I. J. Computer Network and Information Security, **2017, 2, 36-42**
Published Online February 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2017.02.05

An Effective Way of Evaluating Trust in Inter-cloud Computing

Kiran Mary Matthew
Department of Computer Science and Engineering, VIT University, Chennai
E-mail: kiranmary.matthew2015@vit.ac.in

Prof. Abdul Quadir Md
Department of Computer Science and Engineering, VIT University, Chennai
E-mail: abdulquadir.md@vit.ac.in

Abstract—For any communication to be successful, trust is necessary. For inter-cloud communication, clouds interact with each other for resource sharing. Since they are unaware of their opposite party, there should be some mechanism by which the cloud gets an idea about them prior to the communication. This is accomplished through trust management systems that calculate the trust rating of clouds from opinions from their peers. There is no way to ensure whether these peers are genuine in their opinion or not. This paper proposes a method to reduce such problems by considering the latest history of communication of a particular cloud and ignore the opinions given by less trusted clouds.

Index Terms—Inter-cloud computing, trust management, trust rating.

I. Introduction

Inter-cloud computing helps in sharing of resources and solving complex problems. Choosing the right cloud to work with, is a challenging task. For this, we consider the trustworthiness of a cloud. The word trust refers to the confidence to rely on something. In cloud computing, trust is indispensable for inter cloud computing. Since autonomous clouds have to peer up for interaction, it is important to ensure whether a particular cloud is trustworthy or not. Existing mechanism rates a cloud based on the experiences of other clouds with which it has done transactions. A decision is made by cumulating all their opinions.

To prevent the opinion of dishonest clouds from being considered, the probability distribution function of all the opinions is calculated and the extremes are excluded. This doesn't work in a case where the number of fake opinions is more. This paper proposes a different concept of trust calculation using the most recent interactions within a given time span, t. Trust rating will be more accurate when the recent opinions are reviewed.

The rest of the paper is organized in to the following subparts: Section II deals with the problem statement and related works. Section III includes the proposed architecture. Section IV gives the conclusion.

II. Related Works

To determine the trustworthiness of a cloud, the confidence level of peer clouds about the particular cloud's behaviour is collected to calculate the reputation rating. Honesty rating is calculated based on the opinion about a cloud when considering it as a source of information. For a more accurate rating, this paper proposes a concept of considering only the recent opinions within a time span t.

Jemal Abawajy in [1] proposes a reputation-based trust management system for inter-cloud computing. Most of the communication between the user and the cloud takes place without the user having an actual idea about the cloud party's behaviour. This scenario can be dealt with by using trust rating of that cloud that can be obtained from other clouds with which it has interacted. This model allow users to select high-quality cloud services using trustworthiness of a given resource for executing their job by satisfying user's QoS requirements.

The clouds that lie are handled by using an honesty level threshold to decide whether to consider their opinions or not. There is a grid resource manager who is responsible for provisioning and allocation of resources. If the given requests cannot be fulfilled, it is forwarded to the inter-cloud resource manager. Reputation manager is responsible for maintaining the trust information of all the clouds. He uses three ratings to build the reputation of an entity:

- Personal experience: firsthand information of a grid after transaction with another grid
- Reputation rating: confidence about a grid's behaviour as a good service provider
- Honesty rating: measure of how trustful a grid is, as a provider of second hand information.

Josang, Audun, Roslan Ismail and Colin Boyd in [2] describe the concept of trust and reputation in detail. Trust is the extent to which one party is willing to depend on something or somebody in a given situation with relative security, even though negative consequences are possible. Reputation is something that is generally said or believed about a person's or thing's character or standing.

Reputation system can be centralized or distributed.

Reputation computation engines use various techniques for reputation and trust calculation methods like simple summation, average of ratings, probability density function, discrete models, flow models etc. It also discusses some of the well-known applications using online reputation system.

