IPHV8I1040X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

A FRAMEWORK FOR BUILDING PRIVACY TOWARDS STANDARDIZED WEB SERVICES

[#1]ERUKULA.HEMALATHA, M.Tech Student,
[#2]BS RAO, Associate Professor,
Dept of CSE,
MOTHER THERESSA COLLEGE OF ENGINEERING & TECHNOLOGY, KARIMNAGAR, T.S.,INDIA.

ABSTRACT—Enforcing a behavioral pattern in any system will force it to behave in the expected way thorough which it can be secured against any unauthorized access leading to a trusted environment. Security assurance in cloud computing environment is a major challenge associated with lack of trust and vulnerability to unauthenticated access that requires the providers to secure virtualized data centers by preserving data integrity. To improve the customer's confidence on cloud, trust has to be restored by developing trusted computing model for various cloud services ranging from storage, network, and infrastructure to everything as a service. Current trends suggest that the digital world is going to be more and more flexible, interconnected and open to public access and hence the trust associated with it has to be managed based on variety of key security techniques like identity management, digital signatures, credential exchange, certificates and key management. Nevertheless attacks on public as well as private data's in cloud ecosystem exposes the inherent failure in protection mechanism.

Keywords*—: Distributed Service, Integrity attestation, Cloud computing, Multitenant.*

I. INTRODUCTION

In recent days the cloud computing technology is popular because it is an attracting technology in the field of computer science. Cloud computing is internet base computing that usually referred the shared configurable resources is provided with computers and other devices as services. Cloud computing delegate services with a customer's data, software and computation over a network. The customer of the cloud can get the services through the network. In other words, users are using or buying computing services from others. Cloud can provide Anything as a Service (AaaS). Many service model are provided by the cloud they are IaaS ,SaaS and PaaS. Infrastructure as a service (IaaS) offer computers physical or virtual machines and other resources. Infrastructure as a service (IaaS) clouds often offer additional resources such as a virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. Infrastructure as a service (IaaS) cloud providers supply these resources ondemand from their large pools installed in data centers. In the Platform as a service (PaaS) models, cloud providers deliver a computing platform, typically including operating system, programming language execution environment, database, and web server. Application developers can develop, run their software solutions on a cloud platform without the cost complexity of buying and managing the underlying hardware, software layers. With some Platform as a service (PaaS) offers like Microsoft Azure and Google App Engine, the underlying computer and storage resources scale automatically to match application demand so that the cloud user does not have to allocate resources manually. This paper concentrates on software as a service. It is a software licensing and delivery model in which software is licensed on a subscription basis and is centrally hosted. Sometimes referred to as "on-demand software". Software as a service (SaaS) is typically accessed by users using a thin client via a web browser. Software as a service (SaaS) has been incorporated into the strategy of all leading enterprise software companies. One of the biggest selling points for these companies is the potential to reduce Information Technology (IT) support costs by outsourcing hardware and software maintenance and support to the Software as a service (SaaS) provider. The vast majority of SaaS solutions are based on a multi-tenant architecture. To support scalability, the application is installed on multiple machines (called horizontal scaling). In some cases, a second version of the application is set up to offer a select group of customers with access to pre-release versions of the applications (e.g., a beta version) for testing purposes. And contrasted with traditional software, where multiple physical copies of the software each potentially of a different version, with a potentially different configuration, and often customized are installed across various customer sites. While an exception rather than the norm, some Software as a service (SaaS) solutions do not use multitenancy, or use other mechanisms such as virtualization to cost-effectively manage a large number of customers in place of multi-tenancy. Whether multi-tenancy is a necessary component for software-as-a-service is a topic

of controversy. Some limitations slow down the acceptance of Software as a service (SaaS) and prohibit from being used in some cases:

