Intruder Notification System & Security in Cloud Computing: A Review

Shweta Jaiswal, Manish Ahirwar, Raju Baraskar
University Institute of Technology-RGPV
India
shwetajaiswal1303@gmail.com
ahirwarmanish@gmail.com
rajubaraskar@rgtu.net

ABSTRACT: *Large-scale deployment & use of cloud computing in the industry is accompanied & in same time hampered by concerns regarding protection of data handled by cloud computing providers. One of the consequences of moving data processing & storage of company premises is that organizations have less control over their infrastructure. As a result, cloud service (CS) clients must trust that CS provider is able to protect their data & infrastructure from both external & internal attacks. Currently, however, such trust may only rely on organizational processes declared by CS provider & cannot be remotely verified & validated by an external party. Having tools to perform such verifications prior to launch as a VM instance, allows CS clients to decide at runtime whether certain data should be stored to calculations should be made at the VM instance offered by CS provider, This paper combines two components trusted computing, & cloud computing platforms to address issues of trust & security in public cloud computing environments.*

Keywords: IaaS, IMA-Integrity Measurement Architecture, Trusted platform modules (TPM), Cloud

Received: 25 July 2017, Revised 23 August 2017, Accepted 11 September 2017

1. Introduction

In spite of rapid expansion of Infrastructure-as-a-Service (IaaS) technologies such Microsoft Azure, Amazon EC2, services provided by attributes & others [1], IaaS services continue to be plagued by exposure at several levels as a software stack, from web based cloud authorization console to VM side-channel attacks, to information leakage, to collocated malicious virtual machine instances. Need to secure cloud storage & cloud computing environments has been reiterated on numerous occasions.

The continuous vulnerabilities discovered in the software stack underlying IaaS platforms has prompted move towards implementing trust anchors into hardware. Although this move has potential to greatly reduce risks posed by software vulnerabilities, it does not guarantee a secure platform out as a box. Rather, results rely on correct usage as trusted hardware. Trusted Computing initiative & adoption of trusted platform modules (TPM) has been steadily gaining momentum since its inception [7]. Participation of hardware manufacturing industry leaders in Trusted Computing Group 4 is likely to accelerate adoption of this technology across hardware architectures & platforms. Following its initial predominance & narrow focus on

laptop computers trusted computing is making its way into new devices. For example, the use of trusted computing, on mobile platforms is already focusing of several recent research projects [8, 9] with lots to come as increased functionality & ever lot information stored on mobile devices become lot attractive targets to malware. Another important application domain of trusted computing is its use of virtualized systems & cloud computing [10]. Trustworthy integrity verification of software components used in cloud computing infrastructure, as well as information protection using trusted computing techniques may address of security concerns related to premises computing. While it does not actually offer absolute guarantees, trusted computing raises complexity bar to attackers by placing the root of trust at hardware level 5. With a correct implementation, an attacker would need physical access to hardware in order to subvert TPM [11]. However, as the technology is still new & in active development, best practices to use of TPM are yet to be identified. This is especially relevant to virtualized environments & trusted cloud computing, where functionality of a single TPM chip needs to be shared between several virtual machines. Solutions like virtualization of TPMs [12] create new possibilities for implementation of secure launch & secure migration of VMs [13, 14]. In same time new attack techniques demonstrate that software implementation of TPM increases trusted computing base (TCB) & introduces new vulnerabilities [15]. This implies that new solutions to secure VM launch & migration need to be found based on existing components of TPM & with minimal changes to the TCB.

The figure shows issues related to diminishing control & transparency & technologies that may address such issues.

