLA Info.

b. SLA Runtime Monitor: It gets service runtime parameters by contacting Monitor Agent running on CSP after the service is accessed.

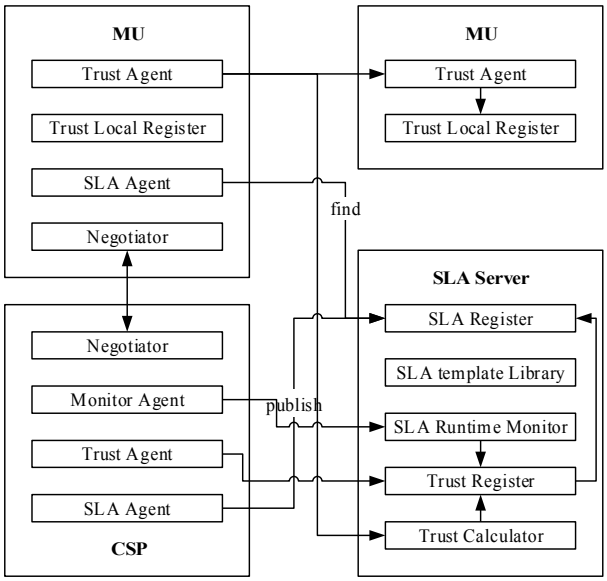

Figure 1: Core Components

c. SLA Templete Library: it provides a reusable-component repository for SLA.

d. Trust Register: it is responsible for providing the centralized storage capacity for MUs and CSPs.

e. Trust Calculator: it calculates the global object trust value of CSPs based on Equation 9 discussed in next Session.

MU has four core components: Trust Agent, Trust Local Register, SLA Agent, and Negotiator.

a. Trust Agent: one the task of it is to query the Trust Calculator in SLAS to get the global object trust value of CSPs, another is to connect Trust Agents in other MUs to get the local subjective trust value of CSPs. Then the mixed trust value of CSPs are calculated based on them.

b. Trust Local Register: it is used for a MU to store local subjective trust value for some CSPs based its own obersevation.

c. SLA Agent: it connets SLA Register in SLAS to find SLA Templetes and registers its own file.

d. Negotiator: it is used to negotiate the concrete SLA parameters between MUs and CSPs.

CSP has also four core components: SLA Agent, Trust Agent, Monitor Agent and Negotiator.

a. SLA Agent: it connets SLA Register in SLAS to query SLA file of specific CSPs.

b. Negotiator: it is used to negotiate the concrete SLA parameters between MUs and CSPs.

Trust Agent: it is used to query the Trust Calculator in SLAS to get the global subjective trust value of Mus.

```
Algorithm 1: CSP Selection
Function cspSelection()
   Send message to SLAS
   broadcast message in
   start a timer
   collect all feedbacks
   computes the trust value
EndFunction
```

Figure 2: Algorithm 1

d. Monitor Agent: it is used to monitor the running parameters of CSP after the service is implemented and report these information to SLA Runtime Monitor in SLAS.

The actual ways of doing the work are as shown in Algorithm 1. When a user wants to request for a Cloud Service (CS), he/she shall first send a trust query message to a SLAS near his location, which is broadcasted in its community simultaneously. The broadcasting request in the community is an essential supplementation because some users might not be willing to denounce their true thoughts in SLAS for some reason. However, they are usually not opposed to reply to the request privately. Once the SLAS gets this query, it will start a timer and forward this query to other SLASs. Then this SLAS will collect all feedbacks from other SLASs within a predefined interval and compute the trust value of this CSP by Equation 8. The user gets the reply from this SLAS and collects all feedbacks in the community, then computes the trust value of this CSP by Equation 9. If the user decides to access this CS, he will send the resource request to this CSP, and then this CSP requests the trust value of this user from the SLAS. The SLAS will compute it based on the historical record of this user. If this user is considered as trustworthy, he or she will be allowed to invoke the service.

3.Trust Model

In this Section, the mixed trust model is presented. In our design, the objective trust model is based on observations that how big the difference is available between the predefined service agreement and the actual results, as reported from SLASs. The subjective trust model is computed based on feedback information received from the MUs, and CSPs after services are accessed. We will give the details as below.

