Received: November 25, 2016

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Trusted Computing Model with Attestation to Assure Security for Software Services in a Cloud Environment

Udhayakumar Shanmugam[1]* **Latha Tamilselvan[2]**

[1]*School of Computer, Information and Mathematical Science,*
B.S. Abdur Rahman University, Vandalur, Chennai 600048, Tamil Nadu, India
[2]*Department of Information Technology, School of Computer, Information and Mathematical Science,*
B.S. Abdur Rahman University, Vandalur, Chennai 600048, Tamil Nadu, India
* Corresponding author's Email: mailtoudhay@gmail.com

Abstract: Predicting the behavioural patterns of an ambiguous environment is a complex task that could risk the integrity of security architecture if left unnoticed. One such ambiguous environment is the cloud computing paradigm where computations are executed remotely in geographically dispersed decentralized data centers, and access to resources distributed beyond a definable and controlled perimeter. Nevertheless, the consumer's confidence in dependable and trustworthy services is still uncertain as a result of security concerns encompassing the cloud. Deploying trusted computing models that can assure security could considerably improve the average consumer's perspective. Our proposed model emphasises an attestation procedure for trust evaluation, measuring and auditing the integrity of the system through a body of evidence. The model implements an attestation rule engine based on a Cloud Attestation Protocol (CAP). The approach for monitoring the behaviour and certifying it for attestation enables the users to select the services with trustworthiness.

Keywords: Trusted cloud computing, Service attestation, QoS trust evaluation, Cloud security, Attestation as a service.

1. Introduction

The convergence of the internet and service-oriented computing has created a massive prediction for on-demand services everywhere and at any time. Traditionally delivered in-house computing and storage tasks have now been largely integrated or replaced by online service providers like Amazon, Google and Microsoft [1]. This resource leasing infrastructure has lowered barriers to access to cheap pay-for-use models of scalable and customized resources, leading to the advent of a new computing paradigm called cloud computing. Information technology's agility and reliability have improved enormously as a result of multi-core servers for high-end computations, networks for virtual connectivity, reliable data storage, and a wide range of software applications. Now, these tailor-made services are available as shared resources in a cloud. The underlying technology and infrastructure of the cloud are an amalgamation of miscellaneous computing paradigms. This computing has evolved through the time of internet computing. The most important of these are the virtualization of High-Performance Computing (HPC) and Service Oriented Architecture (SOA). SOA provides a rule-based service delivery model in a network-centric computing environment, and virtualization is all about abstracting hardware and resources from its operating system. This combination of technology delivers multiple services online: software, platform, and infrastructure. Software as a Service (SaaS) is delivered through the user-interface only model, while Platform as a Service (PaaS) helps to develop applications. Virtual servers with unique IP addresses take care of Infrastructure as a Service (IaaS). Every service offered by the cloud must

support the existing services above so as to make the cloud environment function in accordance with Service Level Agreements (SLA). Security concerns in the cloud are a major roadblock to the full adoption of the cloud. This concern is substantiated in a recent survey [2] conducted among 3000 cloud consumers shows that 84% are concerned about the location of their data, while 88% worry about accessibility to their data. An invaluable service offered in support is Security as a Service (SecaaS). It is primarily involved in protecting user's data in the cloud where there are no traditional boundaries [3]. The issue of cloud security arises because, in the cloud, all data are sent to remote destinations for the carrying out of any computation. Sending data in an unsecured environment puts user's privacy over their data and its integrity at risk. Hence, the environment has less control and visibility, leading to question the cloud service provider's trustworthiness.

The objective of our model is to measure the integrity of software services supplied by the provider at the host's location through an extensible set of measurement modules. The behaviour of the services is analysed for any deviations and trust score for the services is assessed. As an addition to the behaviour assessment the proposed model verify the process through an attestation engine. Further, we also affirm that the service so provided is indeed employed by a trusted user after checking on client behaviour. By providing trusted services, clients, and the providers, we have proposed a novel method to provide an end-to-end trust management system to assess the software services. The Cloud Attestation Protocol (CAP) guarantees the trust model by cross-examining the trust values through an independent and transparent process. The model utilizes Analytical Hierarchy Processing (AHP) for calculating the weightage for each metrics. These features ensure that the trusted computing model can assure confidence among the cloud stakeholders to facilitate a trusted environment.

To substantiate the trustworthiness, the model is experimented and tested through a real-time cloud setup using OpenStack. It is an industry standard open sourced cloud middleware adopted by most of the cloud providers. The results have proved that the model has the best availability with consistent trust value. The accuracy to identify trusted and untrusted services are good in comparison to the existing system. Moreover, since most of the existing systems are either being proposed as a theoretical model or being simulated, our real-time implementation model has the advantage of being a true indicator of trust score.

The rest of the paper is organized as follows: In the next subsection, information on assorted security threats and attacks is given. Trust semantics and the need for trusted computing in the cloud are also discussed. The literature on trusted cloud computing and attestation models is reviewed in Section 2. Section 3 follows with an understanding of attestation models and the proposed security architecture. In Section 4, protocols and algorithms that define the model are discussed. Finally, results and analysis proving the trustworthiness of cloud providers take up Section 5.

2. Related Works

Cloud security has recently received widespread attention from experts across domains, leading to a large number of research projects being undertaken to improve Privacy, Security, and Trust (PST). In this chapter, we survey various trust models that have contributed to our research domain.

