IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 4, Issue 4, April 2017

Decentralized Trust Management and Trust Worthiness of Cloud Environments

[1] Qhubaib Syed [2] Syed Afzal Ahmed [3] Syed Abdul Haq
[1] PG Scholar [2] Associate Professor [3] Head of Department
[1][2] Department of Computer Science and Engineering, Quba College of Engineering & Technology,
Nellore, AP, India

***Abstract*: Now days, cloud computing an ever green technology used by all segment of areas. To choose best cloud service providers trust is an important factor in growing cloud computing. The highly dynamic nature of cloud system services throws different challenges on privacy, security and availability. The communication between consumers and trust management system involves sensitive information that makes the privacy is an important factor. Protecting cloud services from malicious users is a difficult problem. During this paper we tend to planned a brand new approach Cloud Armor, a reputation-based trust management framework that has a collection of functionalities to deliver trust as a service (TaaS), which incorporates i) a completely unique protocol to prove the believability of trust feedbacks and preserve users' privacy, ii) An adaptive and strong believability model for measure the believability of trust feedbacks to shield cloud services from malicious users and to check the trait of cloud services, and iii) an accessibility model to manage the provision of the redistributed implementation of the trust management service. The practicability and edges of our approach are valid by an example and experimental studies employing assortment of real-world trust feedbacks on cloud services.**

***Keywords:*--Cloud computing, credibility feedbacks, trust management, security, privacy**

I. INTRODUCTION

In highly elastic nature of cloud platforms trust management is one of big challenge. As per the research study it is one of top 10 threats of cloud environments. The SLA in between cloud users and cloud providers is not sufficient to create the trust.

Consumers feedback is a good source to asses overall trustworthiness of cloud services. Different researchers provide solutions on feedback based trust management of cloud services. In real time, the service provides experiences malicious attacks (Collision, Sybil). This paper presents novel approaches mainly considering below key issues of trust management of cloud services.

Consumer's privacy. Security is the main concern to maintain sensitive and behavioural information of a customer.

Cloud Services Protection. CSP are protecting from the malicious behaviour of attackers.

Trust Management Service's Availability. A trust management service (TMS) provides an interface between users and cloud services for effective trust management.

In this paper we present a framework for a reputation- based trust management for cloud environments. CloudArmor exploits techniques to identify credible feedbacks from malicious ones. It mainly focuses on zero knowledge credibility proof protocol, credibility model and availability model of TMS.

II. RELATED WORK

Previously we have different types of trust management techniques proposed by researchers to provide trust between cloud users and cloud service providers [12]. Some of the efforts on policy based trust management methods. For instance, Ko et al. [2] propose Trust Cloud framework for responsibility and trust in cloud computing. Particularly, Trust Cloud consists of 5 layers as well as progress, data, system, policies and laws, and laws layers to deal with accountability within the cloud surroundings. All of those layers maintain the cloud responsibility life cycle that consists of seven steps as well as policy coming up with, sense and trace, logging, safe-keeping of logs, reporting and replaying, auditing, and optimizing and rectifying. Brandic et al. [11] propose a completely unique approach for compliance management in cloud environments to establish trust between totally different parties. The approach is developed employing a centralized design and uses compliant management technique to ascertain trust between cloud service users and cloud service suppliers. Unlike previous works that use policy-based trust management techniques, we tend to assess the trait of cloud service mistreatment reputation-based trust management techniques. Name represents a high influence that cloud service users have over the trust management system [1], particularly that the opinions of the assorted cloud service users will dramatically influence the name of a cloud service either absolutely or negatively.

Unlike previous works that don't think about the matter of unpredictable name attacks against cloud services, we have already a tendency to gift a believability model that not solely detects the dishonourable trust feedbacks from collusion and Sybil attacks, however additionally has the flexibility to adaptively adjust the trust results for cloud services that are affected by malicious behaviour.

III. FRAMEWORK AND IMPLEMENTATION

The Cloud Armor framework works trust as a service based on SOA. The cloud providers have different services to its users. The framework with different layers provides trust management in distributed nodes and that gives expose to user fronted provide feedback and inquire trust results. The framework is dividing into three layers.