Different cloud services and deployment models will require different approaches to SLAs, adding to the complexity of SLAs. For example, a list of important criteria may be included, such as availability, performance, disaster recovery expectations, location of the data, portability of the data and privacy of the data, etc. Thus the objective trustworthiness of a CSP should be related to all these parameters associated with SLAs. Let $P^m_s = (p^1_s, p^2_s, p^3_s, \ldots\ldots, p^m_s)$ be the set of parameters included in SLA of services is computed, where m is the total number of parameters. The objective trustworthiness of a CSPs from user j is computed as Equation 1.

$$OS^m_{sj} = (s^1_{sj}, s^2_{sj}, s^3_{sj}, \ldots\ldots, s^m_{sj}) \tag{3.1}$$

where S^i_{sj} is the i-th dimension component of *m*-dimensional vector OS^m_{sj}, $S^i_{sj} \in [0,1]$. The weight of each component is defined as Equation 2.

$$\omega = (\omega^1, \omega^2, \omega^3, \ldots, \omega^m), \sum_{i=1}^{m} \omega^i = 1 \tag{3.2}$$

After the Cloud service is accessed, MU and CSP evaluate each other. Let $q^n_u = (q^1_u, q^2_u, q^3_u, \ldots\ldots, q^n_u)$ be the set of parameters which should be considered, where n is the total number of parameters. The subject trust value of user j can be expressed as Equation 3.

$$SU^n_{js} = (u^1_{js}, u^2_{js}, u^3_{js}, \ldots\ldots, u^n_{js}) \tag{3.3}$$

where u^i_{js} is the i-th dimension component of n-dimensional vector SU^n_{js}, $u^j_{js} \in [0,1]$. The weight of each component is defined as Equation 4.

$$\varphi^n = (\varphi^1, \varphi^2, \varphi^3, \ldots, \varphi^n), \sum_{i=1}^{n} \varphi^i = 1 \tag{3.4}$$

The subject trust value of CSP s can be expressed as Equation 5.

$$SS^m_{sj} = (c^1_{sj}, c^2_{sj}, c^3_{sj}, \ldots\ldots c^m_{sj}), \tag{3.5}$$

where c^i_{sj} is the i-th dimension component of m-dimensional vector SS^m_{sj}, $c^i_{sj} \in [0,1]$. The weight of each component is defined as Equation 6.

$$\phi^m = (\phi^1, \phi^2, \phi^3, \ldots, \phi^n), \sum_{i=1}^{m} \phi^i = 1 \tag{3.6}$$

When MU j wants to evaluate the trust of CSPs, it broadcasts a recommendation query message in the community. Then every user in this community can return a recommendation message which includes the CSP he recommends. MU j collects all these recommendations, and selects top 5 CSPs with highest numbers. Then, MU j sends the trust query message for every CSP in this list L to a near SLAS k, SLAS k is responsible for calculating the global trust value of CSP s from his view at time t_n based on Equation 7.

$$G^m_{skt_n} = \beta\, GS^m_{skt_n} + (1-\beta)\, GO^m_{skt_n} \tag{3.7}$$

where $GS^m_{skt_n}$ is the global subject trust value of CSP s from SLAS k, which is calculated by Equation 8 at time t_n. $GO^m_{skt_n}$ is the global object trust value of CSP s from SLAS k, which is calculated by Equation 9 at time t_n. β is the weight, which is used to define the proportion of global subject trust value in the total value.

$$GS^m_{skt_n} = \gamma\, GS^m_{skt_{n-1}} + (1-\gamma)\left(\sum_{i=1}^{N} \frac{GU^l_{ikt_n}}{\sum_{j=1}^{n} GU^l_{jkt_n}} * SS^m_{skit_n}\right) \tag{3.8}$$

$$GO^m_{skt_n} = \gamma\, GO^m_{skt_{n-1}} + (1-\gamma)\left(\frac{\sum_{i=1}^{N} OS^m_{skit_n}}{N}\right) \tag{3.9}$$

where $GS^m_{skt_{n-1}}$ is the historical value of s' subject trust at time window t_{n-1}, and $SS^m_{skit_n}$ is the subject trust value of CSP s SLAS k receives from user i at time window t_n. $GU^l_{ikt_n}$ is the global subject trust value of user i SLAS k stores at time window t_n, which is shown in Eq. 11. γ is the historical weight. n is the number of received evaluation by SLAS k at time window t_n. $GO^m_{skt_n}$ is the historical value of s' object trust at time window t_{n-1}, and $OS^m_{skit_n}$ is the object trust value of CSP s SLAS k receives from user i at the time window t_n. When SLAS k receives this request, he will forward it to other SLASs to get more data about CSP s. In this case, the final value returned to MU j is achieved by Eq. 10. Here we assume all SLASs are trustworthy and they trust each other.

$$G^{m}_{st_n}=\tau\, G^{m}_{skt_n}+(1-\tau)(\frac{\sum_{i=1}^{M} G^{m}_{sit_n}}{N}) \tag{3.10}$$

where M is the number of all messages received from other SLASs within a predefined time interval, and τ is the weight. Finally, MU *j* select a suitable CSP based on an overall consideration of the recommend number of CSPs and the trust value of them.