- Since data are being stored on the vendor's servers, data security becomes an issue.
- Software as a service (SaaS) applications are hosted in the cloud, far away from the application users. And introduces latency into the environment; so, for example, the Software as a service (SaaS) model is not suitable for applications that demand response times in the milliseconds.
- Multi-tenant architectures, which drive cost efficiency for SaaS solution providers, limit customization of applications for large clients, inhibiting such applications from being used in scenarios (applicable Gohila Priya dharshini.C et al, mostly to large enterprises) for which such customization is necessary.
- Some business applications require access to or integration with customer's current data. When such data are large in volume or sensitive (e.g., end users' personal information), integrating them with remotely hosted software can be costly or risky, or can conflict with data governance regulations.
- Constitutional search warrant laws do not protect all forms of Software as a service (SaaS) dynamically stored data.
- The end result is that a link is added to the chain of security where access to the data, and, by extension, misuse of these data, are limited only by the assumed honesty of 3rd parties or government agencies able to access the data on their own recognizance.
- Switching Software as a service (SaaS) vendors may involve the slow and difficult task of transferring very large data files over the Internet.
- Organizations that adopt SaaS may force into adopting new versions, which might result in unforeseen training costs or an increase in probability that a user might make an error. Relying on an Internet connection means that data are transferred to and from a SaaS firm at internet speeds, rather than the potentially higher speeds of a firm's internal network.

II. RELATED WORKS

Trust revolves around assurance and confidence that people, data, entities, information or processes will function or behave in expected ways. It may be 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. With respect to cloud, trust
can be given as the assurance of the hypervisors ability to isolate and establish trust for guest or hosted virtual machines that are critical, because this forms the root node for multitenant machine computing and trusted inter-operability Trust and trust models has been studied to great extent in earlier works, especially the characteristics of trust has been categorized into five groups, Competence; compete, expert, dynamic, Predictability; predictable, Benevolence; good, goodwill, benevolent, responsive, Integrity; honest, credible, reliable, dependable, Others; open, careful or safe, shared understanding, certainty The relational behavior of trust was classified into hierarchical trust, social group, and social networks. Hierarchical trust considers all relationships in a hierarchical manner and represented by a tree organization where nodes represent individuals and edges represent the trust degrees between the pair of nodes with each defining a trust degree between them through transitivity Zhang et al., have classified the trust functions based on the following four dimensions Subjective trust vs. Objective trust, Transaction-based vs. Opinion-based, Complete information vs. Localized information, Rank-based vs. Threshold based. Capability of an entity's trustworthiness being measured objectively against a universal standard results in objective trust. If the trust being measured depends on an individual's tastes and interest the resulting trust is called subjective trust. Decisions made based on the individual transactions and their results is known as transaction based trust whereas the trust built based on just opinion of the individuals is opinion based trust. If the trust building operation requires information from each and every node, it is called complete information it is known as either global trust function or complete trust function. If the information collected only from one's neighbors' it is called localized information trust function. If the trust worthiness of an entity is ranked from the best to worst, it is rank based trust whereas the trust declared yes or no depending present trust threshold is known as threshold based trust.

Challenges in Trusted Computing

Trusted computing targets computing and communication systems as well as services that are predictable, traceable, controllable, assessable, sustainable, dependable, privacy protectable,etc.The emerging ubiquitous communication/network infrastructures, in conjunction with the Internet, enable heterogeneous computers/services, and even their components to be universally connected towards global computing. Trust and/or distrust relationships in such global computing exist ubiquitously in the course of dynamic interaction and cooperation of user-to-system, system-to-system, component-to-component, and user-to-user who are using the systems. It is another grand challenge to make truly trustworthy computing and communication systems that are

IPHV8I1040X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

massively distributed, loosely coupled, greatly heterogeneous, highly dynamic, etc. Trusted computing model need to identify the implication of trust, distrust and mistrust, also needs to measure the risk associated with it. With respect to security and privacy trust needs to be established for access control, identity management, privacy intrusion, automatic detection and standard protocols for security. For a trusted reliable and dependable system a fault tolerant, robust and survivable system with failure recovery and quality of service needs to be addressed. For a trustworthy services and applications, e-commerce and e-business requires digital rights management, trusted media distribution and web services are the primary challenges. In a socio-economic strand, trust needs to address the standards and interoperation technology with the impact of policy and legal issues of cyber trust. Also non-technical issues like ethics, sociology, culture, psychology and economy are other deciding factors to challenge a trustworthy system.

