A Mobile Agent based Approach of ensuring Trustworthiness in the Cloud

Aravindh Ramaswamy[#1], Aswath Balasubramanian[#2], Palaniappan Vijaykumar[#3], Varalakshmi P[#4]

[#]*Department of Information Technology, Anna University Chennai*
India

[1]aravindhramu@gmail.com,[2]aswath78@gmail.com,[3]vijaypalani6@gmail.com, [4]varanip@gmail.com

Abstract— **Trust in Cloud Computing is a very important factor. At any instant of time, there must be a trustworthy relationship between the Cloud Service Provider and the Cloud Customer. This paper discusses a novel mobile agent based approach of ensuring trustworthiness in the Cloud. The idea revolves around three entities namely a Cloud Broker, Cloud Customer and the Cloud Service Provider. On the basis of penalties, prize points and monitoring mechanisms of mobile agents, trust is ensured.**

Keywords— **Trust, Cloud Computing, Mobile agents, penalty.**

I. Introduction

Cloud Computing is an internet-based computing that includes sharing of software, infrastructure or platform over the internet on an on-demand and rental basis. Several companies like Google, Amazon, and Microsoft have already ventured into Cloud Computing. Several other industries are flourishing by renting resources as and when they need it thereby reducing costs and increasing profits. Based on the services offered, Cloud computing paradigm can be classified under 3 categories:

- Software as a Service (SaaS)
- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS).

In SaaS, the cloud Application is rented as a Service. For eg, in Google Docs. Here, a complete office suite is offered totally free through the internet. In IaaS, infrastructural resources are available on a rental basis; for e.g., a 1 TB resource for $500 per month. Similarly, PaaS offers the platform for deploying cloud applications on a rental basis itself.

On the basis of cloud deployment, three kinds of architectures are possible:

- Private Clouds.
- Public Clouds.
- Hybrid Clouds.

Private or internal Clouds emulate Cloud in a private network. In public or external clouds, resources are dynamically provisioned through the internet. A hybrid cloud consists of several internal and external service providers and will be used for most enterprises.

However, due to security concerns, there still remains a lag that prevents companies to venture completely into the Cloud. Security in Cloud Computing is a core research area; As this involves complete involvement of private data with 3rd party's resources. This paper focuses on maintaining the 'trustworthiness' of the Cloud Entities.

Trust is defined as the degree of belief/confidence one person has on another. The challenge lies in quantizing the magnitude of trust. As far as Cloud Computing is concerned, trust is a very important factor that needs to be established between the Cloud Service Provider (CSP) and the Cloud Customer (CC). Customers who request service may turn out to be malicious by tampering other people's requests. Similarly, the CSP also has to be ensured of trustworthiness because he dispatches the user's jobs for execution and also the user's private data is in his possession.

In this paper, we propose a mobile agent based approach that ensures the trustworthiness of both the customer and the service provider. Section II discusses about work related to security and trust in distributed Computing environments. Section III details the new agent-based trust architecture. Section IV describes the proposed work. Section V shows the simulation results and Section VI concludes the work.

II. RELATED WORK

[1] Proposes a Trust Enhanced Security Model (TESM) for trusted computing platforms. It uses "hard trust" whose value is obtained from computation and "soft trust" whose value is obtained from past experiences. This paper also introduces Binary attestation and property based attestation mechanisms, using which attestation requester can be sure of the trustworthiness of the platform. [2] Deals with preserving privacy for cloud customers and also the "celebrated" data security problem in the cloud data centres. [3] Discusses the use of Cloud Firewall technology that combines the network security aspects with Cloud Computing. [4] Proposes a Trusted Cloud Computing Platform that provides a closed box execution environment. [5] Discusses the usage of cryptographic protocols for providing security in the cloud.

This paper focuses on a mobile-agent based approach of ensuring trustworthiness in Cloud. Both the customer and the service provider are ensured of trustworthiness using this approach.

III. ARCHITECTURE

The proposed architecture for ensuring trustworthiness is shown in Fig. 1. There are three basic entities involved in this trusted cloud system:

978-1-4577-0590-8/11/$26.00 ©2011 IEEE

- The Cloud Broker, a trusted third-party.
- The Cloud Customer (CC).
- The Cloud Service Provider (CSP).