Habib, Sheikh Mahbub, Sebastian Ries, and Max Muhlhauser in [3] explain a multi-faceted trust management system that helps to distinguish between a trusted and untrusted cloud provider. The flexibility of this system lies in the fact that the customer can choose the attributes based on which the trust rating should be done. Cloud services register to the registration manager to be a part of the cloud marketplace. They fill in a questionnaire regarding their service specifications as part of the agreed upon policy. Trust manager is responsible for providing the front end whether the user specifies his requirements.

Noor and Talal H in [4] give an elaborate idea on trust management. It discusses the various trust management techniques based on policies, reputation, recommendation and prediction. Policies include rules such as setting thresholds and limiting the access based on the type of user. Reputation is calculated based on opinions from various cloud providers.

Recommendation makes use of a cloud provider's knowledge about trusted parties that can be utilized by the cloud provider having little knowledge. Prediction is used when there is no knowledge about a particular cloud provider. Some of the open issues in trust management are identification, privacy, personalization, integration, security and scalability. Various research prototypes have also been discussed.

Jemal Abawajy in [5] proposes a trust management system for hybrid clouds. Personal experience of each cloud about a service provider is calculated and shared among other clouds through peer-to-peer communication. To check for the authenticity of the opinions, probability density function of all the opinions is found out. The opinions on the either extremes are the ones that are different from others and hence the fake opinions. This architecture provides a better reliability.

Vijayakumar V and R. S. D. W. Banu in [7] deal with the scheduling of incoming jobs to available resource sites based on the trust factor value. The management of resources in grids are challenging due to:

(a) Geographical distribution of resources and their heterogeneity
(b) Difference in resource policies and practices among autonomous grids
(c) Grids using different access and cost models

The Trust Factor (TF) value of each resource site has estimated by its protection capability and weightage of reputation acquired through the feedback from its past behavior from the user community. The protection capability of a site includes its ability to detect threats like intrusions, viruses, unauthorized access and secure file storage and job completing abilities.

At first, the users submit their jobs to a Grid Organization Manager GOM. It will calculate the trust factor value of all the entities based on the capability of protection and weightage of reputation. A high trust factor valued entity is selected for the execution of current job. GOM will inform the user about the entity selected. On the completion of the job, the user is asked to provide feedback about the entity on some security attributes.

Bonatti and Piero in [8] integrate the concept of both policy based and reputation-based trust management approaches having more enhanced properties than the previous methods. This concept is applicable to both structured and unstructured user environments. It makes use of the policy language, PROTUNE. Here a modular approach is followed where the distribution of basic facts on risk and reputation and their computation are delegated to several external packages.

Kotsovinos Evangelos and Aled Williams in [9] describe BambooTrust, which is a scalable and distributed trust management system. It is based on an existing model named XenoTrust and the Bamboo distributed hash table. Instead of using queries to access the necessary information, user defines a set of rules to decide on how to accumulate the reputation information collected from different parties. The system operation is as follows:

- BambooTrust users submit statements or rule-sets to any BambooTrust node
- Statements and rule-sets get routed to the node that is responsible for storing them
- Those nodes perform periodic evaluation of rule-sets in them
- If the result suggests that the user should be notified then a message is sent to him asynchronously. These messages are time stamped to avoid replay attacks

Kagal, Lalana, Tim Finin, and Yun Peng in [10] bring in the idea of delegation of authorization for the access to specific resources. Here permissions are modeled as the rights of an agent. These rights are associated with actions. Security agent in that domain checks whether the delegated rights are authorized or not.

The user is permitted only when all the credentials have been verified. This forms a delegation chain. If any agent in this chain fails to meet the requirements of the delegated right, the chain is broken and all the agents following the failure are not allowed to perform the action associated with that right. They have used two different models for their study: home automation model and electronic supply chain management model.

Azzedin Farag and Muthucumaru Maheswaran in [11] deal with a peer to peer brokering system that models accuracy and honesty concepts for clouds separately. Here trust level is calculated on a scale of 1 to 5 based on the past experiences. This model assumes that each cloud has its own set of recommenders and trusted peers.