III. TRUSTED COMPUTING ENVIRONMENTS

Trust is a characteristic that often grows over time, in accordance with evidence and experience. To trust any program, we base our trust on rigorous analysis and testing, looking for certain key characteristics

- ***Functional correctness:*** The program does what it is supposed to, and it works correctly.
- ***Enforcement of integrity:*** Even if presented erroneous commands or commands from unauthorized users, the program maintains the correctness of the data with which it has contact.
- ***Limited privilege:*** The program is allowed to access secure data, but the access is minimized and neither the access rights nor the data are passed along to othe entrusted programs or back to an entrusted caller.
- ***Appropriate confidence level:*** The program has been *examined* and rated at a degree of trust appropriate for the kind of data and environment in which it is to be used. Trust is applicable to various domains in the information world from areas where computation occurs to areas where data's get stored. The below Fig. 1 from the trusted computing group describes various areas where trust forms a major alliance in ensuring security.

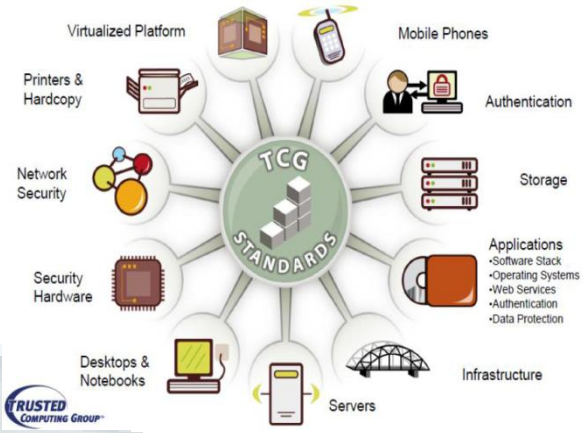

Fig.1Trusted computing groups classification of trust domains.

IV. PROPOSED ATTESTATION MODEL

Attestation of cloud environment can be done through either for web service applications or cloud providers servers and its infrastructures. Since more of the attacks on cloud are very critical and are focused on service provider's infrastructures, it is more necessary to provide attestation of the virtualization environment then the users applications. Hence attestation mechanism requires the following...

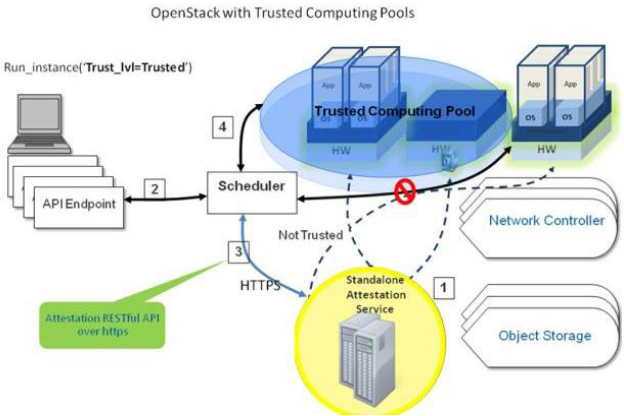

Fig. 2. Attestation server for trusted cloud platform

1) ***Attestation of VMs:*** only expected programs with expected configuration files are loaded inside the VM.

2) ***Attestation of Node Controllers:*** only the expected VM with the expected software stack has been instantiated. The VM the user is currently connecting to, is genuinely loaded by the genuine hypervisor.