Cloud Services Providers Layer. This layer gathers information about diverse services publicly provide by the CSPs. This layer communicates with users and TMS, and marketing information cloud service provides on Web.

The Trust Management Service Layer. This layer provides the main functionality of the frame work like feedback credibility assessment through distributed trust nodes as well as their availability of trust nodes in each region. The interactions of the layer are with CSP, users and Cloud service advertisement to advertise the trust as a service to uses through the Internet.

The Cloud Service Consumer Layer. The third and final layers of the frame work an interface for the different users who uses cloud services. Its interaction mainly on service discovery of cloud service providers, users trust, feedback and trust results of particular cloud. The services are provision only for registered users.

Our framework additionally exploits an online locomotion approach for automatic cloud services discovery, where cloud services square measure mechanically discovered on the net and keep during a cloud services repository. More ever, our framework contains an Identity Management Service (see Figure 1) that is answerable for the registration of users credentials before exploitation TMS and proving the believability of a specific consumer's feedback through ZKC2P.

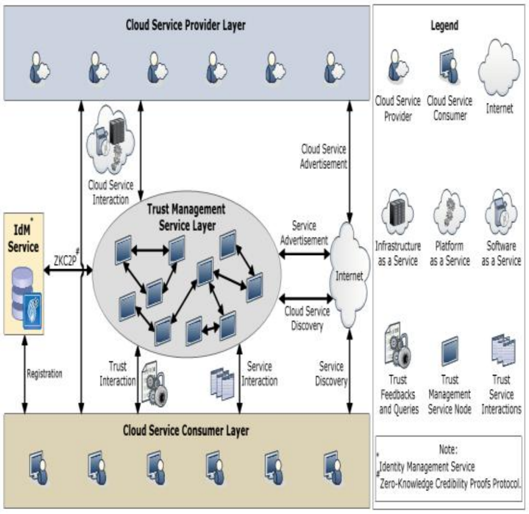

Cloud Service Provider Layer
Legend
Cloud Service Provider
Cloud Service Consumer
Internet
Cloud Service Advertisement
Cloud Service Interaction
Trust Management Service Layer
IdM* Service
ZKC2P#
Service Advertisement
Internet
Cloud Service Discovery
Infrastructure as a Service
Platform as a Service
Software as a Service
Registration
Trust Interaction
Service Interaction
Service Discovery
Trust Feedbacks and Queries
Trust Management Service Node
Trust Service Interactions
Cloud Service Consumer Layer
Note:
*Identity Management Service
#Zero-Knowledge Credibility Proofs Protocol.

Fig. 1. Architecture of CloudArmor Trust Management Framework

A. Trust Management Services

The trust management of the user depends on the identity of user and feedback credibility of user. To work on this TMS uses Zero-Knowledge credibility Proof Protocol (ZKC2P) to permit TMS to method IdM's data (i.e., credentials) victimization the Multi-Identity Recognition issue. In alternative words, TMS can prove the users' fee- dback believability while not knowing the users' credentials. TMS processes credentials while not together with the sensitive data. Instead, anonymized data is employed via consistent hashing (e.g., sha-256). The anonymization method covers all the credentials' attributes except the Timestamps attribute.

Identity management service can provide services to TMS in the detection of Sybil attacks against cloud services without breaking privacy of users. When user access the TMS it register all their credentials for the first time to store their identity. In reputation-based TMS a user provide the feedback about trustworthiness of cloud service or ask for suggestion of the service. It all depends on the collection of history records.

B. Credibility Model

Our projected quality model with feedback collusion detection and Sybil attack detection. Feedback collusion detection depends on feedback density malicious users offer fake feedbacks to control the trust results for cloud services. To beat this drawback we tend to invent a brand new approach feedback density to support determination of credible trust feedbacks. The trust feedback depends on the feedback volume and feedback mass.

Sybil attack depends on the trust identity written record we tend to believe that Multi-Identity Reorganization is applicable by scrutiny the values of user certificate attributes from Identity records. We tend to construct to search out the frequency of the certificate attributes values for a similar specific shopper Vc, t with in the same certificate attribute. We discover the buyer has fairly distinctive credentials.

