Opinion Model Based Security Reputation Enabling Cloud Broker Architecture

Pramod S. Pawar[1,2], Srijith K. Nair[2], Fadi El-Moussa[2], Theo Dimitrakos[2], Muttukrishnan Rajarajan[1], Andrea Zisman[1]

[1] City University London, London EC1V 0HB, United Kingdom
r.muttukrishnan@city.ac.uk, a.zisman@soi.city.ac.uk
[2] British Telecommunications, Security Practice, Adastral Park, Ipswich IP5 3RE, UK
{pramod.s.pawar, srijith.nair, fadiali.el-moussa, theo.dimitrakos}@bt.com

Abstract. Security and trust in service providers is a major concern in the use of cloud services and the associated process of selecting a cloud service provider that meets the expectations and needs of one's security requirements is not easy. As a solution we propose a broker architecture model that enables us to build a security reputation framework for cloud service providers, capturing comprehensive evidence of security information to build its trust and security reputation

Keywords: broker, reputation, subjective logic, security

1 Introduction

Cloud computing has become one of the fastest growing segments of the IT industry. Cloud computing involves a provider delivering a variety of IT enabled resources to consumers as a service over the Internet. Cloud computing services are offered as Software as a Service (SaaS), Platform as a Service (PasS) or Infrastructure as a Service (IaaS) [22]. Virtualization is a core enabling technology for cloud IaaS architectures. Even though several advantages of the use of cloud based services have been identified, in particular the pay-as-you-consume costing model and the minimization of capex costs, the inherent loss of control of data and process to external parties (cloud service providers) have the customers worried.

Since security remains a major concern in the use of cloud services, an individual or an enterprise expects a high level of confidence and trust in the cloud service provider it would like to use. The enterprise needs a process to identify and decide on the most suitable service provider to fulfill its security requirements for its service to be deployed. Reputation systems have been effectively used in making such decisions, however it is highly challenging to apply the concept to the cloud ecosystem, with a security context. This is challenging mainly due to the reluctance of the cloud service providers to publicize their security related information to the internet community or even to a selected group of customers. Relevant information may include events or incidence recorded due to security activities like firewall filtering, intrusion detection/prevention systems, security policies, authentication/authorization, identity management and key management.

However one also need to keep in mind the fact that IT service providers have been providing details of their security systems and associated processes to third party (security) auditors for obtaining security certifications and legal compliance status. These certifications are often essential requirements of the service provider to gain confidence of their customers and the industry as a whole. In order to obtain security certification the service provider needs to share, among other details, the security event related information to the third party auditors. The higher the level of security certification required, the more critical security events information and process details are expected by the auditors. In order to avoid security leakage it is a common practice to obtain non-disclosure agreements with auditors before this critical security information are shared. An enterprise needing cloud services have to rely on the security certifications of the cloud service providers to establish trust in the providers. This approach however constraint the enterprise to match their security requirements based only on the certification information published by the service providers and the associated minimum requirements that needs to be met by the service provider for obtaining the certification, due to unavailability of other detailed information.

As a way of breaking this impasse we propose the use of a Cloud Broker (CB) that inherits and expands on the role of the security auditor, enabling the broker to obtain access to the security events due to the high trust placed by the service providers, which may not be possible with the wider community. The CB provisions the enterprises with security reputation of the cloud service providers based on their security requirements as specified to the CB. The registration with the broker allows the cloud service providers to highlight their security strengths without exposing their internal security details like event information to the wider customer base and at the same time also benefited by CB's potentially wider customer base. The cloud service consumers benefit from the service that provides a closest match between their security requirements and the security reputation of the cloud service providers.

The remaining of the paper is structured as follows: Section 2 provides the background and related work. Section 3 describes the cloud broker architecture and its components. Section 4 describes our approach of the reputation modeling to build the security reputation of the cloud service provider. Section 5 provides applicability of this work in an existing project OPTIMIS – Optimized Infrastructure Services. Section 6 provides concluding remarks and future work.