3) ***Attestation of Storage Controllers:*** the VM is binding to the expected virtual storage, and the state of the virtual storage can only be manipulated by an expected software stack. In order to provide a trusted cloud, our proposed model creates attestation server that can either be placed

IPHV8I1040X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

remotely or within the local data centre. The functionality of the attestation server is to integrate the environment and collect the relevant measurements through an iterative process. The implementation can be done using Ubuntu Enterprise Cloud with Cloud Controllers in a server and Node Controller, Cluster Controller and Storage Controller on another server. Eucalyptus creates the cloud platform for instantiating the Virtual Machines with the help of KVM hypervisor. Thus Eucalyptus provides remote attestation services to cloud users. Trusted Computing enables Eucalyptus users with the capability of verifying the integrity of the Virtual Machines (VMs) and the Elastic Block Storage (EBS) volumes they own on the cloud. The integrity of a VM relies on that of the Node Controller (NC) and, in case, on that of the Storage Controller (SC) serving EBSs to the VM. Attestations of these three components should be made separately in order to provide proofs for the integrity of the entire VM's lifecycle. The cloud providers who deploy Trusted Computing Pools can provide premiere services to users who require services to be only run on compute nodes which are verified in running known and good hypervisors for ensured trustworthy environment. The Fig. 2 depicts how an attestation service acts as a service from outside the cloud with environments accessing the server are either trusted or not trusted based on the results from the attestation server. The trust level is calculated by the run instance API, at the endpoint level of the user and further attestation can be made available based on the measurements taken from the server.

V.PROPOSED SYSTEM

Software as a service and service oriented architecture are the basic concepts of SaaSclouds and this will allow the application service provider to deliver their application via cloud computing infrastructure. In our proposed method we are introducing a new concept called IntTest. The main goal of IntTest is, it can pinpoint all the malicious service providers. IntTest will treat all the service providers as black boxes and this does not need any special hardware or secure kernel support. When we are considering the large scale cloud system multiple service providers may simultaneously compromised by a single malicious attacker. In this we assume that the malicious nodes are not having any knowledge about the other nodes except those which they are directly interacting.

In this proposed system we are making some assumptions. First of all we are assuming that the total number malicious service components are less than that of the total number of benign service providers in the entire cloud. This assumptions is very important because without this assumption, it would be difficult for any attack detecting Scheme to work successfully. The second assumption is the data processing services are important deterministic. That is, the same input that is giving by a benign service component will always produce the same output. And finally we assume that the inconsistency caused by hardware or software faults can be excluded from malicious attacks. Fig. 3shows the overall a architecture of the proposed system. In this the user give request to cloud the survive will be deployed in the cloud the cloud will forward the user request to the SaaS and the response will be send to the cloud by the SaaS. And then the IntTest process will be done. After that the result auto correction will be done. After that the result will be send to the user by the cloud. The architecture shows this IntTest module in detail.

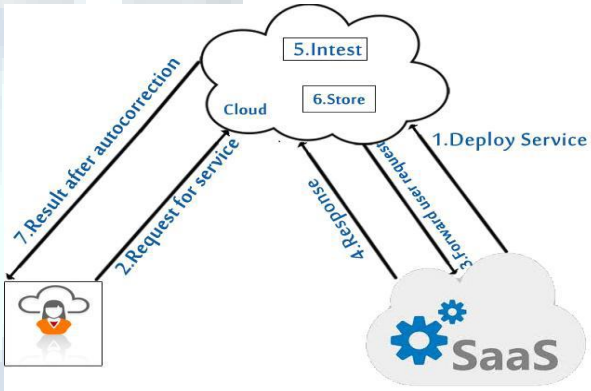

Figure 3: Over all architecture of the proposed method

VI. CONCLUSION

Integration of Cloud and Trust Computing can be a viable solution for communities with high data integrity requirements. Trust computing further unravels the benefits in making the cloud more secure through the means of attestation. The variety of attestation services makes the cloud more safe and secure for consumers.