C. Availability Model

The trust management services availability in all circumstance a different challenge due to count less no of request to TMS at a give time as well as unpredictable behaviours of cloud systems. In this framework we propose availability model which includes operational power calculation to share the work load and replication determination to lower the failures of TMS nodes.

The operational power factor of distributed TMS nodes calculated to find the work load for a particular TMS and with average work load of TMS nodes. The operational power of particular TMS node Op (S tms) is calculated the mean of Euclidean distance.

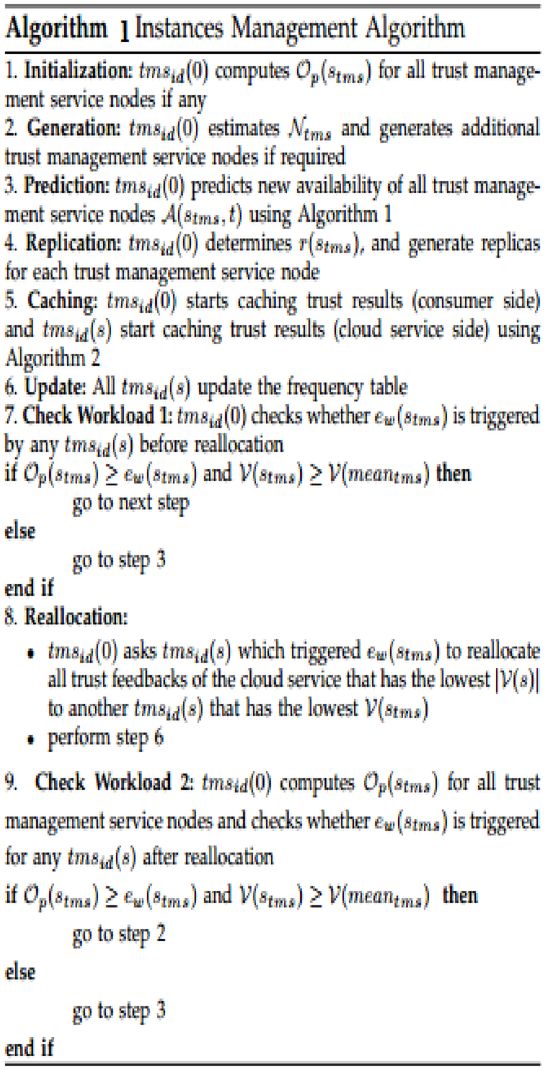
We additionally think about the replication techniques to reduce the likelihood of the crashing of node, hosting a TMS instance to confirm that users will provide trust feedbacks or request a trust management for cloud services. Replication permits TMS instance recover any lost throughout the down time from its replica. Especially, we have a tendency to propose a particle filtering approach exactly predict convenience of every node hosting a TMS instance that then wont to deter mine the best range of the TMS instance's replicas. To predict the provision of every node, we have a tendency to model the TMS instance as an instant purpose convenience.

D. Distributed Instances Management:

Our proposed CloudArmor framework TMS node act as a master while the remaining works as secondary or normal instances. The master is responsible for optimal no of nodes estimation, feedbacks reallocation, trust resulting cache, availability of each node prediction and TMS node replication. The secondary instances square measure accountable for trust management and feedback storage, result caching and frequency table update. Algorithm 1 shows the brief process on how TMS instances are managed.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 4, April 2017



```
Algorithm 1 Instances Management Algorithm
1. Initialization: $tms_{id}(0)$ computes $O_p(s_{tms})$ for all trust management service nodes if any
2. Generation: $tms_{id}(0)$ estimates $N_{tms}$ and generates additional trust management service nodes if required
3. Prediction: $tms_{id}(0)$ predicts new availability of all trust management service nodes $A(s_{tms}, t)$ using Algorithm 1
4. Replication: $tms_{id}(0)$ determines $r(s_{tms})$, and generate replicas for each trust management service node
5. Caching: $tms_{id}(0)$ starts caching trust results (consumer side) and $tms_{id}(s)$ start caching trust results (cloud service side) using Algorithm 2
6. Update: All $tms_{id}(s)$ update the frequency table
7. Check Workload 1: $tms_{id}(0)$ checks whether $e_w(s_{tms})$ is triggered by any $tms_{id}(s)$ before reallocation
if $O_p(s_{tms}) \geq e_w(s_{tms})$ and $V(s_{tms}) \geq V(mean_{tms})$ then
        go to next step
else
        go to step 3
end if
8. Reallocation:
   • $tms_{id}(0)$ asks $tms_{id}(s)$ which triggered $e_w(s_{tms})$ to reallocate all trust feedbacks of the cloud service that has the lowest $|V(s)|$ to another $tms_{id}(s)$ that has the lowest $V(s_{tms})$
   • perform step 6
9. Check Workload 2: $tms_{id}(0)$ computes $O_p(s_{tms})$ for all trust management service nodes and checks whether $e_w(s_{tms})$ is triggered for any $tms_{id}(s)$ after reallocation
if $O_p(s_{tms}) \geq e_w(s_{tms})$ and $V(s_{tms}) \geq V(mean_{tms})$ then
        go to step 2
else
        go to step 3
end if
```