2 Related Work

Reputation system based trust model have been adopted in several open systems such as internet websites, e-commerce, P2P Systems and mobile adhoc networks [7][15][16][6][12][17][9][18]. Resnick et. al. [15][16] discusses the importance of reputation system to decide whom to trust in the Internet where large number of producers or consumers may not know each other. Epinion [17], eBay [15][16] are some of the very popular electronic markets using reputation systems. Trust management systems help reduce free riding of the nodes in the P2P systems where each entity can act as client and server, expecting to contribute in the systems. The trust model for P2P systems in [21] considers transactions and shared experiences as recommendations and uses Bayesian estimation methods to compute trust values. The

Beta reputation model in [8] is based on beta distribution that considers the direct experience as well as feedback from other agents to model the behavior of a system. Both models [8][21] are based on the belief theory, but in [21] the use of Bayesian estimation expects probabilities for each question of interest.

The study of trust is closely related to *uncertainty* and we observe that many of the reputation system proposed have given either no importance or a very low importance to uncertainty. Exceptions are found in the works described in [7][14][10][13][20]. The belief model in [7] uses metric called *opinion* to describe *belief* and *disbelief* about a proposition as well as the degree of *uncertainty* regarding probability of an event. The work on [13][20] proposes opinion metric as in [7] but giving importance to uncertainty due to the evidence that impacts the belief and disbelief about a proposition. In [7] the uncertainty is modeled only based on the amount of total evidence i.e. as the total evidence increases, the uncertainty decreases, while in [13][20] the uncertainty also takes into account the amount of positive and negative evidence contained in total evidence. The work in [13] shows that it provides low prediction errors compared to [7][20]. Opinion models have been extensively used for estimating the quality by combining multiple factors. The opinion model proposed in [13] uses the subjective logic to combine evidences and due to its low prediction errors forms the best choice for building reputation of the cloud service providers.

In the recent years reputation systems have also been used in the cloud computing paradigm [1][3][5][13]. In [3], trust is one of the core component used by software as a service provider, along with risk, eco-efficiency and cost for evaluating the cloud infrastructure provider, for their service. The trust of the cloud infrastructure provider in [3] is evaluated by the model proposed in [13]. The work in [5] identifies several vulnerabilities in cloud services provided by Google, IBM, Amazon and proposes an architecture to reinforce the security and privacy by suggesting a hierarchy of P2P reputation system to protect cloud resources. The focus in [13] and [5] has been on use of conventional trust within a cloud service ecosystem and no specific context of security to build reputation of the cloud service providers is considered.

The concept of a broker as intermediaries between the service providers and service consumers with the aim of relieving the customer from evaluating trust and risk of the service provider has been used in the grid and cloud environments before [11][4][19][2]. The work in [4] proposes broker architecture in grids with the focuses on evaluating the reliability of the risk information from the resource providers. Within the context of cloud computing environment [11], cloud broker can be used as *i) cloud service intermediation:* intermediation for multiple services to add value-additions like identity management or access control *ii) cloud service aggregation:* bringing together two or more fixed cloud based service *iii) cloud service arbitrage:* similar to cloud service aggregation, but more dynamic aggregation to provide flexibility. The work in [11][4] have been focusing in identifying trust and risk of the service providers without any security context.

This paper proposes a broker architecture that enables the gathering of security related events of the cloud service providers, which otherwise is difficult to be shared with the end users, and uses the reputation model proposed in [13] to build the security reputation of the cloud service providers.

3 Cloud Broker Architecture

We introduce a Cloud Broker architecture that enables building of security reputation of individual service provider and sharing the same with its customers. The proposed broker architecture is shown in Figure 1 that includes various components namely: i) *Cloud Service Provider Interface (CSPI)* ii) *Enterprise users Interface (EUI)* iii) *Monitors (M)* and iv) *Trust Engine (TE)*. The entities involved in the architecture are Cloud Service Providers (CSP) and Enterprise Users (EU). The CSP and the EU register with broker. The registration of the CSP at the broker includes the agreement with the broker to share security related information with the broker and in turn the broker has a non-disclosure agreement with the service provider.