After MU *j* submits a service request, the CSP *s* will also check the trust value MU *j* to judge if it can run this service. The computing process is similar and is shown below.

$$GU^{l}_{jkt_n}=\mu\, GU^{l}_{jkt_{n-1}}+(1-\mu)(\sum_{i=1}^{N}\frac{GS^{m}_{ikt_n}}{\sum_{j=1}^{N} GS^{m}_{jkt_n}} * SU^{l}_{jkit_n}) \tag{3.11}$$

$$GU^{l}_{jt_n}=\zeta\, GU^{l}_{jkt_n}+(1-\zeta)(\frac{\sum_{i=1}^{M} GU^{l}_{jit_n}}{M}) \tag{3.12}$$

In order to combine separate pieces of trust information calculated above to calculate the probability that an event is true or false, we adopt Dempster-Shafer evidence theory. Then we can get that for event α_i from node N_j which is observed by node N_k. Its belief function is defined as Equation 13 and plausibility function is defined as Equation 14, here $m:2^{x}\to[0,1]$. The belief function for event α_i is shown in Equation 15 and Equation 16.

$$bel_{N_k}(\alpha_i)=\sum_{e:\alpha_e\in\alpha_i} m_{N_k}(\alpha_e)=m_{N_k}(\alpha_i) \tag{3.13}$$

$$pls_{N_k}(\alpha_i)=1-bel_{N_k}(\bar{\alpha}_i) \tag{3.14}$$

$$bel(\alpha_i)=m(\alpha_i)=\bigoplus_{k=1}^{K} m_{N_k}(\alpha_i) \tag{3.15}$$

$$m_{N_1}(\alpha_i)\oplus m_{N_2}(\alpha_i)=\frac{\sum_{q,r:\alpha_q\cap\alpha_r=N_j} m_{N_1}(\alpha_q)m_{N_1}(\alpha_r)}{1-\sum_{q,r:\alpha_q\cap\alpha_r=N_j} m_{N_1}(\alpha_q)m_{N_1}(\alpha_r)} \tag{3.16}$$

4.Simulation Experiments

In order to evaluate our proposed trust model, a SLASs aided multi-Cloud environment with multiple MUs is simulated. All SLASs are assumed to be implemented in trusted hardware and thus trustworthy. MUs can be trustworthy or untrustworthy. The trustworthy MUs provide the trustworthy feedback for CSP at most times, and the untrustworthy MUs provide untrustworthy evaluation and give false recommendation at most times. We simulate 1000 MUs, of which 90% are trustworthy and 10% are untrustworthy. It assumes that each CSP provides one or more services on the same cloud, who also can be trustworthy or untrustworthy. The trustworthy CSPs take the majority successful interactions at each time and give their trustworthy feedback for MUs. Untrustworthy CSPs take the minority successful interactions at each time and give their untrustworthy feedback for MUs. In the simulation environment, the number of CSPs is 20 and 18 of them are trustworthy. All simulations are carried for 100 times. In each simulation, MUs initiate a request to a RSP randomly and the feedback is given by both sides after transaction. The results of mutual evaluation between the two sides are as Table 1.

	Trustworthy rating	**Untrustworthy rating**
TCSP	[0.8, 1]	[0, 0.3]
UCSP	[0, 0.3]	[0.8, 1]
TMU	[0.8, 1]	[0, 0.3]
UMU	[0, 0.3]	[0.8, 1]

Table 1: Results of Mutual Evaluation

The evaluation results for different types of CSPs are shown in Figure 3 (a) and (b). Here, the TMU is the result from trustworthy MUs' evaluation to CSPs, the UMU is the result from untrustworthy MUs' evaluation to CSPs, and the total is the result from all MUs' evaluation to CSPs. The experiment results show that the trustworthy CSPs feature relatively higher trustworthiness values and the untrustworthy CSPs feature relatively lower trustworthiness if there are only a few of untrustworthy MUs in the system. In Figure 4, we show the results for MUs from CSPs' feedback. Here, the TMU is the trustworthy MUs' score, and the UMU is the untrustworthy MUs' score. We also can get the conclusion that malicious users can be prevented from accessing the system by our approach.