2. Trusted Computing

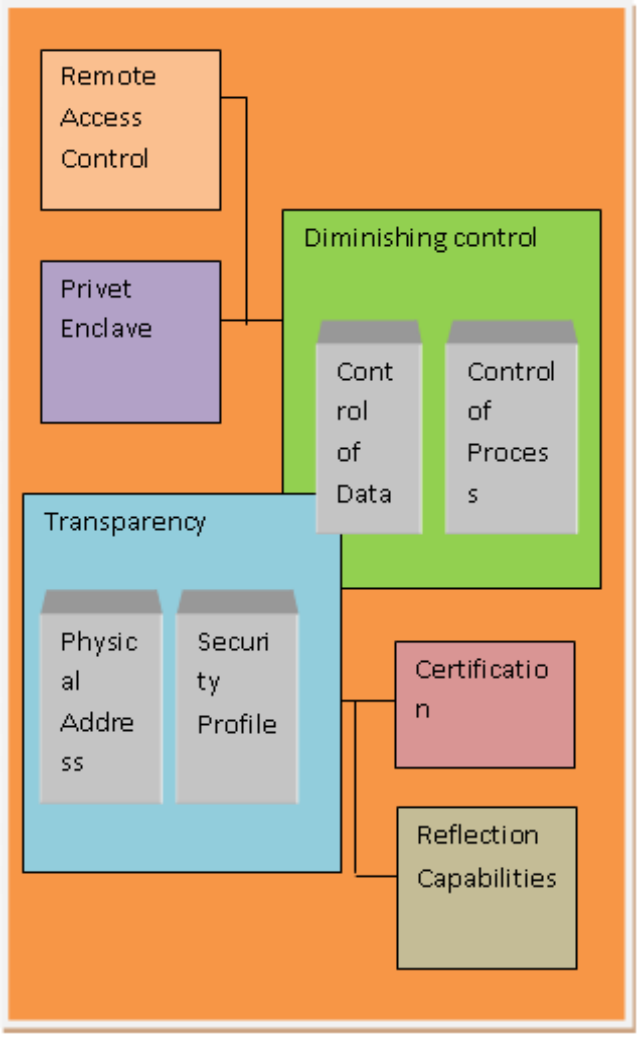


Figure 1. Trust in cloud computing

The TPM may be used to allow external parties to ensure that a certain host bearing TPM is booted into a trusted state. That is performed by verifying set of digests (called measurements) of loaded software, successively produced throughout the boot procedure of the device. Measurements are stored in a protected storage, built into TPM chip & are therefore resistant to software attacks, although vulnerable to hardware tampering.

C0: Input/Output, this performs protocol encoding & decoding, as well as directed information flow over communications bus.

C1: Non-volatile Storage is a persistent storage that is used to store non-migrateable keys {Endorsement Key (EK) & Storage Root Key (SRK) {as well as owner authorization & persistent configurations.

C2: Platform Configuration Registers (PCR) may be implemented in either volatile or non-volatile storage. TCG specification prescribes at least 16 PCRs, where PCR 0-7 are reserved to internal TPM use & registers 8-16 are available for OS & user space application use.

C3: Attestation Identity Keys (AIK): This component stores persistent keys that are used to sign & authenticate validity of information provided by TPM in case of external attestation. AIK may also be stored in encrypted form in an external data store, to accommodate multiple users on the same platform.

C4: Program code contains firmware that is used in order to measure platform devices & is a representation of the core root of trust measurement (CTRM).

C5: A Random number generator (RNG) is implemented in TPM in order to assist in key generation;

C6: A SHA-1 engine is implemented to hash generation to assist in signature creation.

C7: RSA key generation is a component to create asymmetric encryption keys based on Rivest, Shamir, Adelman protocol.

C8: RSA engine is used in order to perform signing, public-key encryption & decryption operations based on the RSA algorithm.

C9: Opt-in component allows to maintain activation state of TPM chip, possible states being enabled, disabled, deactivated.

C10: Execution Engine is a component that executes operations prescribed by logic in program code.