This paper, discussed about various approaches and techniques used in providing the service integrity of SaaS cloud model. Each technique has its own advantages and disadvantages. Most integrity attacks can be effectively destroyed by the advanced techniques and approaches. All methods are approximate to our goal of providing the service or search results with integrity, we need to further perfect those approaches or develop some efficient methods.

REFERENCES

[1] Garay.J and Huelsbergen.L, "Software integrity protection using timed executable agents," in Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS), Taiwan, Mar. 2006.

[2] Juan Du Daniel J. Dean, Yongmin Tan, Xiaohui Gu, Senior and Ting Yu Scalable Distributed Service Integrity Attestation for Softwareas- a-Service Clouds .

[3] Du.J, Wei.W, Gu.X, and Yu.T, "Runtest: Assuring Integrity of Dataflow Processing in Cloud Computing Infrastructures," Proc.ACM Symp. Information, Computer and Comm. Security (ASIACCS),2010.

[4] Du.J, Shah.N, and Gu.X, "Adaptive Data-Driven Service Integrity Attestation for Multi-Tenant Cloud Systems," Proc. Int'l Workshop Quality of Service (IWQoS), 2011. Virtual Computing Lab, http://vcl.ncsu.edu/, 2013.

[5] Ho et al.T, "Byzantine Modification Detection in Multicast Networks Using Randomized Network Coding," Proc. IEEE Int'l Symp. Information Theory (ISIT), 2004.

[6] Hwang.I, "A Survey of Fault Detection, Isolation, and Reconfiguration Methods," IEEE Trans. Control System Technology, vol. 18,no. 3, pp. 636-653, May 2010.

[7] Lamport.L, Shostak.R, and Pease.M, "The Byzantine Generals Problem," ACM Trans. Programming Languages and Systems, vol. 4,no. 3, pp. 382-401, 1982

[8] Shi.E, Perrig.A, and Doorn.L.V, "Bind: A fine-grained attestation service for secure distributed systems," in Proceedings of the IEEE Symposium on Security and Privacy, 2005.

[9] Xu.W, Venkatakrishnan.V. N, Sekar.R, and Ramakrishnan .I. V, "A framework for building privacy-conscious composite web services," in IEEE International Conference on Web Services, Chicago, IL, Sep. 2006, pp. 655–662.

[10] Zhang.H, Savoie.M,Campbell.S, Figuerola.S, von Bochmann.G, and Arnaud.B.S, "Service-oriented virtual private networks for grid applications," in IEEE International Conference on Web Services, Salt Lake City, UT, Jul. 2007, pp. 944–951.

[11] S. Udhayakumar, S. Chandrasekhar, L. Tamilselvanan, and F. Ahmed, "An Adaptive Trust Model for software services in Hybrid cloud Environment," *Proc.15th WSEAS conference on Systems, Recent Researches in Computer Science*, 2011, pp. 493-502

[12] H. L. Zhang, B. Li, X. Wang, H. Q. Chen, and S. Z. Wu, "Application-Oriented Remote Verification Trust Model in Cloud Computing," *2nd IEEE International Conference on Cloud Computing Technology and Science*, 2010, pp.405-408.

[13] C. P. Pfleeger, *Security in Computing*, Fourth Edition, Prentice Hall, 2006.

[14] D. Harrison McKnight and Norman L. Chervany, "Conceptualizing Trust: A Typology and E-Commerce Customer Relationships Model," in *34th Hawaii International Conference on System Sciences*, Island of Maui, HI, USA, 2001.

[15] L. B. De Oliveira and C. A. Maziero, "A Trust Model for a Group of E-mail Servers," *CLEI Electronic Journal*, vol. 11, no. 2, pp. 1-11, 2008.

[16] Q. Zhang, T. Yu, and K. Irwin, "A Classification Scheme for Trust Functions in Reputation-Based Trust Management," in *International Workshop on Trust, Security, and Reputation on the Semantic Web*, Hiroshima, Japan, 2004.