The list of recommenders is maintained in a recommender trust table. Each broker is responsible for

all the resources in its domain. Accuracy concept is used to enable peer review-based mechanisms to function with imprecise trust metrics. Honesty concept is used to reduce model's sensitivity towards dishonest domains. This system works even when the number of malicious recommenders is high.

Carbo, Javier, Jose M. Molina, and Jorge Davila in [12] propose a trust management system that uses fuzzy sets to store the trust information about others. Reputation is updated from time to time based on the general scenario. This system is useful especially in the case of merchants and buyers where the trust factor changes rapidly. The six steps involved in the shopping process are:

- Prediction of user need
- Characterization of desired product
- Choosing the right merchant
- Agreement negotiation
- Payment and delivery
- Rating of the service

It has good prediction capabilities and robustness. Trust management through fuzzy reputation has high prediction rates and robustness against manipulation. But it is totally dependent on the linguistic fuzzy sets received as input.

Noor, Talal H., and Quan Z. Sheng in [13] propose a framework for Trust-As-A-Service. Some of the issues in trust management like trust rating accuracy and feedback storage are discussed in detail. Service oriented architecture is used for the delivering of this service. This framework has three layers:

- Cloud service provider
- Cloud service consumer
- Trust management service

The cloud service provider layer provides infrastructure as a service. The trust management service layer consists of a number of distributes trust management nodes. There is also a registry system that is responsible for the advertisement of service, its discovery and registration. Implementation is done based on NetLogo platform.

An adaptive credibility model is introduced to assess the trustworthiness of the cloud service and to distinguish between credible and malicious feedbacks. Evaluation of this model is done by using both analytical and empirical analysis. The capability and majority consensus factors provided by the consumers are used to calculate the trust of a cloud service.

Weeks Stephen in [14] defines a framework for expressing trust management system. This framework consists of the following components:

- Principals
- Authorization
- Licenses
- Assertions
- Authorization maps

Principals should satisfy the property that they are distinguishable. Authorization includes the permissions of the principals. Authorization map is the mapping between the principals and the authorization.

Wei, Fan, Chen Ahmed and Pathan in [15] propose on developing a general Subspace based MAlicious peeRs deTecting algorithm (SMART). It is based on Multiscale Principal Component Analysis (MSPCA) and control chart. Most of the detection algorithms either focus on malicious peers of particular categories or use global assumptions.

But SMART is based on reputation information alone. It reconstructs the original reputation matrix based on subspace method and find malicious peers based on Shewhart control chart. Simulations have shown that it is good at finding malicious peers with mixed behaviour. Its effectiveness is less if the malicious peers are able to guess the reputation pattern of honest peers.

Seyyed and Parisa in [16] discuss about the security, privacy and trust challenges in the cloud computing environment. Trust is defined as the confidence that an entity behaves in expected ways. Trust can be human to human, machine to machine, human to machine or machine to human.

Various technical, operational and legal challenges of trust have also been discussed. Compromised virtual machines, interoperability in grid technology, availability of resources and trust across distributed environments are some of the technical challenges. Operational challenges include complexity in ensuring compliance with data protection rules, operation failure and unauthorized use of information.

Examples of legal challenges are accountability, confidence in security and consumer protection rules. Some of the recommendations to deal with such challenges are:

- National and international laws
- Operational transparency
- Detection of emerging threats
- Localization
- Personal encryption methods
- Establishment of international organization for cloud crimes
- Policies in distributed environment
- Increasing levels of data security

In [17], Derahman, Abdullah and Azmi suggest using a rate of change factor along with final trust value to make it more immune against fake reputation feedbacks in a federated cloud environment. Sybil attacks are also dealt with here. These attacks use multiple fake identities to corrupt the final trust value through false feedbacks.

They introduce a trust node that acts a broker between cloud service providers and customers. It runs all the trust manager services. If the corresponding broker fails, service request can be forwarded to other neighbouring brokers. Broker assign session key to intermediate nodes which is then given to nodes of other cloud networks using Quantum Key Distribution (QKD) protocol.