The Trusted Computing Group (TCG) specifies important functions for IT security and focuses chiefly towards providing cloud security through Trusted Multi-tenant Infrastructure (TMI) [4]. A key security aspect contributed by TCG is the development of a TPM chip (Trusted Platform Module), a secure cryptoprocessor perform platform authentication, disk encryption and password protection. Thus, in our proposed work, it is assumed that the host accessing cloud services must connect via a TPM. In Jingwei Huang [5], the author describes how stakeholders in the cloud can be verified with parameters and criterion like accreditation, policy compliance audit, certificate attributes and reputation. The TR model by Sheikh et al. [6], describes trust as a facilitator in the cloud that integrates Quality of Services (QoS) parameters like compliance, interoperability, customer support, federated identity management, and service deployment. Though these QoS parameters reflect the provider's capability, it lacks judging the security and privacy capability of the service instance code. In DR@FT by Wenjuan et al. [7], a domain-based integrity model where integrity is measured based on information flow, is presented. It classifies high-integrity and low-integrity processes and verifies the latest changes in a target system. AdapTest [8] projects an attestation framework for a multi-tenant cloud system which reduces attestation overheads and shortens detection delays. These attestation models have been developed for virtual machines, however there is a need to attest the application and users trust score, which our model incorporates. A formal trust metrics for a messaging

service in the hybrid cloud that classifies trusted components based on the reputation of consumers through weights has been proposed by Udhayakumar et al. [9].

Similar trust models and frameworks like Bayesian models, Eigen Trust, reputation, behaviour, probability and credential-based trust models have been discussed in earlier studies. These approaches have focused chiefly on a peer-to-peer network, wireless sensor networks [10], [11] and mobile computing [12]. Thus the drawbacks in these conventional techniques are these models evaluate the trust as a subjective logic with indirect trust assessment. Moreover, existing works are mostly towards selection of trusted services from providers; they neither do assure the trust nor certify it. To overcome these, our model evaluates the trust scored by the service instance as well as the user through objective parameters with direct trust evidences. Further, it supports an attestation process with a central attestation engine to authenticate services and provide end users certificates.

3. Trusted Software Service Model

The growing importance of cloud computing makes it imperative for consumers, providers and society in general to establish trust. Establishing confidence with reputation systems has been successfully used since the inception of the internet to support users identifies trustworthy service providers. Trust revolves around the assurance and confidence that people, data, entities, information or processes will function or behave in expected ways. Trust is defined as '*the belief the trusting agent has in the service provider's willingness and capability to deliver a mutually-agreed service in a given context and in a given time slot'* [5]. Trust can be of two types, based on the trustor's expectancy: *trust in performance and trust in belief*. A *trust in performance* can be expressed as...

$$trust\_p(d,e,x,k) \equiv madeby(x,e,k) \supset believe(d,k \supset x) \quad (1)$$

Here the trustee's performance is described through first-order logic in equation 1, where *d* trustor's trust and *x* is the performance of trustee *e* in a particular context *k*. This relationship means that if *x* is made by *e* in context *k*, then d believes *x* in that context.

A *trust in belief* expressed in equation 2, is about what the trustee believes: let *trust_b(d, e, x, k)* represent that trustor *d* trusts trustee *e* regarding *e's belief (x)* in context *k*. It means if *e* believes *x* in context *k*, then *d* also believes *x* in that context:

$$trust\_b(d,e,x,k) \equiv believe(e,k \supset x) \supset believe(d,k \supset x) \quad (2)$$

Trust in belief is transitive, trust in performance is not: however, trust in performance can be propagated through trust in belief. It is yet another grand challenge to make truly trustworthy cloud computing environment.

3.1. Trusted Service Execution

Since the cloud delivery model is constructed on service-oriented architecture, we have multiple trust nodes for cloud consumers and cloud providers, channelized through cloud brokers. This model allows the setting up of various SLAs and privacy assurance strategies. Trust changes dynamically, and hence the system should be adaptive to changes, requiring that the system monitors its behaviour in every instance. Trusted service so provided follows a path to make itself trustworthy through a layered approach, as depicted in Figure.1. Cloud Services (CSs) in remote cloud data centers are accessed by Cloud Consumers (CCs) through a procedure of service requests. This initiates a search process from Cloud Brokers (CBs) to get a range of similar services available in the cloud. Cloud Providers (CPs) specify certain attributes that form a particular user's SLA [13]. The featured attributes are advertised and users get responses from the provider about compliance, privacy, and integrity. Once the consumer has finalized the choice of service - based on the service provider's reputation, the brand value, or the user's previous experience - the service so requested is provisioned. Now the brokers help the auditors to perform sundry intermediate services.