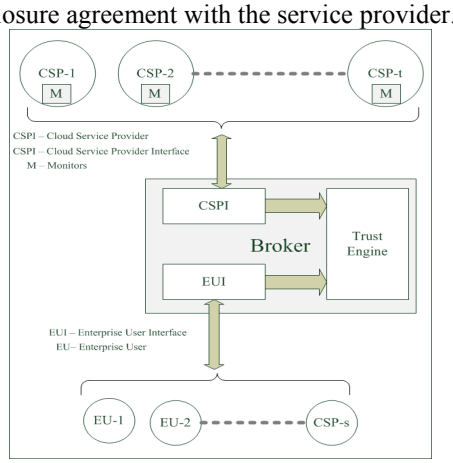

Figure 1: Cloud Broker Architecture

3.1 Cloud Service Provider Interface (CSPI)

This interface enables the service provider to provide details of its security practices and security measures in place, allowing advertising its security strengths. In our experience, we find cloud service providers try to provide the following security measures as a basic step towards securing their customers environment: *i) Protecting individual virtual environment ii) Filter traffic between each virtual instances iii) Hardening the hypervisor iv) Protecting the network infrastructure v) Protecting the data stored at each individual virtual instance vi) Policy enforcement for authentication and access management to individual virtual instances vii) Patch management*

3.2 Enterprise User Interface (EUI)

This interface allows the enterprise users to input their security requirements, select most appropriate cloud service provider for their security needs, provide

feedback on the services and also register complaints. The requirements associated with a service and the security features expected, are encoded in the *service manifest* as discussed in [3].The feedback and the complaints form a vital piece of evidence to model the cloud service providers reputation based on its security strength.

3.3 Monitors

The broker receives security violations events of the service provider by registering to the pub-sub [18] monitors in the service provider's infrastructure. The threats that prevent organizations from adoption of the cloud services and the areas for gathering metrics are identified as follows: *i) Insecure Authentication or Authorization:* Interface allowing customers to manage cloud services in order to perform provisioning, management, orchestration, and monitoring their virtual instances *ii) Insider Attack:* An insider from cloud service provider could have privileged access to confidential data or gain control over the cloud service with no or little risk of detection *iii) Multitenant Attack:* Cloud environment is meant to allow multiple users share resources (CPU, network, memory, storage, etc.) and an improper isolation of the multi-tenant architecture may lead to have access to any other tenant's data *iv) Data Leakage:* Customers data on the cloud could be compromised, deleted or modified *v) Malware Propagation:* Any malware that infects a virtual instance could propagate over the shared host or to hypervisor, spreading rapidly, giving ability to eavesdrop on customer's transactions.

3.4 Trust Engine

The trust engine contained in the cloud broker is the core part of the architecture that performs the *trustworthiness* calculation for the cloud service providers. Figure 2 shows the internal work flow used for computing the reputation of cloud service provider based on the inputs received from the interfaces of the broker.

i. *Evidence:* The evidential information is gathered from the three sources namely monitors, cloud service provider interface and enterprise user interface. These evidences are provided to the Opinion Model.

ii. *Opinion Model :* The evidences received from different monitors are used to form an opinion about a cloud service provider based on the opinion model proposed in [13]. The opinion of a proposition x, represented as $w(x)$ or w_x is defined in terms of *belief* $b(x)$ or b_x, *disbelief* $d(x)$ or d_x and *uncertainty* $u(x)$ or u_x where $b(x)+d(x)+u(x)=1$. The opinion model in [13] is given as follows:

$$W_x = (b_x, d_x, u_x, a_x) \quad (1)$$

$$b_x = c\ r / t \quad (2)$$

$$d_x = c\ s / t \quad (3)$$

$$u_x = t / (r\,s + f^2 + 1) \quad (4)$$

$$c = 1 - u_x \quad (5)$$

where: *r* is amount of positive evidence; *s* is amount of negative evidence; *t* is total evidence given as *t=r+s*; *c* or *c(t) or c(r,s)* is certainty as a function of total evidence; and *f* is distance of focus to the centre of an ellipse formed by mapping the positive and negative evidence to major and minor semi-axes of an ellipse.