Trust level of a node is measured using reputation feedback. This feedback is collected from consumers based on their experience. Reputation in increased by one for a good service and decremented by one otherwise. Reputation of a cloud is 50% customer's contribution and the rest, by other cloud service providers.

Xu, Jiuyun, et al in [18] propose a trust management framework to have different reputation for different user groups through local reputation. This is done by calculating similarity among customers using a decision tree to accumulate the feedback. This result is further filtered using time decay factor.

Fatima and Belabbes in [19] propose a trust model based on QoS and CertainTrust model. Trust value on a cloud provider is represented using two factors ie. trust value and performance value. Ina CertainTrust model, trustworthiness of a service is given as the belief that a proposition is true.

An opinion is represented as a triplet of values including average rating, certainty and initial expectation. The user first send request and the initial global trust value is calculated. Transaction is approved only if this value is above the threshold. This value is then updated using the performance and trust values.

Whenever the value becomes less than the threshold, the transaction is stopped. Experiments prove that this method provides more accurate results in comparison to other existing solutions.

III. Architecture of the Proposed Trust Management Scheme

For a trust management system to be successful in an inter-cloud environment, the participation of other clouds is inevitable. These interactions can be used to have a good idea of whether to enter in to communication with it or not. Since collecting opinions can bring in the delay factor, only the recent interactions are evaluated.

Figure 2 shows the proposed architecture of the system. There is a large cloud set S which contains smaller subsets of clouds,

$$S_1, S_2, S_3, \ldots, S_n$$

The three components are:

(1) Global trust manager: It keeps track of all the local trust managers. It maintains a faulty list which consists of ids of clouds whose trust factor is less than three. This list is updated periodically. When a local manager goes down, it chooses another cloud from the same subset with the highest id as the manager.

(2) Local trust manager: He is responsible of each subset. It stores all the details of the clouds belonging to its subset like id, type of service and trust factor. It is responsible for calculating the trust factor of the clouds under it using the technique of boxplot and eliminates the outliners. This procedure is done periodically so that the trust value never becomes stale.

[6] Boxplot is a graphical method of grouping data based on their quartiles. Quartiles divide the ratings in to four quarters. First of all, the median is calculated. For this, all the ratings are arranged in ascending order.

If the count of ratings is even, then the median is the mean of the middle terms.

If it is odd, the middle term is the median. For example, consider two sets of values:

3, 7, 8, 10

Since the number of values is even, the median will be the mean of the middle terms

$$ie. \frac{7+8}{2} = 7.5$$

1, 2, 4, 6, 9

Since the number of values is odd, the median will be the middle term, i.e. 4

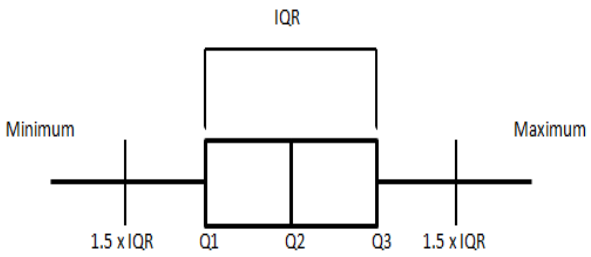

Fig.1. Boxplot

Figure 1 gives the diagrammatic representation of a boxplot. Minimum and maximum denote the minimum and maximum values in the set of values respectively.

There are three quartiles: first quartile Q1 is the mid value between the smallest value and the median. The second quartile Q2 is the median value. The third quartile Q3 is the mid value between the median and the highest value.

Interquartile range (IQR) is calculated as:

$$IQR = Q3 - Q1 \quad (1)$$

Outliners are the values which comes before 1.5 x IQR of the first quartile and after 1.5 x IQR of the third quartile. These values are neglected since their dispersion from the median is very high.