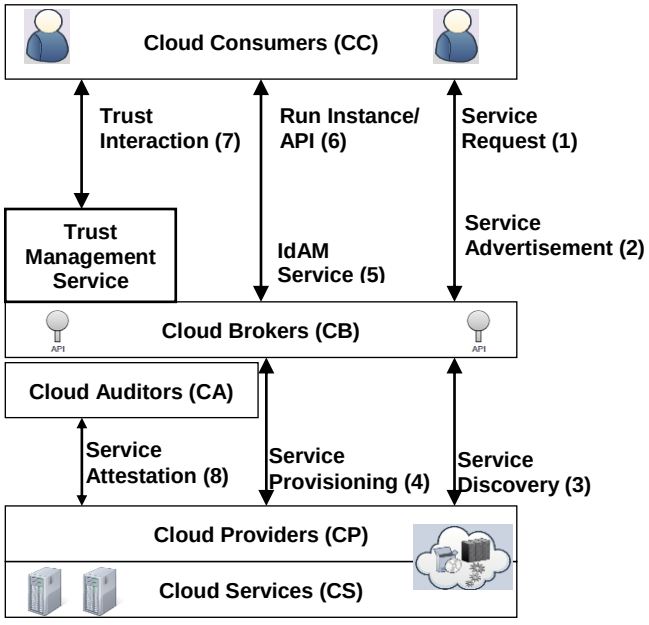

Figure.1 Cloud Architecture for Service Execution

In our approach, brokers execute the Identity Management (IdM) process for authenticating consumers before the service is instantiated by a cloud consumer. Once the service is in use, cloud auditor's(CA) coordinating with CB's form a trust evaluation framework to compute the service's trustworthiness, measuring it by means of a performance monitor. The CA performs an attribute assessment, usually regarded as the reliable information source for trust judgment. Being an important entity in the cloud, CAs ensures trustworthiness through the process of service attestation, where a certificate of acceptance is issued for the Quality of Service (QoS) delivered by the service provider. Since auditors perform an independent assessment of services, a third-party professional organization accredited by an auditing standards board or a national standards body, or a professional association would indubitably make the best CAs [14]. The reason behind the selection of a third-party auditor is because an internal process might lead to collusion among brokers, auditors, and providers, leading to a conspiracy to better performance through a false reputation. Finally, the trust management service governs trust gained through a process of iteration and orders the most trusted services.

3.2 Centralized Trust Evaluation Framework

In our approach, we propose a new evaluation system that measures trust and an attestation protocol from a third party attestation engine. The process flow diagram in Fig.2 shows that if a service is to be trusted, it must initially run through miscellaneous integrity checks.

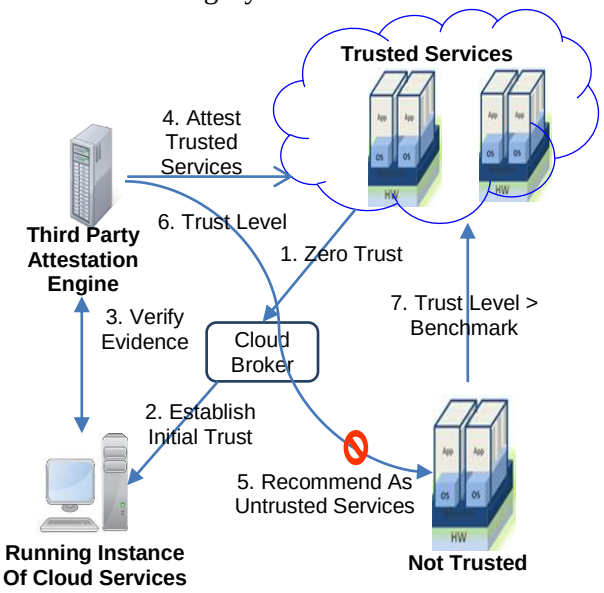

Figure. 2 Trusted Computing Model with Attestation

An evidence-based trust evaluation and a recommendation engine are two chief mechanisms needed for a trust model. Our framework initially collects service parameters like bandwidth, boot processes, system requirements, browser settings and network controllers and stores them in a benchmark server. After having set initial boundary conditions with available parameters, it refines itself for the next instance by verification with the set benchmarks. In this way, evidence of integrity is collected for a particular context to prove whether or not the service can be trusted.

3.3 Evidence Based Trust Attributes

Evidence can be a chain of trust judgements in relation to earlier runs or with respect to trustees' trusted entity. To express this, let us adopt a scenario from [5], where, in a particular context in time T_c, a cloud user C_c trusts a trustee's attributes T_a to make a claim about a service S that has attributes A with parameter value P_v, .Then, when a specific assertion for a service $A(S, P_v)$ is made in a context T_c, C_c believes in the claim as shown in the below expression.

$$EvT(C_c,T_a\ A(S,P_v), T_c) \wedge madeBy(A(S,P_v), T_a, T_c) \wedge in(T_c) \rightarrow believe(C_c, A(S,P_v)) \quad (3)$$

Hence, in evidence-based trust, an attribute trusted by a trustee can also be trusted by a cloud user in a particular context. The evidence so collected is shown in Table 1. Configuration, speed, and flexibility are stored in benchmark servers for evaluation of initial conditions prior to accepting services to be delivered to users.