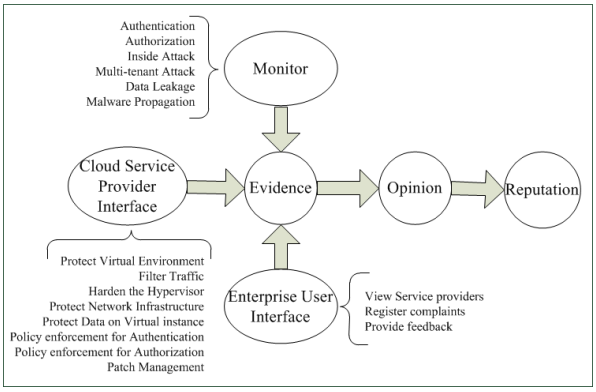

Figure 2: Trust Engine

The opinion formed by the monitors is combined with the opinion formed based on the enterprise user's feedback and complaints. The subjective logic by Josang [7] is used to combine multiple opinions to form a single opinion using the operators such as conjunction, consensus that allows performing logical operations on opinions. This paper uses the opinion model proposed in [13] and the subjective logic operators [7]. The conjunction operator is standard logic "AND" operating on the opinions. The consensus operator enables combining the opinions of entity A and entity B representing an imaginary entity [A,B]'s opinion about proposition *x*.

iii. Reputation: The probability expectation of an opinion is used to provide the reputation rating. The expectation of an opinion is given as *E(w(x))=b+au* where *E(w(x)) € [0,1]* and *a(x)* is base rate that provides the weight of uncertainty that contributes to the probability expectation.

Figure 2 shows process of modeling the security reputation by broker. The first step is the broker getting evidential information from two sources a) Monitor and b) Customer interface. The second step is to convert the evidence obtained to compute an opinion. The third step is to calculate the reputation of a service provider based on the opinion formed. The details of reputation calculation are given in section 4.

4 Reputation System

The reputation of a cloud service provider is calculated in terms of its *trustworthiness(T)* using opinion obtained from computations, namely i) *Incidence Monitoring(M)*: Security incedence events received from monitoring ii) *Enterprise User Rating(EUR)*: Ratings provided by the enterprise user for satisfaction of the

security features provided by CSP. The *trustworthiness(T)* is given by applying the conjunction operator of subjective logic on the opinions obtained from each of these computation and then calculating the expectation of the combined opinion.

$$T=\text{Expectation}\ (W_{M\ \Lambda\ EUR}) \quad (6)$$

Where W_M is the opinion obtained from the monitoring(M) as well as the W_{EUR} is the opinion obtained from the enterprise user ratings(EUR). The symbol Λ is the *conjunction operator* used to combine the two opinions.

4.1 Incidence Monitoring

The incidence monitoring records evidence about the incidences related to parameters such as authentication, authorization, inside attacks, multi-tenent attack, data leakage and malware propogation. These incidences can either be identified by the cloud service provider and sent to the broker or the broker after receiving the security events carries further analysis to identify the incidences from the data received. Both approaches have their own advantages and disadvantages.