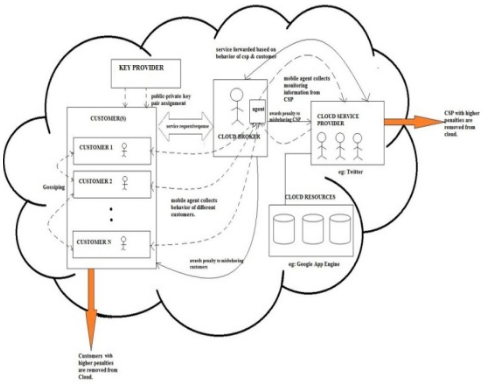
KEY PROVIDER
public-private key pair assignment
CUSTOMER(S)
CUSTOMER 1
CUSTOMER 2
CUSTOMER N
Gossiping
service request/response
CLOUD BROKER
agent
service forwarded based on behavior of csp & customer
mobile agent collects monitoring information from CSP
awards penalty to misbehaving CSP
CLOUD SERVICE PROVIDER
eg: Twitter
CSP with higher penalties are removed from cloud.
mobile agent collects behavior of different customers.
CLOUD RESOURCES
eg: Google App Engine
awards penalty to misbehaving customers
Customers with higher penalties are removed from Cloud.

Fig. 1. Overall Architecture of the Agent-based Trust mechanism

The Cloud Broker is a trusted third-party. He monitors both the Customer and the Cloud Service Provider and ensures them of trustworthiness and purges the malicious persons from the Cloud. The "Key Provider", is a sub-entity of the Cloud Broker and is responsible for assigning keys to the Customer. The public key and the private key are analogous to the username and password respectively (except that these credentials are not modifiable by the user here). The Customers are the persons who visit the Cloud for accessing the Cloud Service. In addition to this , the Cloud Broker maintains an agent pool from which agents are dynamically added/ dropped based on service requests from the customer.

IV. PROPOSED WORK

The scenario starts when the customer enters the Cloud. The Key Provider assigns a pair of keys (private and a public key) to each customer. All transmissions are assumed to be through a secured channel.

Each Service Provider first registers with the Cloud Broker. This registration is mandatory for an entity to be qualified as a Service Provider. Next, the Broker has to ensure the trustworthiness of the CSP to further direct the customer requests to CSP. Hence the Cloud Broker first sends a dummy job request to the CSP. Once the job request has been sent to the CSP, an agent is invoked by the Broker from the Agent pool to monitor the Virtual Machine (VM) instances that have been instantiated by the CSP. Our assumption is that a trustworthy CSP will fully utilize the Cloud Resources and provide fast and efficient service. Hence by monitoring VMs, the Cloud Broker can get to know whether the CSP is correctly executing the job or not. The agent monitors this and reports back to the Cloud Broker. If the performance is satisfactory, the CSP is retained in the cloud. Else, if he is found to be suspicious, then he is awarded a penalty that results in decrementing the trust index of the CSP. This penalty awarding is performed till the tolerance threshold of the Cloud Broker. Once this threshold is reached, the CSP is purged out of the cloud with the belief that he is a malicious entity. Once the Cloud Broker finds that the trust index of the CSP is within the tolerance threshold, he permits the customer's jobs to reach the CSP.

Meanwhile, once the Customer's jobs have been executed, each Customer shares his User Experience with other customers through "Gossiping". Each Gossip Message is of the format as shown in Fig. 2. This collaborative Gossip Message is used to calculate the trust index of the CSP.

MESSAGE_IDENTIFIER is a unique identifier of each message. SOURCE_CUSTOMER_IDENTIFIER refers to the customer identifier who originated the advertisement message. If B (BROADCAST_FLAG) is set to 1, then the DESTINATION_CUSTOMER_IDENTIFIER field is not set. If it is 0, then the DESTINATION_CUSTOMER_IDENTIFIER field is set. It specifies the id of the customer to whom the gossiping message is sent. The USER_EXPERIENCE field indicates the user experience of the customer who originated the message. It is either +1 or -1. +1 is an indicator of a happy and satisfied customer while -1 indicates a dissatisfied customer. This is used to dynamically alter the trust index of the CSP. S/G flag specifies if the message is a service request or a gossip message. If it is 1, then the message is a service request. Else it is a gossip message. If the S flag is specified, the B and DESTINATION_CUSTOMER_IDENTIFIER are set to don't care fields. All Service messages are sent to the CSP only. After a Service message has been sent, the Cloud Broker then expects the job request from the customer immediately after this message.