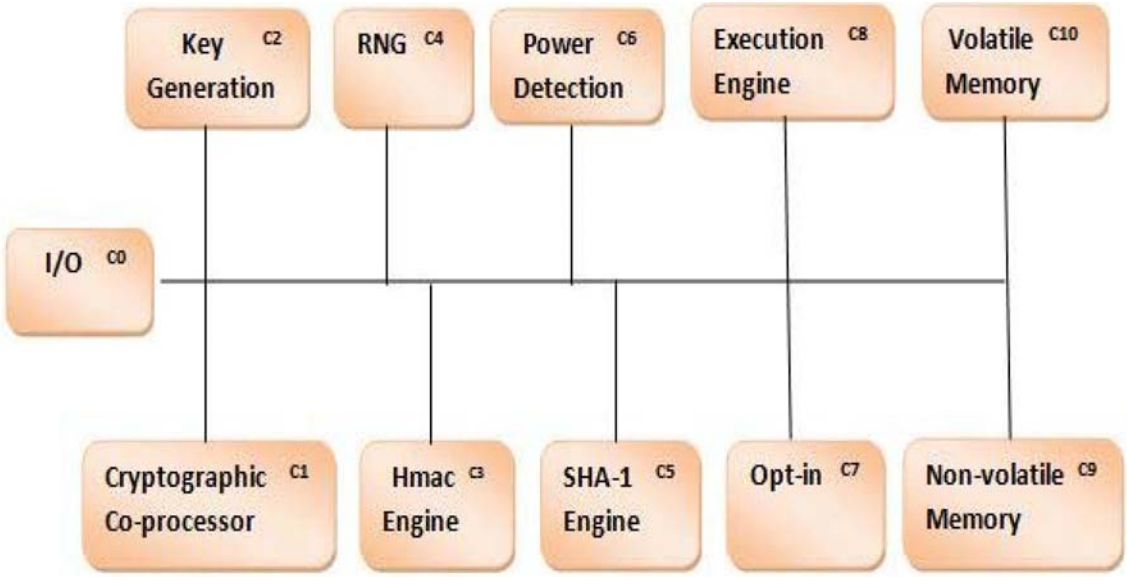


Figure 2. TPM Structure

Problem Statement: Problem of trust in public cloud environments is addressed by examining the state of the art inside cloud computing security & subsequently addressing issues of establishing trust in the launch of a generic virtual machine in a public cloud environment. As a result, the paper proposes a trusted launch protocol that allows CS clients to verify & ensure integrity of VM instances at launch time, as well as the integrity of the host where VM instance is launched. Protocol relies on the use of Trusted Platform Module (TPM) to key generation & data protection. TPM also plays an essential part in integrity attestation of the VM instance host. Along with a theoretical, platform agnostic protocol, the thesis also describes a detailed implementation design of protocols using Open Stack cloud computing platform. In order to verify, implement the ability of the proposed protocol, a prototype implementation has built using a distributed deployment of Open Stack. While protocol covers only trusted launch procedure using generic virtual machine images, it presents a step aimed to contribute towards the creation of a secure & trusted public cloud computing environment.

Binding	TPM over protection of message by means of asymmetric cryptography Using encryption keys generated & maintained by TPM. Thus, a message Encrypted using a particular TPM's public key is decryptable only by using a private key of same TPM.
Signing	Functionality is implemented according to the same principles of asymmetric encryption.
Sealing	A special case of binding functionality, where encrypted messages produced through binding are only decrypted-able in a certain platform state (defined through PCR values) to which message is sealed. This ensures that an encrypted message may only be decrypted by a platform which is in a certain prescribed state.
Sealed-sign	Offers possibility to verify that platform producing signature is in A certain specific configuration. Verification is based on measurements from a set of pre-determined PCRs that are evaluated to determine whether the platform has required configuration.

Table 1. Operations to secure message exchange provided by TPM

In this paper, we consider aspects of secure launch of generic VMs (VMs) in an entrusted public cloud computing environment. In this context, by generic VMs we mean VMs made available by the cloud service provider, however, assumed to be identical with vendor-issued models2. Scenario implies that the actor that launches a VM instance (further referred to as \client") necessary trusted launch of a VM instance available with IaaS provider. A specific requirement is that trustworthiness of virtualization environment where VM instance is launched should be verifiable through an automatic, scalable & least-intrusive way. An additional requirement is that the solution should be implementable using an open source cloud computing platform & should minimize potential for introducing new vulnerabilities through implementation of the solution. **Solution need:** Based on above defined security aspects of IaaS in public clouds & stated use case, we revisit requirements for a satisfactory solution to the above defined problem:

• The launch should provide to a user's mechanism to ensure that the VM has been launched with a trustworthy host. In order to establish whether a VM instance, launched in public cloud may be trusted, the client needs to have a verification mechanism to ensure that VM instance is running on a host which is considered secure", at least from software point of view. Verification should be provided by a party or component which is trusted by clients.

• The client should have possibility to reliably determine that it is communicating generic VM launched on a secure host, & not with a different generic VM instance. Given that a generic VM instance cannot, by definition, posses any properties known to the client that would make it identical to the client, it is important to provide reliable tools to CS client to distinguish a trusted VM instance from other types of generic VM instances.

• The integrity of VM must be verifiable by the target node Besides need to ensure integrity of the host where VM instance is running, it is equally important in the scenario of an interested cloud service provider to verify integrity of VM image. This paper

considers trusted launch of VMs using generic virtual machine images, i.e. VM images that have not undergone modifications of any kind, something which facilitates verification of VM images at the time of their launch.