For each monitoring parameter, the number of incidents occuring within a time window *w* are observed. Every incident identified, adds to the negative evidence and absence of incidents increases the positive evidence. Based on the positive and negative evidences, opinions are formed for each of the parameters. Let W_{AT}, W_{AR}, W_{IA}, W_{MT}, W_{DL}, and W_{MP} be opinions formed for CSP based on the monitoring parameter of authentication, authorization, inside attacks, multi-tenent attack, data leakage and malware propogation respectively. Consider for example that there are *n* monitors associated with monitoring of authentication incidence at CSP-1. Then the opinion W_{AT} for CSP-1 is given as the *consensus* of all *n* monitors. Considering all monitoring parameters, the overall opinion W_M for CSP-1 is given by applying *conjunction* operator over the *consensus* opinion, which is as follows:

$$W_M = W_{AT}^{M1,\ldots,Mn} \Lambda\ W_{AR}^{M1,\ldots,Mn} \Lambda\ W_{IA}^{M1,\ldots,Mn} \Lambda\ W_{MT}^{M1,\ldots,Mn} \Lambda\ W_{DL}^{M1,\ldots,Mn} \Lambda W_{MP}^{M1,\ldots,Mn} \quad (7)$$

Where $W_{AT}^{M1,.,Mn}$ is consensus opinion by monitors M1 to Mn regarding authentication. Similarly consensus opinions for other parameters are obtained.

4.2 Enterprise User Rating

For every usage of the services from the CSP, the enterprise user rates the satisfaction of security features and capabilities provided by the CSP corresponding to the requirements set forward initially by the user. Consider *q* enterprise users registered with the broker and provide ratings to the CSP for each of the monitoring parameters. The overall opinon W_{EUR} for CSP-1 based on the enterprise user rating is given by applying the *conjunction* operator over the consensus opinion, as follows:

$$W_{EUR} = W_{AT}^{EU1,EU2\ldots,EUq} \Lambda\ W_{AR}^{EU1,EU2\ldots,EUq} \Lambda\ W_{IA}^{EU1,EU2\ldots,EUq} \Lambda\ W_{MT}^{EU1,EU2\ldots,EUq} \Lambda\ W_{DL}^{EU1,EU2\ldots,EUq} \Lambda\ W_{MP}^{EU1,EU2\ldots,EUq} \quad (8)$$

Where $W_{AT}^{EU1,EU2...,EUq}$ is consensus opinion for CSP-1 given by enterprise user EU1 to EUq based on the authentication. Similarly $W_{AR}^{EU1,EU2...,EUq}$, $W_{IA}^{EU1,EU2...,EUq}$, $W_{MT}^{EU1,EU2...,EUq}$, $W_{DL}^{EU1,EU2...,EUq}$ and $W_{MP}^{EU1,EU2...,EUq}$ are the consensus opinion for CSP-1 by EU1 to EUq based on authorization, inside attacks, multi-tenent attack, data leakage and malware propogation respectively.

4.3 Trust of Cloud Service Provider

The *trustworthiness(T)* of the cloud service provider is given by the calculating the expectation of the opinions W_M and W_{EUR} given by Incidence *monitoring* and the *Enterprise User* respectively. The *trustworthiness(T)* can be represented as:

$$T = \text{Expectation}(W_M \wedge W_{EUR}) = \text{Expectation}(W_{M \wedge EUR}) \quad (9)$$

Where $W_{M \wedge EUR} = (b_{M \wedge EUR}, d_{M \wedge EUR}, u_{M \wedge EUR}, a_{M \wedge EUR})$ and the expectation of the opinion $W_{M \wedge EUR}$ is given as :

$$E(W_{M \wedge EUR}) = b_{M \wedge EUR} + (a_{M \wedge EUR})(u_{M \wedge EUR}) \quad (10)$$

5 Applicability of this architecture

The cloud broker architecture proposed in this paper is a very generic and not limited to any specific environment. However, a practical, environment specific implementation of the proposed architecture is being used in the OPTIMIS [3][11] project. OPTIMIS toolkit is a set of software components for simplified management of cloud services and infrastructures that assists the cloud service providers to provide optimized services based on the TREC (Trust, Risk, Eco-efficiency and Cost).