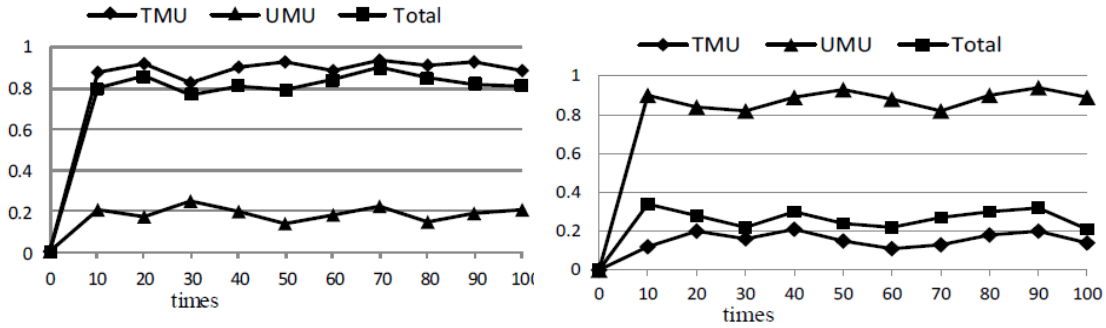

Figure 3(a): Evaluation results for trustworthy CSPs **(b):** Evaluation results for untrustworthy CSPs

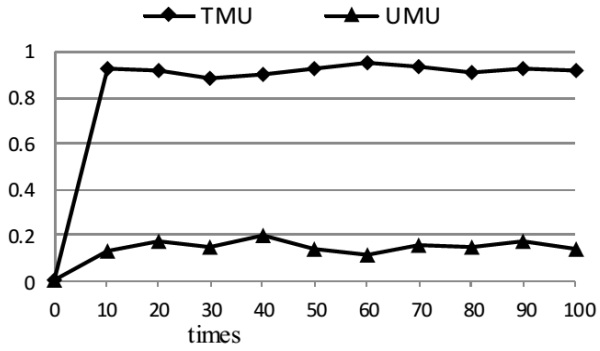

Figure 4: Evaluation results for MUs

In the second experiment, we set 1000 MUs, of which 50% donot give any feedback after they access the service and 50% in the rest always give habitual praise to any service provider regardless of its QOS, namely, the trust value 0.9. But when a MU broadcasts a recommendation query message in the community, all receivers should reply sincerely. It is assumed that each service request is assigned with 10-20 neighbors randomly, who maybe

untrusted. Each neighbor recommends a candidate based on his opinion. Figure 5(a) shows the cumulative number that the untrusted CSP is chosen when the neighbors' advices are not accepted. Instead, the cumulative number when neighbors' advices are accepted is shown in Figure 5(b). It is obvious that the neighbors play a very important role in meeting the users' "localization" feature at this scene.

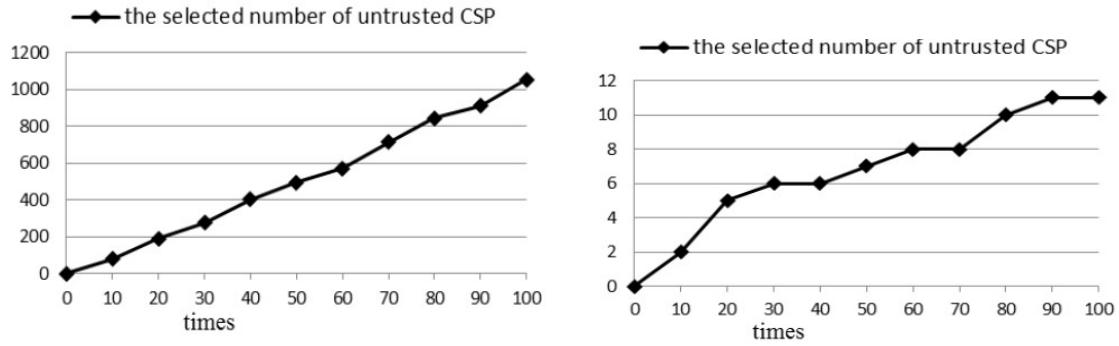

Figure 5(a): Neighbors' advice are not accepted **(b):** Neighbors' advice are accepted