Table 1. QoS attributes

QoS Attributes	Parameters	Service	Context
Speed	Data rate and Bandwidth	Initial Request	Peak /Non-Peak Time
Availability	Avg. no. of successful service	Response to request	Under lesser loads and heavier loads
Accuracy	No. of services completed	Completeness of service	Successful usage of service
Usability	Feedback values	Satisfied service	Satisfaction
Flexibility	Acceptance of special requirements	New service offerings	Acceptance under failure of service
Configuration	Hardware and software requirements	Service running conditions	Service utilization

Table 2. Security attributes

Security Attributes	Parameters	Behaviour
Confidentiality	Availability of TPM	Possibility of attacks
Authorization	Avg. no. times trying to access root	Attempt to access unauthorized services
Authentication	No. of failed sign-on	No. of times login attempt threshold
Certification	Any third-party certificates	Evidence and proof of behaviour
Audit logs	Abnormal timing of access	Simultaneous login
IP security	Abnormal IP address and proxy changing patterns	Check for IP spoofing attacks

A well-established Service-Level Agreement (SLA) that reflects the quality of service being offered by the CSP needs to be agreed upon at the initial point of access. In Table 2, the consumer is being evaluated with various security features and its behaviour. For example availability of TPM as a BIOS level encryption mechanism and 3rd party certificates can help in judging the user's security protection mechanism. Further, the behavioural pattern of a consumer can be assessed if a person is trying to access the cloud service at different locations or with different IP number. Authorization and authentication checks the person's intention to attack the system's integrity. Malicious services are restricted by cloud brokers and barely-trusted ones are given additional support to perform well by cooperating with peer groups. Thus, through an iterative process of evaluation and recommendation, services are pooled for trustworthiness. In the following section, we present the implementation details of the cloud model with preliminary evaluations and conditions for trustful services access.

4. Implementation and Results

The trusted computing model is set up in a private cloud environment using Open Stack implementation supporting the infrastructure of a quad-core processor with 8GB memory for the compute node, and a quad-core processor with 4GB memory for the controller node. The underlying host is Ubuntu 14.04 and the guest operating systems are Fedora 19 and Ubuntu for the compute node.

The application is now loaded in the Ubuntu web server. Once a node accesses the application, the plugin is initialized to get it loaded at the client's browser and obtain all the client node's measurements. Information on the successful initial load status is checked with the benchmark to initiate the request. Once acceptance is granted by the attestation engine, the attester who acts as controller or middleware communicates with the provider to start the application's instantiation. A flag set in the address bar is ready for continuous reassessment through an iterative process. The network speed is fixed at 512Kbps and the memory a minimum 1GB RAM. We have considered, for the test cases, 10 machines with varied support systems. To calculate trust, we need to identify the weightage factor for each metric and then evaluate trust based on success or failure. To assess weightage, we apply a pair-wise comparison for each metric so as to arrive at a rational decision-making process. This evaluation is achieved by applying Analytical Hierarchy Processing (AHP), a structured technique for analyzing complex decisions [15].

A Decision matrix is built as shown in Table 3, based on systematic evaluation of various elements by comparing them to each other two at a time with respect to their impact on an element above them in the hierarchy. Using these weights, we can now calculate the trust value to assess the initial check for each cloud consumer using the equation (4). Here W_i, is the weights for metrics M, and F_j is the normalized measurement for each consumers. Let, i be the total metrics and j be the total attributes.

$$Trust\ Value(check) = \sum_{i=1,j=1}^{M} Wi.Fj \tag{4}$$

Table 3. Weights with priority ranking for Initial Check

Metrics	Network Speed Wi_1	Memory Wi_2	Plug-In support Wi_3	Firewall support Wi_4	TPM chip Wi_5
Weights (W)	0.055	0.039	0.648	0.117	0.139
Priority Ranking	4	5	1	3	2

The result of the initial check for client has accepted 6 out of 10 consumers, as shown in Figure 3.

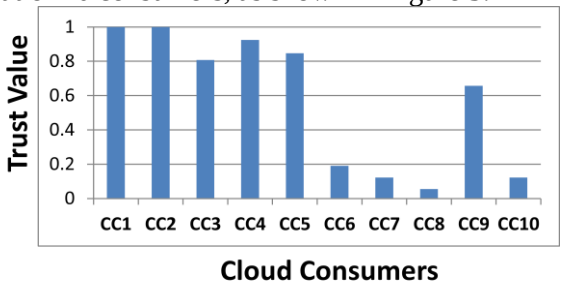

Figure. 3 Initial Check Trust Score

Table 4. Measurement for assessing trusted client

CCs	NLA	FLA	IP		SI		3PC		FBS	
	N_l	M_f	Y/N*	Fj_1	Y/N	Fj_2	Y/N	Fj_3	Y/N	Fj_4
CC1	1	Nil	Y	1	Y	1	Y	1	Y	1
CC2	1	1	Y	1	Y	1	N	0	Y	1
CC3	3	6	Y	1	Y	1	N	0	N	0
CC4	5	6	Y	1	Y	1	N	0	Y	1
CC5	1	4	Y	1	Y	1	Y	1	N	0
CC9	1	Nil	Y	1	Y	1	N	0	N	0

**Y/N – Yes or No*

Finally, a trust score in the range of 0 to 1 is calculated. The threshold is initially fixed with a value of 0.5, with anything below 0.5 being rejected access to cloud services.

Table 4, identifies the criteria used to assess the trusted client through parameters like checking the IP address (IP), time of access, and possession of a third-party certificate (3PC). Also, it checks whether the client is accessing the service with malicious intent hence it finds the number of times the login attempt has failed (FLA) M_f, the client's attempts (NLA) to gain access to multiple logins (N_l), and any change in the user's IP address are verified. User's committed to utilizing cloud services would provide feedback (FBS) on the quality of the service. Thus, in evaluating trust values, the feedback component has priority over other parameters through a better weightage factor as shown in Table 5. The positive (T_{+ve}) and negative score (T_{-ve}) for the trusted client (T_C) for K metrics are calculated using the equations 5, and equation 6.

$$Trust\ Value\ (client + ve) = \sum_{i=1,j=1}^{K} Wi.Fj \quad (5)$$

$$Trust\ Value\ (client - ve) = \left\|(N_l^2 W_l + M_f W_f) M_f\right\| \quad (6)$$

The analysis has shown that for CC3 and CC4 the access pattern is not normal, and both are found to be accessing the account using more than one login account.