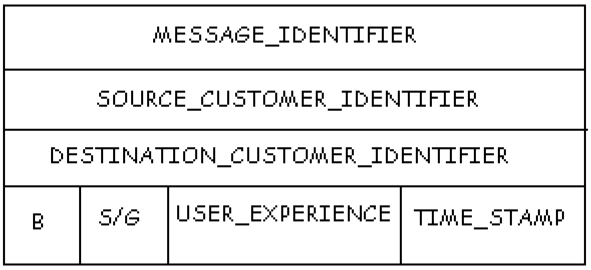
<table>
<tr><td colspan="4">MESSAGE_IDENTIFIER</td></tr>
<tr><td colspan="4">SOURCE_CUSTOMER_IDENTIFIER</td></tr>
<tr><td colspan="4">DESTINATION_CUSTOMER_IDENTIFIER</td></tr>
<tr><td>B</td><td>S/G</td><td>USER_EXPERIENCE</td><td>TIME_STAMP</td></tr>
</table>

Fig.2. Message packet of the Customer(s)

One possibility is that customer may turn out to be malicious. So it's the duty of the Cloud Broker to ensure the trustworthiness of the Customer too. So another agent is invoked from agent pool of the Cloud Broker. This agent monitors the Gossip messages of the Cloud Customer. The agent then monitors the gossip messages. If the customer is malicious, he may try to tamper the reputation of the Cloud Service Provider. So he may send false gossiping messages with the User Experience field set to -1 even if the CSP is trustworthy. Since the Cloud Broker knows whether CSP is trustworthy or not he can correctly guess if the customer is lying or not. So, such customers are awarded a penalty that reduces the trust index of the customer. This process is carried out till the tolerance threshold of the Customer is reached. Once this threshold is crossed, the malicious Customer is purged out from the cloud. Customers who share true experience regarding the CSP are awarded “prize points” that increases their trust index. The trust index value of the customer and the CSP are obtained from the equations (1) and (2).

All customer messages are encrypted with the customer’s private key by default and sent to the Cloud Broker. It is the Cloud Broker who has to forward the messages to the destination specified. Each message from the customer is time stamped. So this avoids Message-Replay attacks. Also since only the Cloud-Broker knows the decryption key/ public key of the customer, there is no possibility for Man-in-the-middle attack. Fig. 3 shows this process very clearly.

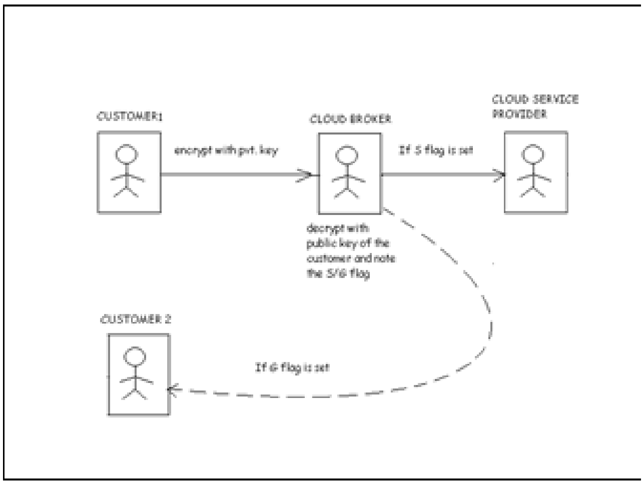

Fig. 3. Redirection of customer messages by the cloud Broker

The trust index of the CSP (TI_{CSP}) is provided by:

$$TI_{CSP} = e^{\pm t} * \left(\frac{Number\ of\ VMs\ launched\ for\ a\ job}{Number\ of\ Vms\ required\ for\ a\ job} + \sum_{i=1}^{n} UE_i + \sum_{i=1}^{k} PE + P \right) \quad (1)$$

And Trust Index of the ith customer is given by:

$$TI_{customer} = e^{\pm t} * (UE_i * (PE \text{ or } P \text{ awarded to CSP by broker}) + PP_i) \quad (2)$$

Where:

n <= number of customers who have been serviced
k <= tolerance threshold
UE$_i$= User Experience of ith customer
Penalty (PE) is punishment due to malicious behaviour
Prize (P) is award for trustworthy behaviour of CSP by Cloud Broker.
Prize points (PP$_i$) -> award for ith trustworthy customer by Cloud Broker.