In this approach, each TMS node works on feedbacks given to a set of cloud services each TMS node works on feedbacks give to a set of cloud services and modify the frequency table. The frequency records show the information about feedbacks handled and responsible for which cloud service. Example 1 explains how feedbacks can be reallocated from one TMS instance to a different instance. In this example, there are three TMS instances and the workload threshold e_w (s_{tms}) is set to 50 percent. TMS instance tms$_{id}$(1) threshold, therefore according to Algorithm 1, the trust feedbacks for the cloud service (2) are reallocated to tms$_{id}$(2), which has the lowest feedbacks.

TABLE I

Example 1: Reallocation (e$_w$(s$_{tms}$) = 50%)
Frequency Table Before Reallocation (Step 1) (tms$_{id}$(1), |υ(1) |: 200, |υ(2) |: 150, |υ(3) |: 195) (tms$_{id}$(2), |υ(4) |: 30, |υ(5) |: 20, |υ(6) |: 45) (tms$_{id}$(3), |υ(7) |: 90, |υ(8) |: 35, |υ(9) |: 95) Check Workload (Step 2) (tms$_{id}$(1), O$_p$(1tms): 0.617) (tms$_{id}$(2), O$_p$(2tms): 0.278) (tms$_{id}$(3), O$_p$(3tms): 0.205) Frequency Table After Reallocation (Step 3) (tms$_{id}$(1), |υ(1) |: 200, |υ(3) |: 195) (tms$_{id}$(2), |υ(2) |: 150, |υ(4) |: 30, |υ(5) |: 20, |υ(6) |: 45) (tms$_{id}$(3), |υ(7) |: 90, |υ(8) |: 35, |υ(9) |: 95)

IV. EXPERIMENTAL EVALUATION AND SETUP

In this section we report system setup and experimental evaluation to validate proposed system results.

A. System Setup

Specifically, the trust management service (TMS) consists of 2 main components: the Trust Data Provisioning and also the Trust Assessment Function.

I) The Trust Information Provisioning:

This part is chargeable for aggregation cloud services and trust data. We have a tendency to developed the Cloud Services Crawler module supported the Open Source Web Crawler for Java (crawler4j3) and extended it to permit the platform to mechanically discover cloud services on the net. We implemented a set of functionalities to modify the crawling process and created the crawled information additional comprehensive. In addition, we developed the Trust Feedbacks Collector module to gather feedbacks directly from users within the kind of history records and stored them in the Trust Feedbacks Database. Indeed, users usually outright to establish their identities for the first time they attempt to use the platform through registering their credentials at the Identity Management Service (IdM) which stores the credentials in the Trust Identity Registry. Moreover, we have a tendency to develop the Identity information Collector module to gather the whole variety of established identities among the full identity.

II) The Trust Assessment Perform:

This perform is answerable for handling trust assessment requests from users wherever the trait of cloud services are compared and also the factors of trust feedbacks are calculated (i.e., the quality factors). We tend to developed the Factors Calculator for attacks detection supported a group of things. Moreover, we tend to develop the Trust administrative official to check the trait of cloud services through requesting the collective factors weights from the Factors Calculator to weigh feedbacks and so calculate the mean of all feedbacks given to every cloud service. The trust results for every cloud service and also the factors' weights for trust feedbacks are hold on within the Trust Results and Factors Weights Storage.