5.Conclusion

In this paper, we propsed a novel trust management framework for a MCC environment to effectively evaluate the trustworthiness of CSPs and MUs by using the mixed trust model. We proposed a two-level solution where the service requester first asked the community users to recommend their suitable potential service providers, then selected the most suitable one by acquiring these CSPs' global trust value. Besides, the CSPs also can decide whether the service should be provided by evaluating MU's trustworthiness. Trustworthiness is calculated by integrating objective and subjective trust. The this mixed model primarily highlights that the objective and subjective trust are represented by different approaches from the perspective of trust management. The objective trust model is based on the comparison between predefined SLA and actual implementation results. The objective trust of an entity is considered as a global trust and computed by integrating direct information and the second-hand information. The subjective trust model is calculated based on feedback information received from the MUs and CSPs. The subjective trust value of an entity is computed from local direct information. The experiments show that the proposed solution is effective and robust for both CSPs and MUs to evaluate each other's trustworthiness at the presence of few untrustworthy nodes. Particularly, it provides a more effective method to fulfill roaming users' individualization requirement.

References

[1]Ruan K, *Cloud forensics definitions and critical criteria for cloud forensic capability: an overview of survey results* [J], Digital Investigation, Volume 10, Issue 1, pp. 34–43, 2013.

[2]Hoang T. Dinh, Chonho Lee, Dusit Niyato, Ping Wang, *A survey of mobile cloud computing: architecture, applications, and approaches* [J], Wireless Computations and Mobile Computing, pp.1587–1611, 2013.

[3]Niroshinie Fernando, Seng W. Loke, Wenny Rahayu, *Mobile cloud computing: A survey* [J], Future Generation Computer Systems, Volume 29, pp. 84–106, 2013.

[4]M Blaze, J Feigenbaum, J Lacy, *Decentralized Trust Management* [C], Proceedings of IEEE Conference security & Privacy, (1996) 30(1), pp.164-173.

[5]V Kumar, S Madria, B Kashyap, M Mohania, A Survey of Trust and Trust Management in Cloud Computing[M], *Managing Trust in Cyberspace*, Chapter: 3, pp.41 – 69, 2013.

[6]T. H. Noor, Q. Z. Sheng, Z. Maamar, S. Zeadally, *Managing Trust in the Cloud: State of the Art and Research Challenges*[J], Computer, Volume49 Issue2, pp.34 – 45, 2016.

[7]C Bharathi, V Vijayakumar, KV Pradeep, *An Extended Trust Management Scheme for Location Based Real-time Service Composition in Secure Cloud Computing*[J], Procedia Computer Science, 50, pp. 103-108, 2015.

[8]Wenjuan Fan, Harry Perros, *A novel trust management framework for multi-cloud environments based on trust service providers*[J], Knowledge-Based Systems, 70, pp. 392–406, 2014.

[9]Mucheol Kim, Sang Oh Park, Trust management on user behavioral patterns for a mobile cloud computing[J], Cluster Computing, Volume 16, Issue 4, pp. 725-731, December 2013.

[10]A Hammam, S Senbel, *A trust management system for ad-hoc mobile clouds*[C], International Conference on Computer Engineering & Systems, (2013) 8255(1), pp. 31-38.

[11]Hui Xia, Zhiping Jia, Lei Ju1, Xin Li, Youqin Zhu. *A Subjective Trust Management Model with Multiple Decision Factors for MANET based on AHP and Fuzzy Logic Rules*[C], IEEE/ACM International Conference on Green Computing and Communications, (2011) pp. 124-130.

[12]Ing-Ray Chen, Jia Guo, Fenye Bao, Jin-Hee Cho, *Trust Management in Mobile Ad Hoc Networks for Bias Minimization and Application Performance Maximization*[J], Ad Hoc Networks, Volume 19, pp. 59–74, August 2014.

[13]Wenan Tan, Jingxian Li, Anqiong Tang, Tong Wang, Xiaoming Hu, *Trust Evaluation Model Based on User Trust Cloud and User Capability in E-Learning Service*[M], Communications and Information Processing Volume 288 of the series Communications in Computer and Information Science, pp. 583–590.

[14]Tong, X., Zhang W., Yu, L., and Huang, H., *Subjectivity and Objectivity of Trust*[C], Proceedings of Agents and Data Mining Interaction, 2013, pp.105-114