The exponential factor is used as a multiplier to elucidate the fact the trust values decay with time. This is in accordance with the fact that trust values decay with time. This can be explained with a small example. Consider two persons Alice and Bob who are totally new to each other. Alice does some favor to Bob and Bob’s trust for Alice increases. After a couple of years, they meet again. Naturally, the trust which Bob had for Alice two years ago would have reduced by then. This is an analogy to show that trust reduces exponentially with time. The ‘±’ in the exponential multiplier is used in such a way that if the T.I. value is negative, positive exponent is used to further reduce the T.I. value with time and the negative exponent is used for positive T.I. values to reduce it with time.

V SIMULATION RESULTS

The cloud environment is created using Open Nebula, an open source toolkit for creating public, private and hybrid clouds. The mobile agents are created using IBM Aglets SDK. We consider the following scenarios to show the performance: Consider a cloud site that contains 1 customer, a cloud broker and a Cloud service Provider. We shall see the effect of User experience sharing, penalties and prize-points on the trust index of the customer and CSP. Consider that all the customers provide a same job request to the CSP. We show two situations. In the first scenario, the Customer is malicious and in the second the CSP is malicious. The tolerance threshold of the Cloud Broker is fixed as 5.

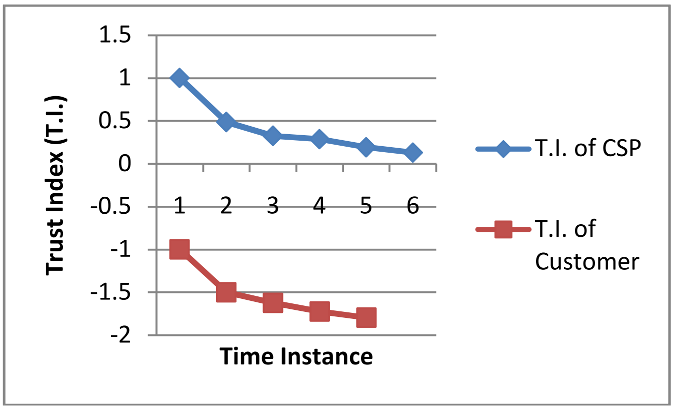

Fig.4 Scenario 1: Customer is malicious and he sends false UE about genuine CSP

Fig. 4 shows the effect of trust index when the customer is malicious. There is no trust index value for the last time index because the malicious customer is purged out of the cloud due to consistent negative (false) feedback about a trustworthy CSP.

Fig. 5 shows the effect of trust index when the CSP is malicious. This time the CSP is purged out of the cloud and hence there is no trust index value in the last time instant.

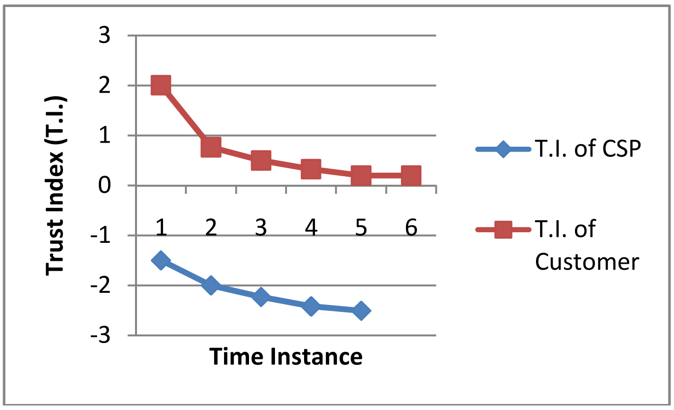

Fig. 5 Scenario 2: CSP is malicious and customer is genuine.

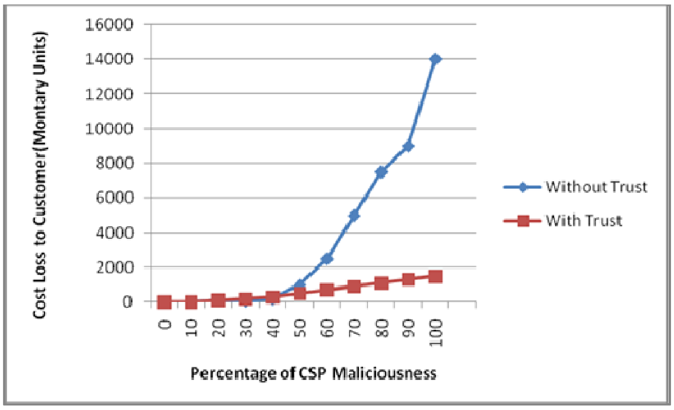

Fig. 6. Effect of CSP maliciousness on Customer in terms of money

Fig. 6 clearly shows that if CSP becomes increasingly malicious, the customer is ultimately affected leading to huge loss of investments. Hence it is very essential to purge off the untrustworthy Service Providers from the Cloud. But in the presence of the trusted environment, loss to the customer is reduced significantly.