B. Experimental Evaluation

We mainly focused on the strong implementation proposed system against different types of attacks like Sybil and collision. For experimental functions, the collected information was divide into six streams of cloud services, 3 of that were used to validate the believability model against collusion attacks, and also the different 3 teams were used to validate the model against Sybil attacks wherever every cluster consists of a hundred users. Every cloud service cluster was used to represent a distinct assaultive behaviour model, namely: Waves, Uniform and Peaks as shown in Figure 2. The behaviour model represent the full range of malicious feedbacks introduced in a very specific time instance (e.g. (e.g., jV(s)j = sixty malicious feedbacks once Tf= 40, Figure 3(a)) once experimenting against collusion attacks. The behaviour models conjointly represent the full range of identities established by attackers in a very amount of your time (e.g., jI(s)j = seventy eight malicious identities once Ti = 20, Figure 3(c)) wherever one malicious feedback is introduced per identity once experimenting against Sybil attacks. In collusion attacks, we have a tendency to simulated malicious feedback to extend trust results of cloud services (i.e., self-promoting attack) whereas in Sybil attacks we have a tendency to simulated malicious feedback to decrease trust results (i.e. slandering attack). To gauge gauge the hardiness of our believability model with relation to malicious behaviours (i.e., collusion and Sybil attacks), we have a tendency to used 2 experimental settings: I) measuring the hardiness of the believability model with a traditional model Con(s, t0, t) (i.e. turning Cr(c, s, t0, t) to one for all trust feedbacks), and II) measuring the performance of our model exploitation 2 measures specifically preciseness (i.e. however well TMS did in police work attacks) and recall (i.e. what percentage attacks square measure actual attacks). In our experiments, TMS started profitable cloud services that had been tormented by malicious behaviours once the attacks share reached 25%, therefore the profitable method would occur only if there was a major injury within the trust result.

For the collusion attack, we simulated malicious users to extend trust results of cloud services For the collusion attacks, we tend to simulated malicious users to extend trust results of cloud services (i.e., self promoting attack) by giving feedback with the range of [0.8, 1.0]. Figure 2 depicts the analysis of six experiments that were conducted to gauge the strength of our model with reference to collusion attacks. We note that the closer to 100 the time instance is, the higher the trust results are when the trust is calculated using the standard model. This happens as a result of malicious users is giving dishonourable feedback to extend the trust result for the cloud service.

On the opposite hand, the trust shows nearly on amendment once calculated victimization the planned planned credibleness model (Figure 2). This demonstrates that our credibility model is sensitive to collusion attacks and is ready to detect such malicious behaviours.

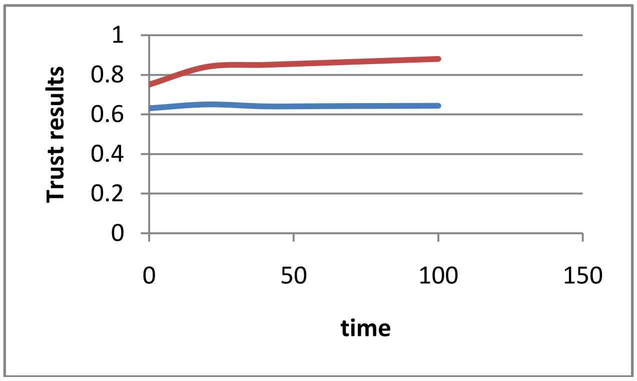


Fig 2. Robustness against collusion attacks

For the Sybil attacks experiments, we have a tendency to simulated malicious users to decrease trust results of cloud services (i.e., slandering attack) by establishing multiple identities and giving one malicious feedback with the vary of [0, 0.2] per identity. Figure 3 depicts the analysis of six experiments that were conducted to judge the lustiness of our model with reference to Sybil attacks. From Figure five, we will observe that trust results obtained by exploitation the traditional model decrease once the time instance becomes nearer to one hundred. this is often as a result of malicious users UN agency square measure giving dishonest feedback to decrease the trust result for the cloud service. On the opposite hand, trust results obtained by exploitation our projected credibleness model square measure beyond those obtained by exploitation the traditional model (Figure 3). This is often as a result of the cloud service was rewarded once the attacks occurred.