The average of the remaining values is calculated to get the final trust factor r.

$$a = \frac{r_1 + r_2 + r_3 + \cdots + r_x}{x} \quad (2)$$

(3) Clouds: Each of them has a unique id and trust factor. They have a log system that keeps the history of all its communication along with the timestamp.

After interaction with another cloud, a cloud can rate its performance on a scale from 1 to 10 and store it in the log. Local trust manager makes use of these values to calculate the final trust factor.

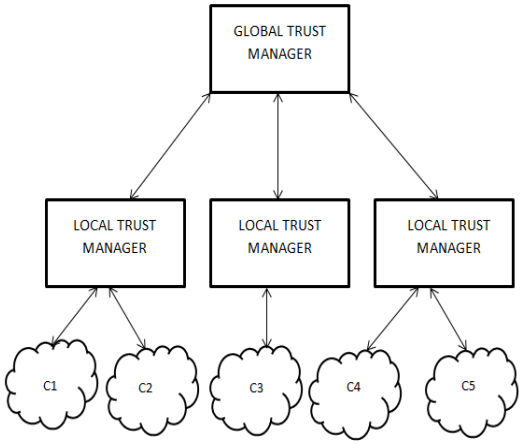

Fig.2. Proposed Architecture for Trust Management

Proposed algorithm for trust management

1. Algorithm TrustMng
2. INPUT: Log history of cloud C1,faulty list, time span t
3. BEGIN:
4. FOR each log entry whose timestamp Ti comes within t
5. FOR each cloud Di
6. DO
7. IF(Di in faulty list) THEN Move to the next entry
8. ELSE

   Extract the corresponding trust rating r_i and store in list L

9. List L sorted in ascending order
10. Boxplot generation IQR = Q3 - Q1

    Q2 = median
    FOR each r_i
    {IF (r_i < 1.5 x IQR OR r_i > 1.5 x IQR)
    r_i marked as outliner list
    END FOR
    }

11. FOR r_i not in the outliner list Calculate the average a
12. END FOR
13. END FOR
14. END FOR
15. RETURN the trust factor a
16. END TrustMng

Fig.3. Proposed Algorithm for Trust Manage

Figure 3 gives the proposed algorithm for trust management, TrustMng. Suppose the trust factor of C1 is to be calculated. Its log history, the faulty list and time span t are given as inputs.

All the logs whose timestamp Ti is within this time span are selected. From that, clouds not included in the faulty list are taken. The trust ratings in their log are extracted and stored in a list L.

This is then sorted in ascending order. Both the median and the IQR are calculated as in (1). On the elimination of the outliners, the average of the remaining ratings is taken to obtain the final trust factor of C1. This value is forwarded to the corresponding local trust manager. Trust factor with a value less than three is recorded in the faulty list.

IV. Experimental Results

This method of trust factor calculation can be demonstrated using a set of sample values. ABCloud is one of the cloud service providers that provide software-as-a-service (SaaS). The local trust manager associated with this cloud goes through the log details of this cloud and identifies the clouds with which it communicated during the last time span t.

The details of these clouds are sent to the global trust manager. It is responsible for trust rating collection of those clouds for ABCloud through the corresponding local trust managers. It then returns these values to the local trust manager who requested it.

Given below are the set of those values collected from the fourteen parties it had communicated with:

10 , 5 , 2 , 4 , 1 , 1 , 4 , 2 , 3 , 2 , 4 , 4 , 3 , 4

The first task is to find the median. For that, the values should be arranged in the ascending order.

1 , 1 , 2 , 2 , 2 , 3 , 3 , 4 , 4 , 4 , 4 , 5 , 10

Since there are fourteen values which are even, the median will be the mean of the middle values

$$ie. \ \frac{3+4}{2} = 3.5$$

Therefore Q2 = 3.5

Q1 is the mid value between the minimum and the median i.e. 2

Q3 is the mid value between the median and the maximum i.e. 4

The above set of values was given in to an online boxplot generator, Alcula.