Table 5. Decision matrix for positive score

Metrics	Auth. -Wi_1	SI -Wi_2	3PC -Wi_3	FBS -Wi_4
Weights	0.110	0.131	0.247	0.510
Ranking	4	3	2	1

Table 6. Decision Matrix for Client with Negative Score

Metrics	NLA W_l	FLA W_f
NLA	1	2
FLA	0.5	1
Weights(W)	0.666	0.333
Priority Ranking	1	2

Table 7. Trust evaluation based on negative score

Cloud Consumer	Negative Score T_{-ve}	Normalized Score $\widehat{T}_{-ve}$	Positive Score T_{+ve}	Trust Value	Access Status
CC1	0	0	1	**1**	Granted
CC2	1	0.001	0.751	**0.750**	Granted
CC3	47.52	0.047	0.241	**0.193**	Granted
CC4	110.88	0.110	0.751	**0.640**	Flagged for Suspicion
CC5	7.92	0.007	0.488	**0.480**	Granted
CC9	0	0	0.241	**0.241**	Granted

The weightage between the numbers of login attempts through different user names (W_l) is given more importance than the number of failed login attempts (W_f), as shown in Table 6. As more login attempts for a particular IP number can be interpreted as malicious activity, its value can be scaled accordingly to show the intensity of the activity. Based on the negative score, a benchmark is set to enable the cloud consumer to be checked and flagged for further monitoring and his suspicious behavior can be tracked at all levels. Any value greater than 100 is set as the benchmark and CC4 is flagged for suspicion, as in Table 7.

To assess service trustworthiness, it is necessary to identify metrics associated with the cloud service – such as response time (RT) - which assesses the latency essential to access service from the server. Apart from the time taken for delivery of packets, it is necessary to check the size of the cloud service to ensure whether there has been an attack in any form to embed malware, spyware or even adware. Successful service initialization (SI) and successful service completion (SC) help assess whether the client has been able to utilize the application properly till his work is completed or identify whether the service has been abruptly terminated as a result of faults like network errors, server failure, or http request faults. It is vital to know whether the application has enabled log files (LOG) to be created for the service running in the host system. If the malicious activity is executed during transactions or by the client, it then acts as evidence to support a claim made between stakeholders.

Another key aspect is the feedback, which is necessary to assess the client's opinion of the application, provided by the end user towards the service used. Today, ratings and user feedback reports (USR) act as prime enablers for future consumers of the service. Also, the number of user satisfaction report Vs total downloads is the key recommender for new users who are yet to decide on accessing similar applications. Finally, SLA decides the cloud service provider assurance.

Table 8. Decision Matrix for Service Usage

Metrics	RT	SI	SC	LOG	USF	SLA
	Wi_1	Wi_2	Wi_3	Wi_4	Wi_5	Wi_6
Weight	0.086	0.074	0.152	0.306	0.233	0.147
Ranking	5	6	3	1	2	4

Table 9. Measurements for Trustworthy Service

CCs	RT		SI		SC		LOG		USF		SLA		T_S
	E/NE	Fj_1	*S/A*	Fj_1	*C/S*	Fj_1	*Y/N*	Fj_1	*Y/N*	Fj_1	*Y/N*	Fj_1	
CC1	E	1	9/10	0.9	5/9	0.5	Y	1	Y	1	Y	1	**0.92**
CC2	E	1	8/10	0.8	8/8	1	Y	1	Y	1	Y	1	**0.98**
CC3	E	1	7/10	0.7	3/7	0.4	Y	1	N	0	Y	1	**0.65**
CC4	E	1	3/10	0.3	1/3	0.3	N	0	N	0	N	0.25	**0.19**
CC5	N	0	5/10	0.5	4/5	0.8	Y	1	N	1	Y	1	**0.84**
CC9	E	1	8/8	1	8/8	1	Y	1	Y	1	Y	1	**1**

*E-Enabled, NE –Not Enabled

In Table 8 the AHP weightage matrix is given and in Table 9 the measurements before and after service usage for all the metrics has been calculated. For every CCs, values are noted for 10 different accesses and it is normalized in the range of 0 to 1. The trust value for the service usage *(T_S)* is calculated as per the equation 7. Here W_i, is the weights assigned for *R* metrics, and F_j the individual weights for each consumers for a particular metrics, normalized according to usage levels.

$$Trust\ Value(service) = \sum_{i=1,j=1}^{R} Wi.Fj \tag{7}$$

The trust value for services is near-perfect for CC1 and CC2, and CC9 has received a value of 1 since all metrics are well above the threshold. But as response time and other parameters are poor for CC4, it has failed to prove its trustworthiness. However, since CC4 had earlier been flagged for suspicion, it is now considered a Flagged Cloud Consumer (FCC), and it is not considered for the next phase of evaluation and reassessment.

The rejection of CC4 is based on the rejection algorithm and is primarily constructed on the trust value of the FCC at the service stage FT_s, in turn evaluated based on the value of the FCC during client evaluation FT_c.