Fig.7 shows the effect of maliciousness of Cloud Service Provider. Since the Cloud Broker is a trusted third party, his Trust Index (T.I.) values remain constant. If a customer reports correctly regarding the malicious CSP, he is considered trustworthy and hence his T.I. value increases. But if he reports incorrectly regarding the malicious CSP, then he is also untrustworthy and hence his T.I. values decrease. The tolerance threshold of the Cloud Broker is assumed to be 5. So after the tolerance threshold reaches a value of 5 units, the malicious CSP and the malicious customer are removed from the Cloud.

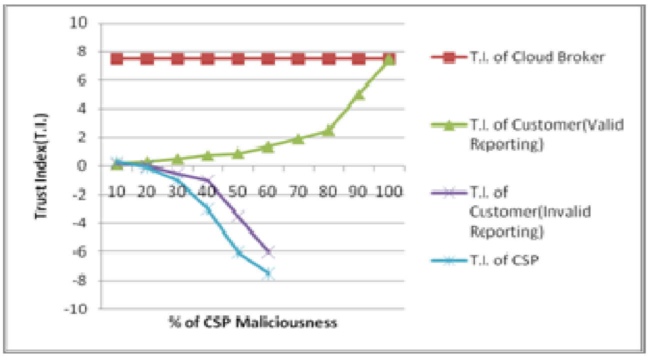

Fig. 7 Effect of CSP Maliciousness on other Cloud Entities

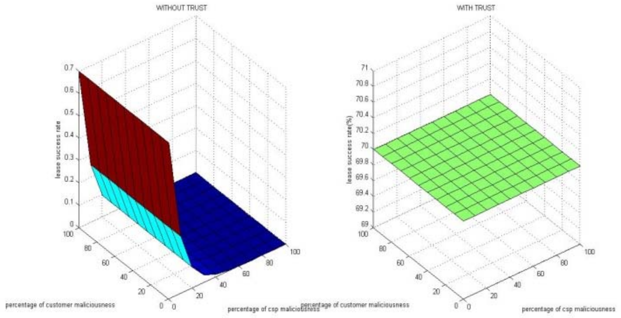

Fig. 8 Effect of Customer and CSP Maliciousness on Lease Success Rate

Fig.8 shows the effect of the customer's and CSP's maliciousness on the Lease Success Rate. Lease Success Rate is defined as the ratio of number of successfully executed leases to the total number of leases. Each customer job is assumed to consist of several VMs. So in a cloud environment, usage of cloud resources to run the VMs is done with the help of leases (or contracts). Without a trusted environment, with the increase in the maliciousness of the customer and the CSP, lease success rate reduces exponentially. But in the presence of the trusted environment, the lease success rate is maintained consistently.

VI CONCLUSION

This paper illustrates a novel agent-based mechanism of ensuring trustworthiness in the cloud in the sides of the customer and the Cloud Service Provider.

With the presence of penalties and prize points, the trustworthy customer(s) / CSP(s) are identified and the malicious ones are purged out from the Cloud. The monitoring and control part is played by the Cloud Broker who is responsible for ensuring trustworthiness in the cloud. All the messages are encrypted and sent via a secure channel to prevent eavesdroppers.

REFERENCES

[1] Aarthi Nagarajan and Vijay Varadharajan, *Dynamic Trust Enhanced Security Model for Trusted Platform based Services*, Future Generation Computer Systems, 2010.

[2] Jian Wang, Yan Zhao, Shuo Jiang, Jiajin Le, “Providing Privacy Preserving in cloud computing", International Conference on Test and Measurement, 2009.

[3] Weili Huang, Jian Yang, “New Network Security Based on Cloud Computing” , 2nd International Workshop on Education, Technology and Computer Science, 2010.

[4] Nuno Santos, Krishna P. Gummadi, Rodrigo Rodrigues, “Towards Trusted Cloud Computing,” IEEE International Conference, 2010.

[5] Christian Cachin, Idit Keidar, Alexander Shraer, “Trusting the Cloud”, IBM Research, Zurich Research Laboratory.