TREC components are part of the basic toolkit. The trustworthiness of an IP (Infrastructure Provider) enables the SPs (Service Provider) to identify and select the IP having proven capabilities to provide the required service. The risk assessment performed provides the SP with the risk involved in the construction, deployment and operation of a service. The eco-efficiency aids in selecting a cloud service provider based on the energy consumption. Along with the trust, risk and eco-efficiency factor, cost forms the trade-off factor in providing of the optimized service.

The broker architecture [11] in the OPTIMIS project already have a support of the TREC toolkit, SLA agreement and the monitoring infrastructure which can be enabled to build the security reputation of the IP using the proposed reputation model [13] described in section 4 and the security related events captured in section 3. Figure 3 shows the high level sequence diagram for broker implementation in OPTIMIS project. Following are the sequence of steps: *a)* The SP uses the *IDE (Integrated development Environment)* to create a service which is described in a *service manifest b)* The *IDE* passes the service manifest and the optimization objective to the *SD (Service deployer)* for deployment of the service *c)* The *SD* uses the cloud broker interface to submit the *service manifest* and the optimization objective *d)* The cloud broker has *Registry* where all SPs and IPs register before using the cloud broker services *e)* The broker after receiving a request for deployment of a service gets the list of IPs from the *Registry f)* The *TREC* component of the broker

contains the historical assessments of all SPs and IPs stored in the *DB (Database) g)* Based on the TREC assessments, the broker filters the IPs and the *DO (deployment Optimizer)* initiates SLA negotiations with the filtered IPs *h)* In the process of negotiation, the broker interacts with the AC(Admission control) which checks its current infrastructure status and provides offers based on the request made *i)* Once all the offers for all the components of the service is received the broker applies the optimization algorithm to provide the SP with the ranked list of IPs for each of its service components based on the TREC *j)* The SP deploys all its components considering the ranked list *k)* The service is deployed using the CO (cloud Optimizer) at the IP side. The CO provides all VM(virtual machines) related information to the SP, which in turn is forward to the Broker *l)* The broker passes the VM information, to the TREC components to receive monitoring events for these service components

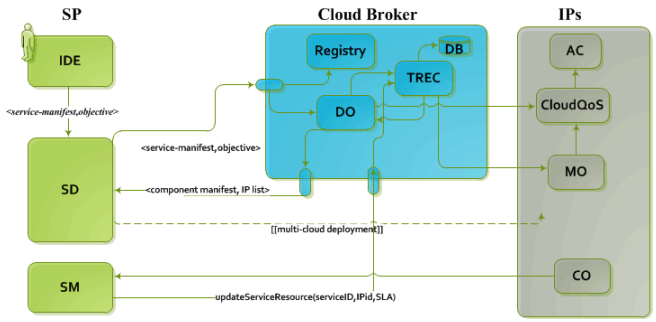

Figure 3: High level sequence diagram for broker in OPTIMIS

6 Conclusion and Future Work

In this paper we propose security reputation systems using broker architecture for cloud service providers, allowing customers to achieve a level of expectation from cloud service providers about their deployed security systems. By having a broker and using security reputation based on evidence helps customers build confidence in using a specific service provider and also gives incentive to cloud providers to demonstrate their security capabilities to the customers. As future work we intend to identify a comprehensive security requirements that map to the monitoring infrastructure which will enable the broker to provide the cloud service provider and the enterprise user with a generic interface to specify its capabilities and requirements. We also aim to perform a rigorous evaluation of the proposed architecture by using the simulated as well as real data of the cloud service providers using the OPTIMIS infrastructure.

Acknowledgment

This work has been partially supported by the EU within the 7th Framework Programme under contract ICT-257115-Optimized Infrastructure Services (OPTIMIS).