Rejection algorithm
$\forall T_c$ where T_{-ve}> 100 flag the consumer as suspect, $\perp T_{+ve}$ (independent of T_{+ve})
if
$FT_s < \frac{FTc}{2}$; $\forall FCC$ || $FT_s < 0.5$; $\forall FCC$
then
reject that FCC
else
consider for reassessment for service trust evaluation

The following graph in Figure. 4 show how each cloud entities has performed at various phases of the trust relationship.

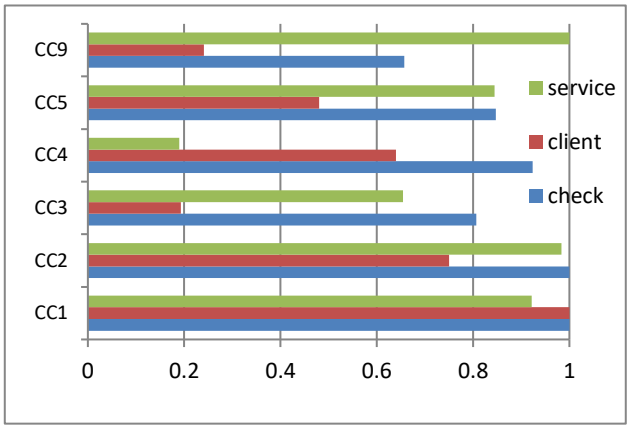

Figure. 4 Comparative Analyses of Cloud Consumers over Trust Phases

To unify trust gained at various stages into a single trust value for a trusted environment, we need to find the weightage using the AHP in Table 10. The trust value for the environment T_E is calculated using the formula below in equation 5.

$$Trust\ Value(E) = \frac{W_1 T_{IC} + W_2 T_C + W_3 T_S}{2} \tag{8}$$

The weighted trust value is assumed to be starting from the mid-point of the 0 to 1 scale in order to make trust an iterative function whereby the value can gain momentum to reach the peak value of 1 or fail to lose trust gained.

Table 10. Weightage for Trust Phase

Trust Phases	Trust for Initial Check	Trust for Client	Trust for Service Usage
Weights(W)	0.121	0.319	0.558
Priority Ranking	3	2	1

As can be seen from Table 11, the value is in the range of 0 to 0.5, enabling the function to progress accordingly.

Table 11. Final Trust Value for the Environment

Cloud Consumer	W_1T_{IC}	W_2T_C	W_3T_S	Trust Value T_E
CC1	.122	.319	.514	**0.477**
CC2	.122	.239	.549	**0.454**
CC3	.098	.061	.365	**0.262**
CC5	.102	.153	.471	**0.363**
CC9	.079	.076	.558	**0.356**

After trust for the environment is calculated, trust progresses on inputs from transactions and updates that are concurrently happening. For example, in the second iteration of service access, the user may prefer to store the log file, feedback can be properly given and the service may also be utilized to the fullest extent. Thus, we have considered two iterations for service usage and their value is proportionally added with T_E, the graph in Figure 5, showing how trust grows to reach its goal of a trustworthy cloud environment.

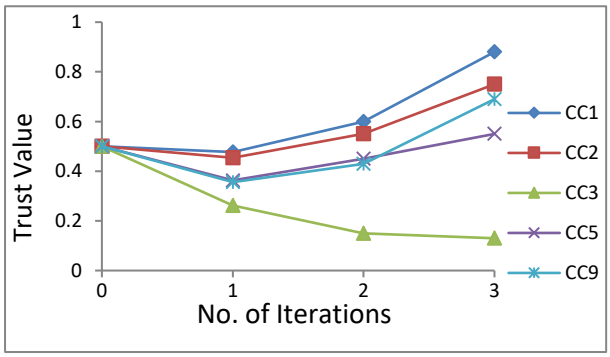

Figure. 5 Iterative trust gain factor for each Consumers

4.1 Adaptive Attestation Process

The infrastructure of the cloud provides a multi-tenant configuration in which software deployed in virtual machines is accessed by multiple users concurrently across a varied set of environments. Though verifying software through evidence can provide a means for trusted service, nevertheless the consumer would question the authenticity of a trusted service being run locally. In grid computing, remote scientists who form a Virtual Organization (VO) run remote attestation protocols to verify that a remote grid node's software environment complies with the VO's security policy [16]. Therefore, just as in the grid environment, there needs to be an attestation protocol that runs remotely over any VM