3. Literature Work

Hamid Banirostam et al. [1] Presenting a User Trusted Entity (UTE) proposed approach is assumed to make cloud computing infrastructures genuine in order to entitle infrastructure service developed to provide a closed execution environment.

One advantage of initiating UTE is that managers of Infrastructure as a Service (IaaS) systems have no prerogative inside UTE. Therefore cloud computing managers cannot collude with Trusted Coordinator functionality. It has been supposed UTE should be kept by a third agent without any inducement to integrate with IaaS services & highly trusted to ensure confidential execution of guest virtual machines. In addition, UTE allows users to verify IaaS server & determine the security of cloud service before startup of the VM.

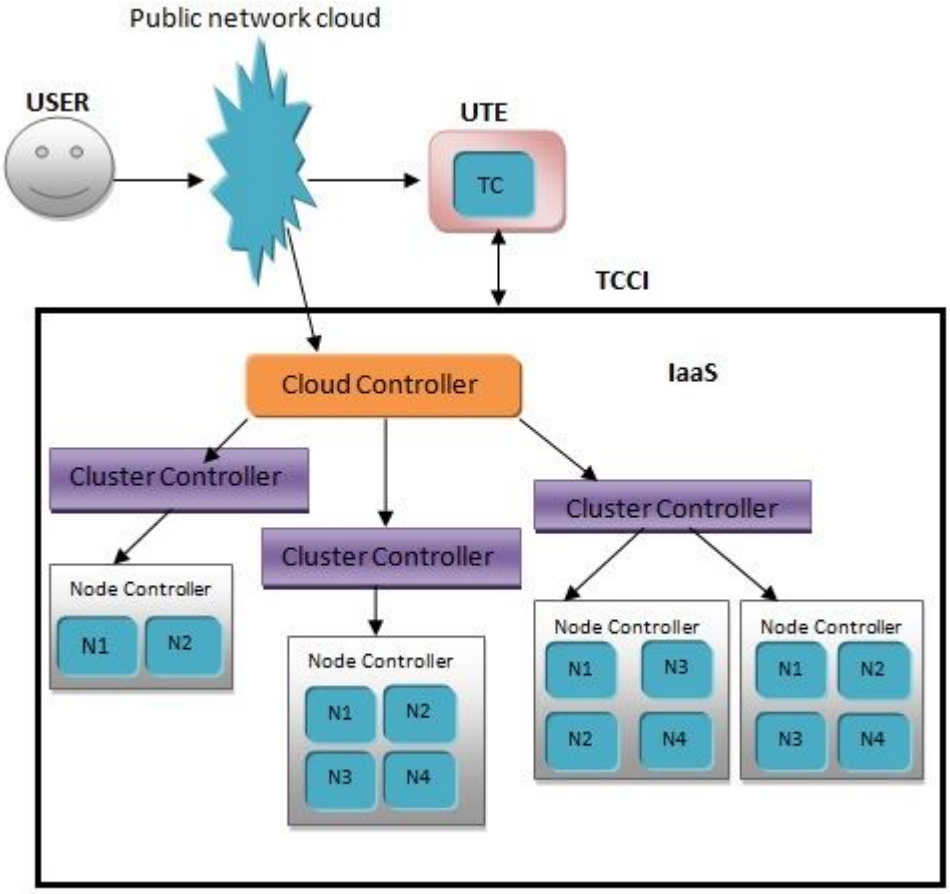


Figure 3. Trust model

Figure 3 shows the components of trust based cloud computing of noticed software contexts. These components are: a set of trusted nodes showed by symbol N. Trusted coordinator between nodes also has been shown by the symbol of TC, unreliable Cloud Manager (CM) demonstrated by CM symbol makes a set of services available by users. In the proposed approach, TC has been kept in, a User Trusted Entity (UTE). Presenting a scheme of a Trusted Cloud Computing Infrastructure (TCCI), this paper provides a closed execution space via developing trusted infrastructure context in IaaS backend. Their proposed approach performs operation without TC supervision. This causes in increasing system rate, however, instead decrease security. Challenges of confidentiality, data accuracy & integrity were studied in this paper & a scheme of a trusted\ infrastructure (TCCI) was proposed to resolve these problems. This scheme enables cloud computing infrastructures. The most important advantage of User Trusted Intity is that system administrators managing IaaS do not have any prerogatives so that none of them may

intercede in functionality of TC. It is assumed in this paper that UTE is managed by a third party with no incentive to conspire & also highly trusted by IaaS server. The ensure confidential running of guest VMs & also let user verifies IaaS server & before VM start-up check if the entrust cloud server is safe or not.