References

1. M. Alhamad, T. Dillon and E. Chang. SLA-Based Trust Model for Cloud Computing 13th International Conference on Network-Based Information Systems (2010)
2. C. Dumitrescu, I. Raicu and I. Foster. DI-GRUBER: A Distributed Approach to Grid Resource Brokering. In Proceedings of the ACM/IEEE conference on Supercomputing (2005)
3. A.J. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R.M. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, S.K. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgó, T. Sharif, and C. Sheridan. OPTIMIS: a Holistic Approach to Cloud Service Provisioning, Future Generation Computer Systems (2011)
4. I. Gourlay, K. Djemame and J. Padgett. Reliability and Risk in Grid Resource Brokering. IEEE International conference on Digital Ecosystems and Technologies (DEST) (2008)
5. K. Hwang, S. Kulkarni and Y. Hu. Cloud Security with Virtualized Defense and Reputation-based Trust Management. Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (2009)
6. Y. Jin, P. Bloch and G. Cameron, A Comparative Study: Does the Word-of-mouth Communications and Opinion Leadership Model Fit Epinions on the Internet?. Proceedings of the Hawaii International Conference on Social Sciences (2002)
7. A. Jøsang. A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9(3):279–311(2001)
8. A. Jøsang and R. Ismail. The Beta Reputation System. In Proceedings of the 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy (2002)
9. R. Kerr and R. Cohen. Modeling trust using transactional, numerical units. In PST '06: Proceedings of the Conference on Privacy, Security and Trust, Markham, Canada (2006)
10. F. Li and J. Wu. Uncertainty Modeling and Reduction in MANETs IEEE Transactions on Mobile Computing, Vol. 9, No. 7 (2010)
11. S. K. Nair, S. Porwal, T. Dimitrakos, A. J. Ferrer, J. Tordsson, T. Sharif, C. Sheridan, M. Rajarajan and A. U. Khan. Towards Secure Cloud Bursting, Brokerage and Aggregation. Web Services (ECOWS), IEEE 8th European Conference (2010)
12. L. Page, S. Brin, R. Motwani, and T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Technical report, Stanford Digital Library Technologies Project (1998)
13. P. S. Pawar, R. Muttukrishnan, S. K. Nair, and A. Zisman. Trust Model for Optimized Cloud Services. Sixth IFIP International Conference on Trust Management (2012)
14. I. Ray, N. Poolsappasit and R. Dewri. An Opinion Model for Evaluating Malicious Activities in Pervasive Computing Systems.
15. P. Resnick and R. Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. The Economics of the Internet and E-Commerce. Michael R. Baye, editor. Volume 11 of Advances in Applied Microeconomics. Amsterdam, Elsevier Science. pp. 127-157
16. P. Resnick and R. Zeckhauser, J. Swanson, K. Lockwood. The Value of Reputation on Ebay: A Controlled Experiment. Experimental Economics, Volume 9, Number 2, pp. 79-101(23) (2006)
17. J. Schneider, G. Kortuem, J. Jager, S. Fickas and Z. Segall. Disseminating trust information in wearable communities. Personal and Ubiquitous Computing, Volume 4, Number 4, 245-248, DOI: 10.1007/BF02391568
18. M. Srivatsa and L. Liu. Secure Event Dissemination in Publish-Subscribe Networks. 27th International conference on Distributed Computing Systems (ICDS'07) (2007)
19. S. Venugopal, R. Buyya and L. Winton. A Grid Service Broker for Scheduling e-Science Applications on Global Data Grids. In Concurrency and Computation: Practice and Experience, Volume 18, Issue 6, Wiley Press, New York, USA, pp. 685-699 (2006)
20. Y. Wang and M. P. Singh. Evidence-Based Trust: A Mathematical Model Geared for Multiagent Systems. ACM Transactions on Autonomous and Adaptive Systems, Vol. 5, No. 4, Article 14, (2010)
21. P. Wu and G. Wu. A Reputation-Based Trust Model for P2P Systems. International Conference on Computational Intelligence and Security (2009)
22. http://csrc.nist.gov/publications/PubsSPs.html#800-145. The NIST Definition of Cloud Computing. Special Publication 800-145.