Figure 4 shows the obtained results. From the boxplot, it is clear that the value 10 is an outliner.

$$(Q \times 3) + (1.5 \times IQR) = 4 + (1.5 \times 2) = 4 + 3 = 7$$

This is because it exceeds the limiting value i.e. 10 > 7

Therefore this opinion is ignored. Then the average of the remaining set of values is calculated

$$i.e. \frac{5+2+4+1+1+4+2+3+2+4+4+3+4}{13} = 3$$

Thus the trust factor value of the cloud service provider, ABCloud is 3.

This computation is carried out by the local trust manager at regular intervals of time as specified. Since the trust factor changes with time, it should be ensured that the value never becomes stale.

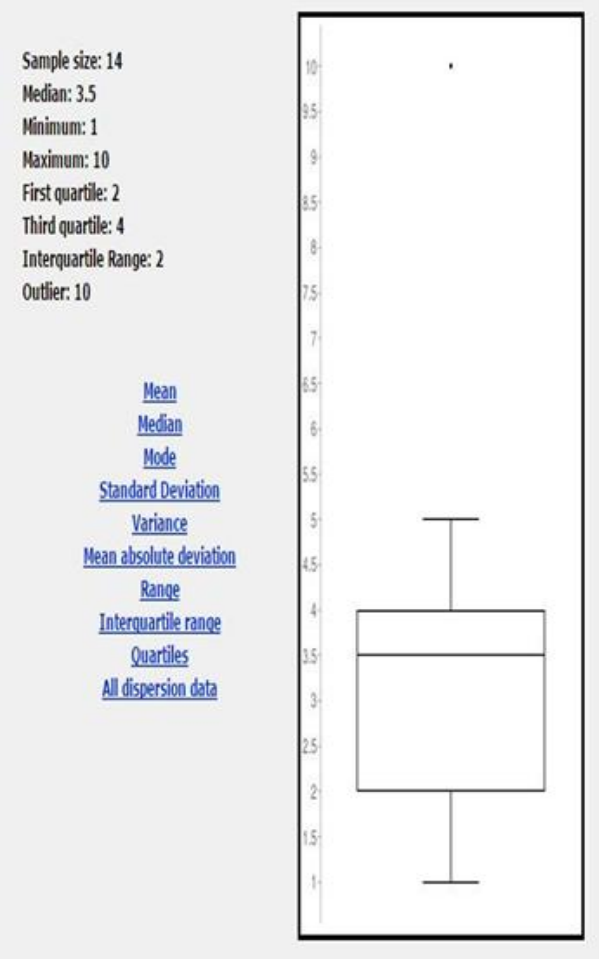

Fig.4. Boxplot

V. Conclusion and Future Work

Trust management is one of the biggest challenges in cloud computing especially when it comes to inter-cloud interaction. There are many existing methods of calculating trust rating. One such technique uses probability density function. The disadvantage is that dishonest clouds cannot be identified. Another method uses opinion from peers about a particular cloud to do transaction. The disadvantage is that the result goes wrong when the number of fake opinions is high.

This paper proposes an effective method that takes in to consideration only the latest ratings. The proposed architecture hierarchy has two levels: global and local trust managers. Global trust managers coordinate various local managers for communication. It also maintains a list that takes care of dishonest clouds.

Local trust managers are responsible for the clouds in their domain and calculate their trust factor. Trust factor is a value ranging from 1 to 10. Boxplot method is used to identify the outliners. These values will be more accurate. The major problem to be dealt here is time required to retrieve trust factor of all the clouds in the system.

Another problem is the time required to sort all of their trust factors. Performance degradation happens when the number of clouds is very large. Future work includes automation for identifying fake clouds using dynamic trust checking mechanism with the help of machine learning algorithms and evaluating its performance and the use of probability density function incorporated in the boxplot.

References

[1] Abawajy, Jemal. "Determining service trustworthiness in inter-cloud computing environments." Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on. IEEE, 2009.