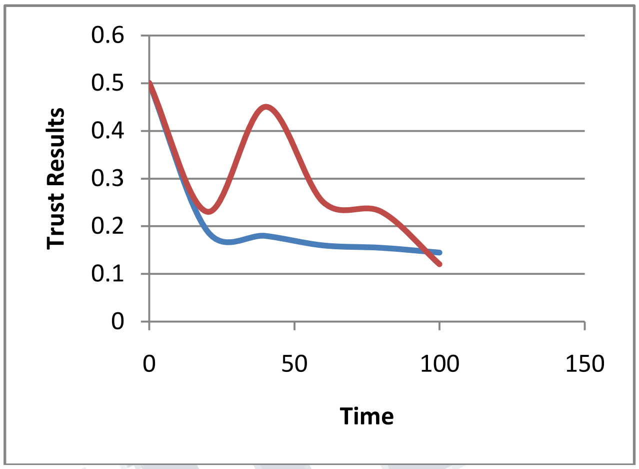


Fig 3. Robustness against Sybil attacks

V. CONCLUSION

Inconsistence nature of cloud systems are maintaining and improving trust between cloud consumers and CSP a significant challenge. Cloud service clients feedback is a good asset to examine overall trustworthy of malicious user may group together to i) mislead trust feedback which is a disadvantage cloud services or ii) by maintaining the multiple accounts to trick the trustworthy of cloud services and mislead feedback. In this paper, we have a tendency to design new approach that checks the reputation-based attacks and allowing users to effectively identify trustworthy cloud services. In this we also presented credibility model which identifies malicious attacks (collusion and Sybil attack). We also consider the trust management server and its associated nodes operational power of provide high uptime and maintaince. The result analysis shows that our framework capability of detecting malicious behaviour.

REFERENCES

[1] C. Dellarocas, “The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms,” Management Science, vol. 49, no. 10, pp. 1407–1424, 2003.

[2] R. Ko et al., “TrustCloud: A Framework for Accountability and Trust in Cloud Computing,” in Proc. SERVICES’11, 2011.

[3] K. Hoffman, D. Zage, and C. Nita-Rotaru, “A Survey of Attack and Defense Techniques for Reputation Systems,” ACM Computing Surveys, vol. 42, no. 1, pp. 1–31, 2009.

[4] S. M. Kim and M.-C. Rosu, “A Survey of Public Web Services,” in Proc. of WWW04, 2004.

[5] S. M. Khan and K. W. Hamlen, “Hatman: Intra-Cloud Trust Management for Hadoop,” in *Proc. CLOUD’12*, 2012.

[6] S. Pearson, “Privacy, Security and Trust in Cloud Computing,” in Privacy and Security for Cloud Computing, ser. Computer Communications and Networks, 2013, pp. 3–42.

[7] J. Huang and D. M. Nicol, “Trust Mechanisms for Cloud Computing,” Journal of Cloud Computing, vol. 2, no. 1, pp. 1–14, 2013.

[8] K. Hwang and D. Li, “Trusted Cloud Computing with Secure Resources and Data Coloring,” IEEE Internet Computing, vol. 14,no. 5, pp. 14–22, 2010.

[9] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A View of Cloud Computing,” Communications of the ACM, vol. 53,no. 4, pp. 50–58, 2010.

[10] S. Habib, S. Ries, and M. Muhlhauser, “Towards a Trust Management System for Cloud Computing,” in Proc. of TrustCom’11,2011.

[11] I. Brandic, S. Dustdar, T. Anstett, D. Schumm, F. Leymann, and R. Konrad, “Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds,” in Proc. of CLOUD’10, 2010.

[12] W. Conner, A. Iyengar, T. Mikalsen, I. Rouvellou, and K. Nahrstedt, “A Trust Management Framework for Service-Oriented Environments,” in Proc. of WWW’09, 2009.