F. John Krautheim et al [2] they introduce a new mechanism of rooting trust in a cloud computing environment called Trusted Virtual Environment Module (TVEM). Trusted Virtual Environment Module is a software appliance that provides enhanced features of cloud virtual environments over existing TPM virtualization method, which includes an improved application cryptographic algorithm flexibility, program interface & a layout modular architecture. We define a unique Trusted Environment Key that integrates trust from information owner & service provider to create a dual root of trust of TVEM that is decided of every virtual environment & separate from the platform trust. This paper presents design, requirements & the architecture of our approach.

Dhananjay S. Phatak et al. [3] Private Virtual Infrastructure is a security architecture of cloud computing, which uses a new trust model to share responsibility of security in cloud computing between service provider & client, decreasing risk exposure to both Private Virtual Infrastructure architectures comprises a cluster of trusted computing fabric platforms that host virtual servers running an application server with a Locator Bot security service. Cloud Locator Bot pre-measures cloud platform of security properties to determine trustworthiness of the platform. Locator Bot uses Trusted Execution Technology & virtual Trusted Platform Modules to pre-measure target environment & securely provide Private Virtual Infrastructure with cloud, thus protecting information by preventing data from being placed in malicious or untrusted environments. Private Virtual Infrastructure — including Locator Bot — provides organizational tools to maintain control of their information in cloud & realize benefits of cloud computing, with assurance that their information is protected.

This paper presents a cloud trust model, Private Virtual Infrastructure architecture, & a Locator Bot protocol in enough detail to support further analysis or implementation Rizwana Shaikh et al [4] Cloud computing has become a part of combative market today. Various cloud computing service contributor are available with their services in a cloud infrastructure. To survey & calculate a particular service based on its security properties is a challenge. This paper presents in a computation by using a trust model.

4. Proposed Solution Method

From the above discussion, it is clear that TPM became an essential element of cloud computing, however TPM technology is still a new procedure & existing are working good & Intruders are also becoming familiar with available TPM & by time to time available TMP shows failures, hence it is highly necessary to develop a new TMP to Security in Cloud Computing. The proposed work is using VMware, eyeOS & OSSEC to intrusion detection.

VMware: it validates users to set up VMs on a single physical machine, & use them simultaneous along with genuine machine. Each virtual machine may execute its individual operating system, including versions of Microsoft Windows & Linux.

EyeOS : It is a private-cloud application platform with a web-based desktop interface. Commonly called a cloud desktop due to its unique user interface, eyeOS delivers a whole desktop from the cloud with file management, personal management information tools, and collaborative tools & with integration of client applications.