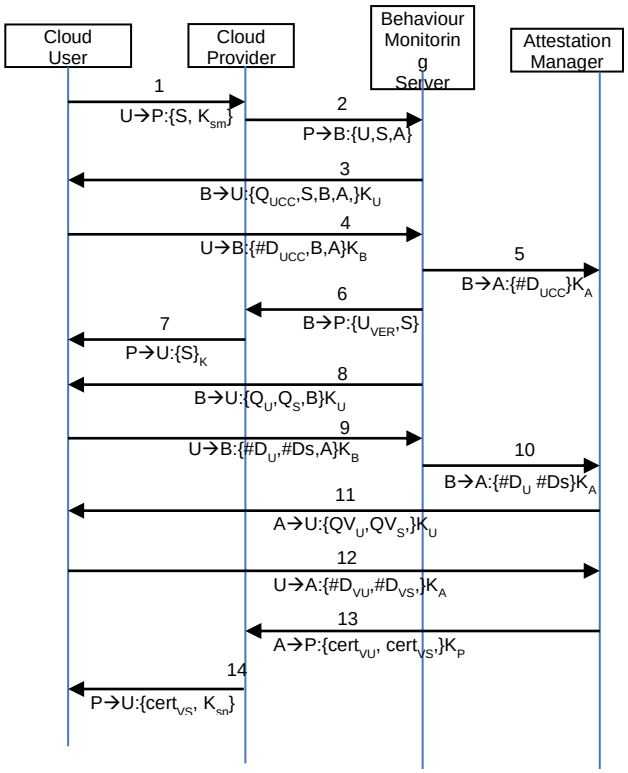

Figure. 6. Cloud Attestation Protocol

Notation	Meaning
U	User
P	Provider
S	Service
K_{sm}, K_{sn}	Session Key
A	Attestation Manager
B	Behaviour Monitor
Q_{UCC}	Query User Conf. Check
K_U,K_A,K_B,K_P	Encryption Key
$\#D_U$, $\#D_S$:, $\#D_{UC}$	Hash of Measurement Data
U_{VER}	User Verification
$\#D_{VU}$, $\#D_{VS}$	Hash of Verified Data
$cert_{VU}$, $cert_{VS}$	Certificate of Attestation

governed by a third-party certified authority to study the behaviour and categorise trustworthy applications and platforms.

Attestation is guaranteed when certain principles [17] are met. The server also holds dynamic data that changes for every iteration the change being driven by rules set in the engine based on the Cloud Attestation Protocol (CAP) given in Figure 6.

4.2 Comparison with other trust models

The comparative analysis of the proposed model with other models as shown in Table 12 proves that, attestation services are very essential in renowned trust models and the trust evaluation needs to be dynamic in nature.

Table 12. Comparative Analysis of the Existing and Proposed Trusted Computing model

Trust Models	Static /Dynamic	Trust Focus	Assessment Method	Attributes	Implementation Environment	Attestation Model
Proposed Trust Model	Dynamic	User& Services	Behavior	Ranking/ QoS	Open Stack	Adaptive
AdapTest [8]	Dynamic	User & Provider	Probability	Relationship	10 node cluster	Adaptive
Turnaround [18]	Static	Cloud Service	Reputation	QoS	Simulation Tool	Nil
Game Theory [19]	Static	Cloud User/Provider	Game Theory	SLA/Policy	Simulation	Nil
Trusted Selection [20]	Dynamic	SOA	Probability	Social Networks	Simulation	Nil
Cooperative [21]	Static	Cloud Service	Game Theory	Coalition Policy	AWS	Nil

With respect to implementation perspective, every other model have either simulated or deployed in a virtual set up, whereas our work focused on a real time cloud middleware implementation thru open stack. Using AHP as a means to weighted assessment of metrics for identifying deviation and ranking is an advantage of our model. The minimal selection of cloud users and non-assessment of service providers are the shortcomings of our model.

Time taken to decide whether the resources are trustworthy or untrustworthy, is one important overhead that needs to be compared for the worthiness of our behaviour model in Table 13. The model compares with well-known trust models like metric based [18], and the recent game theory based trust model called NEM [19]. The metric based trust uses more number of parameters resulting in increased evaluation time. Therefore, in comparison with these models our trusted cloud attestation model, evaluates the trust score based on QoS parameters and then performs an attestation process to find whether the new resources are trusted or untrusted. So the overhead is primarily due to the additional evaluation by independent server. Thus our model gives a better result of trustworthiness as shown in graph of figure 7.

Table 13. Trust evaluation time

No of Instances	TCAM (ms)	NEM (ms)	Metric Based (ms)
1	0.078	0	0.15
2	0.165	0	0.22
3	0.375	0.16	0.38
4	0.493	0.22	0.52
5	0.671	0.3	0.7
6	0.802	0.31	0.95
7	0.969	0.38	1.2
8	1.021	0.4	1.23
9	1.074	0.5	1.43
10	1.173	0.58	1.7

Similarly, the availability of the trust evaluation system is almost 97% while the NEM is with 95% and Metric based trust model is with 92%.

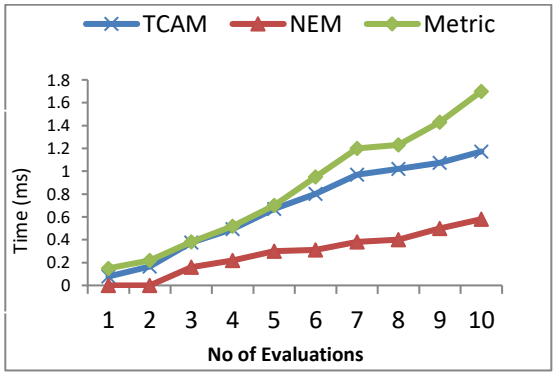

Figure 7. Graph comparing the overhead.

Thus the proposed model is by far a better approach in terms of time to evaluate and total available time for service. Therefore with the help of attestation process our model assesses, evaluates, certifies and complies for a trusted cloud service environment.