[2] Josang, Audun, Roslan Ismail, and Colin Boyd. "A survey of trust and reputation systems for online service provision." Decision support systems 43.2 (2007): 618-644.

[3] Habib, Sheikh Mahbub, Sebastian Ries, and Max Mühlhäuser. "Towards a trust management system for cloud computing." Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on. IEEE, 2011.

[4] Noor, Talal H., et al. "Trust management of services in cloud environments: Obstacles and solutions." ACM Computing Surveys (CSUR) 46.1 (2013): 12.

[5] Abawajy, Jemal. "Establishing trust in hybrid cloud computing environments." Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on. IEEE, 2011.

[6] https://en.wikipedia.org/wiki/Box_plot.

[7] Vijayakumar, V., and R. S. D. W. Banu. "Security for resource selection in grid computing based on trust and reputation responsiveness." International Journal of Computer Science and Network Security 8.11 (2008): 107-115.

[8] Bonatti, Piero, et al. "An integration of reputation-based and policy-based trust management." networks 2.14 (2007): 10.

[9] Kotsovinos, Evangelos, and Aled Williams. "BambooTrust: Practical scalable trust management for global public computing." Proceedings of the 2006 ACM symposium on Applied computing. ACM, 2006.

[10] Kagal, Lalana, Tim Finin, and Yun Peng. "A framework for distributed trust management." proceedings of IJCAI-01 Workshop on Autonomy, Delegation and Control. 2001.

[11] Azzedin, Farag, and Muthucumaru Maheswaran. "A trust brokering system and its application to resource management in public-resource grids." Parallel and Distributed Processing Symposium, 2004. Proceedings.

18th International. IEEE, 2004.
[12] Carbo, Javier, Jose M. Molina, and Jorge Davila. "Trust management through fuzzy reputation." International Journal of Cooperative Information Systems 12.01 (2003): 135-155.
[13] Noor, Talal H., and Quan Z. Sheng. "Trust as a service: a framework for trust management in cloud environments." Web Information System Engineering–WISE 2011. Springer Berlin Heidelberg, 2011. 314-321.
[14] Weeks, Stephen. "Understanding trust management systems." Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on. IEEE, 2001.
[15] Wei, Xianglin, et al. "SMART: A subspace based malicious peers detection algorithm for P2P systems." *International Journal of Communication Networks and Information Security* 5.1 (2013): 1.
[16] Hesarlo, Parisa Sheykhi. "Security, privacy and trust challenges in cloud computing and solutions." *International Journal of Computer Network and Information Security* 6.8 (2014): 34.
[17] Derahman, M. N., A. Abdullah, and M. F. Azmi. "Robust Reputation Based Trust Management Framework for Federated-Cloud Environments." *International Journal of Applied Engineering Research* 11.21 (2016): 10601-10605.
[18] Xu, Jiuyun, et al. "Local reputation management in cloud computing." *2015 IEEE World Congress on Services*. IEEE, 2015.
[19] Filali, Fatima Zohra, and Belabbes Yagoubi. "Global trust: a trust model for cloud service selection." *International Journal of Computer Network and Information Security* 7.5 (2015): 41.

Authors' Profiles

Kiran Mary Matthew is pursuing her Mtech degree in Computer Science and Engineering with specialization in Cloud Computing at VIT University, Chennai. Her area of research includes key management in cloud computing and trust management in a hybrid cloud environment.

Prof. Abdul Quadir Md is a Ph.D. student and currently an assistant professor in the School of Computing Sciences and Engineering at VIT University, Chennai. His research focuses on Trust Management in Multi Cloud Environment. He has taught a number of courses on computers over the years.

How to cite this paper: Kiran Mary Matthew, Abdul Quadir Md,"An Effective Way of Evaluating Trust in Inter-cloud Computing", International Journal of Computer Network and Information Security(IJCNIS), Vol.9, No.2, pp.36-42, 2017.DOI: 10.5815/ijcnis.2017.02.05