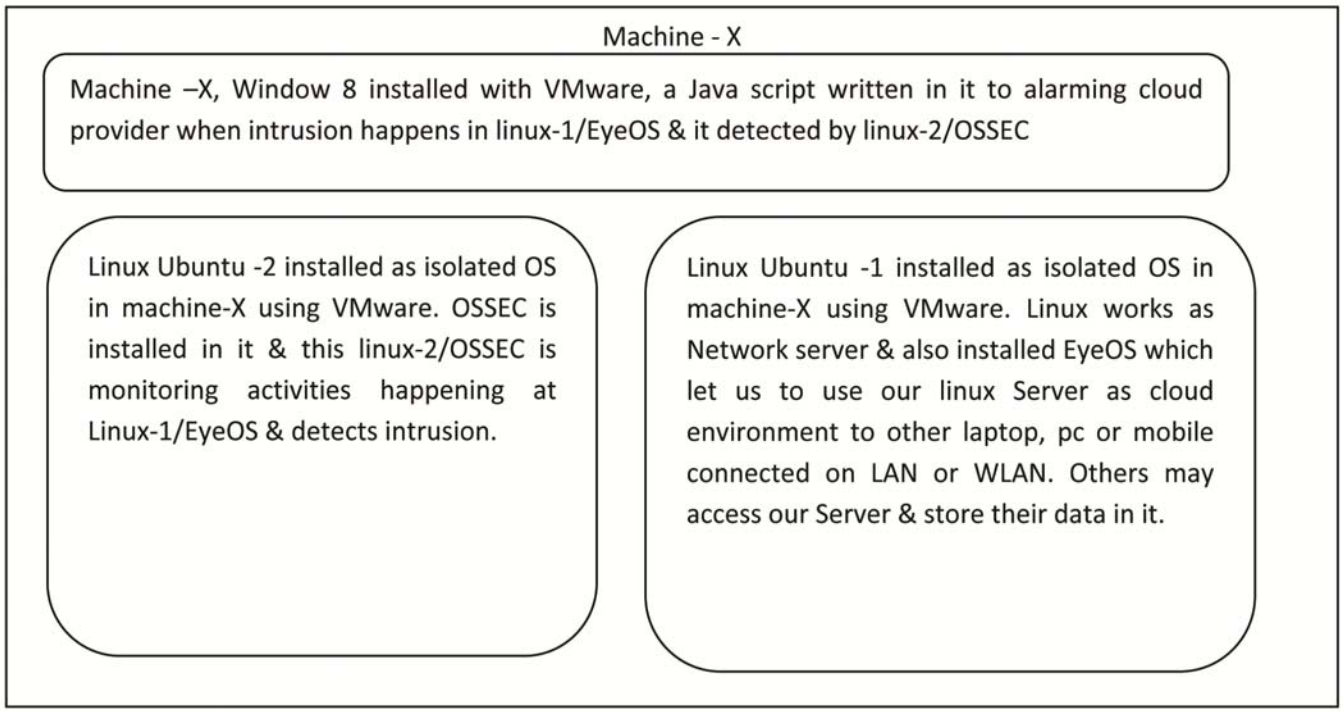
OSSEC: OSSEC is a free, open source host based intrusion detection system (HIDS) it performs integrity checking, log analysis, Windows registry monitoring, root kit recognition, time based notification & active response. It produces intrusion detection to most operating systems, including Linux, Solaris & Windows. OSSEC has a centralized, cross-platform architecture allowing multiple systems to be easily monitored & managed work flow shown in figure 3 is self explanatory that shows planning to propose research work.

Trust Model in Cloud Environment: Trust model may be used in a cloud environment. A trust calculation environment is prepared which includes various components. A framework is designed of calculation of trust in a cloud environment with multiple cloud service providers & their respective services. Architecture in figure3 shows various components of trust evaluation in a cloud environment. Major components are:

A trust calculation environment is prepared which involve various components. A framework is designed of calculation of trust

	Banirostam et al.[1]	Krautheim et al.[2]	pathak et al.[3]	Shaikh et al.[4]	K. Shelke et al.[5]	M. Khan et al. [6]	h Alias Kashif et al. [7]	Saravanakumar et al.
Security	Very Highly trusted	Highly trusted	High	Moderate	High	Unclear	Trust issue	Privacy issue
Authentication necessary	Yes	Yes	No	No	Yes	Yes	No	No
Type of attack	Non Linear & random	Nonlinear	Linear	Nonlinear	Malicious attacks	Linear	Linear	Linear
Application area	Security in infrastructure as a service (IaaS)	Security in infrastructure as a service (IaaS)	Security in infrastructure as a service (IaaS)	Infrastructure as a service (IaaS)	Infrastructure as a service (IaaS)	Infrastructure as a service (IaaS)	Infrastructure as a service (IaaS)	Infrastructure as a service (IaaS)
Complexity	Highly	Highly	Comparatively less	Moderate	High	High	High	Medium
Used technique	Trusted Cloud Computing Infrastructure (TCCI)	Trusted Virtual Environment Module (TVEM)	Private Virtual Infrastructure (PVI)	Identity Management System (IDA)	Intrusion Detection System (IDS)	Trusted Virtual Environment Module (TVEM)	Trusted Platform Module (TPM)	TVM
Speed	High speed	Moderate speed	Moderate	Less	High	High	Moderate	High
Algorithms	Trusted Cloud Computing Platform based presenting a User Trusted Entity (UTE)	Solve the major security in cloud computing to establish trust relationships & runs a virtual environment by separate service provider	Introduce Locator Bot (LoBot) to implement PVI on cloud resources with a level of assurance that is required to meet data confidentiality & privacy concerns of sensitive information	A framework is designed of calculation of trust in a cloud environment with multiple cloud Service providers & their respective services	Intrusion detection of grid & cloud computing	Trusted Platform Module (TPM)	Trusted Computing Group (TCG) by using utilizing the Trusted Platform Module (TPM)	Trusted Virtual Module (TVM)
Efficiency /Reliability	Medium	Comparable High	High	Medium	High	Medium	Competitive high	Medium
Space Complexity	More Space	Higher Space	Medium	Less	High	More Space	Low	Comparatively Less
Cost	High	High	High	Moderate	High	High	High	High
Accuracy	Highly	Moderate	Moderate	Moderate	Highly	Moderate	Highly	High
Data type	Custom software	Software etc.	Software & files, etc.	Software etc.	Custom software	Software etc.	Software etc.	-