5. Conclusion

Cloud service offerings from service providers and vendors are increasing exponentially to satisfy the demands of consumers always in need of software applications to meet their requirements. In this paper, a comprehensive trust environment has been developed through a centralized trust evaluation framework and cloud attestation model. The framework identifies various metrics collected as evidence from cloud transactions for assessing the system, process and integrity of data. The attestation model is a key feature of this work, enabling the appraiser and target to have a clear view of the process and procedure to measure and attest a service. This CAP model ensures that the behaviour

of a system is predicted, based on well-known reference values kept as a threshold and, in turn, acts as a trust initiator. Further, trust is evaluated right from the initial check to the client's service usage to assess reliability and integrity. The metrics collected are quantitatively weighed and analysed using the AHP. As the model is evaluated in a real-time OpenStack private cloud environment, there is a reassurance that the trust value is an indicator of the early behaviour of the environment towards a secure cloud. Future work is focussed towards optimizing a trust algorithm to reduce time complexity and establish a Web of Trust (WoT) through a trust federation using our CAES mechanism.

References

[1] L. Leong, D. Toombs and B. Gill, "Worldwide Survey of Cloud Infrastructure Service Providers", in *by Gartner*, 2015.

[2] P. Rudlin, "Personal data in the cloud: A global survey of consumer attitudes", A *White Paper on survey on cloud computing, Fujitsu Research Institute*, 2013.

[3] J. Laundrup, M. Pohlman and D. Fielder, "Security Guidance for Critical Areas of Focus in Cloud Computing v3.0", *Security Guidance Working Group of Cloud Security Alliance*, 2011.

[4] Trusted Multi-Tenant Infrastructure Reference Framework, a *reference manual from Trusted Computing Group*, 2013.

[5] J. Huang and D. Nicol, "Trust mechanisms for cloud computing", in *Journal of Cloud Computing Advances, Systems and Applications*, 2013.

[6] S. M. Habib, "Trust as a facilitator in cloud computing: a survey", in *Journal of Cloud Computing: Advances, Systems, Applications*, 2012.

[7] W. Xu, Z. Hu, and J. Pierre, "Remote Attestation with Domain-based Integrity Model and Policy Analysis", in *IEEE Transaction on Dependable and Secure Computing*, pp. 429-442, 2012.

[8] J. Du, N. Shah and X. Gu, "Adaptive Data-Driven Service Integrity Attestation for Multi-Tenant Cloud Systems", in *Proceedings of the Nineteenth International Workshop on Quality of Service*, 2011.

[9] S. Udhayakumar, S. Chandrasekaran, T. Latha and A. Fareez, "An Adaptive Trust Model for Software Services in Hybrid Cloud Environment", *Recent Researches in Computer Science, Proceeding on 15th WSEAS International Conference on Computers*, pp. 497-502, 2011.

[10] F. Ishmanov and S. W. Kim, " A Novel Trust Establishment Method for Wireless Sensor Networks", in *KSII Transactions on Internet and Information Systems*, pp.1529-1547, Vol.9, 2015.

[11] M Zhang, C. Xu, J. Guan, R. Zheng, Q. wu and H. Zhang, "A Novel Bio-inspired Trusted Routing Protocol for Mobile Wireless Sensor Networks", *Transaction on Internet and Information Systems*, vol. 8, no. 1, pp. 74-90, 2014.

[12] A. Kumar, K. Gopal and A. Aggarwal, "Design and Analysis of Lightweight Trust Mechanism for Accessing Data in MANETs", *Transaction on Internet and Information Systems*, Vol. 8, no. 3, pp. 1119-1143, 2014.

[13] K. Saravanan and M. Rajaram, "An Exploratory Study of Cloud Service Level Agreements - State of the Art Review", in *KSII Transactions on Internet and Information Systems*, Vol. 9, No.3, pp.843-869, 2015.

[14] M. Hogan, F Lui, A. Sokol and J. Tong, "NIST Cloud Computing Standards Roadmap Version 1.0", in *Special Publication 500-291 of National Institute of Standards and Technology* , 2011.

[15] T. Evangelos and H.M. Stuart, "Using The Analytic Hierarchy Process For Decision Making In Engineering Applications", in *International Journal of Industrial Engineering: Applications,* Vol. 2, No.1, pp. 35-44, 1995.

[16] H. Chen, J. Chen, W. Mao and F. Yan, "Grid Security from Two Levels of Virtualization", in *Elsevier Journal of Information Security Technical Report*, Vol. 12, pp. 123-138, Elsevier, 2007

[17] G Coker, J. Guttman and P. Loscocco, et al., "Principles of remote attestation", in *International Journal of Information Security*, Vol. 10, Issue 2, pp. 63-81, June 2011.

[18] G. Atoosai and M. G. Arani, "A Trust Model Based on Quality of Service in Cloud Computing Environment", *In International Journal of Database Theory and Application*, Vol.8, No.5, pp.161-170, 2015.

[19] K. Gokulnath and R. Uthariaraj, "Game Theory Based Trust Model for Cloud Environment", *In the Scientific World Journal, Hindawi Publishing Corporation*, Volume 2015.

[20] C. Hang and M. Singh "Trustworthy service selection and composition", *In ACM Transactions on Autonomous and Adaptive Systems*, Vol. 6. Article 5, 2011.

[21] S. Udhayakumar and T. Latha, "A Cooperative Trust Model with Adaptive Migration for Secure Cloud Services", *In Advances in Information Sciences and Service Sciences*, Volume 6, Number 6, December, pp. 37-47, 2014.