Table 2. Comparisons among related work



Proposed work flow

in a cloud environment with multifarious cloud service providers & their respective services. Architecture in a figure 3 shows the various elements of trust evaluation in a cloud environment. Major elements are:

• **Cloud Service Manager:** Details about a specific cloud service like type of service, Service provider & number of users registered with that service & other accounting information is available with a cloud service manager. Lists of all available services are accumulated in it. Across with that, it also maintains a trust value associated with a cloud service that gives it security strength. A cloud service has to get registered with the cloud service manager of first time by the cloud service contributor before its use.

At this point of time static trust calculation is done with respect to static parameters. Over a period of time with service usage dynamic parameters are also considered & dynamic trust is evaluated. Any user who wants to select a particular cloud service will get detailed information about service & its strength from cloud service manager & according to select a cloud service.

• **Trust Model:** It is trust authority that makes use of service details to manipulate static or base trust values. It also uses service log & web of manipulative dynamic trust.

• **Service Logs:** It is database of log information about services. It consists of log records comprising of information such as; failed transactions, service utilization, number of successful & response time & much more.

These are made available to trust model to calculate trust value associated with a specified service.

• **Web Research:** It involves sources of user feedback & comments to draw accomplishment about dynamic security of cloud services.

The trust model compute values of various cloud services. Cloud users want to use one of cloud services depending upon their requirements. A cloud user may approach to a cloud service manage of involve services. A cloud service manager includes details about all available services along with their security strengths in terms of trust values. Based on user requirement & security strength a cloud service is selected. The trust model acts as raking service to determine the security strength of cloud

services. It evaluate both static & dynamic trust value in terms of security that may be used by users to determine security & reputation of cloud services.

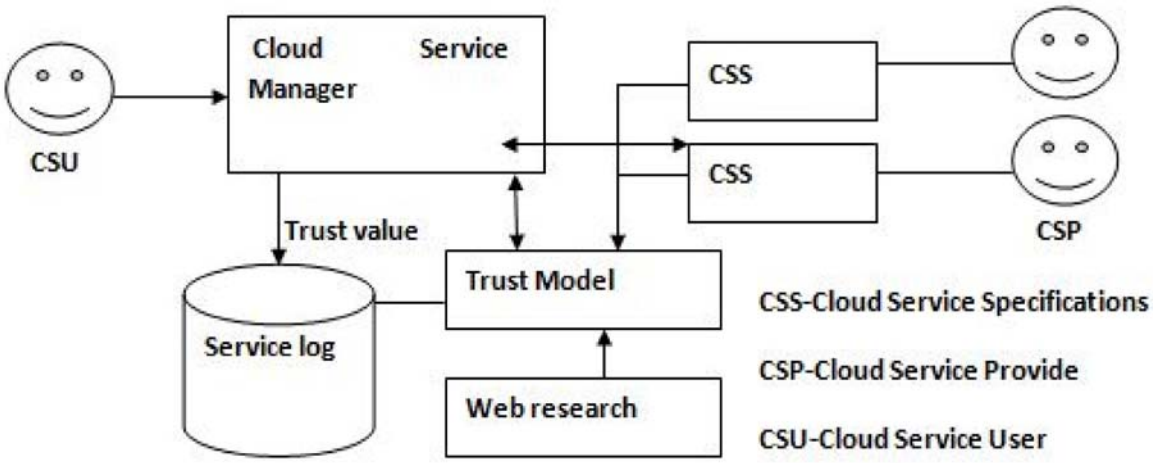


Figure 4. Trust modal by [4]

4. Conclusion

Broadly considered, aim of this paper has been to examine possibilities to increase security (in its broadest sense {confidentiality, integrity, availability) of virtualized environments in public cloud computing. Three domains {trusted computing, cloud computing & virtualization technology were included in background study phase. While each of the three domains is actively evolving as a result of large numbers of industry & academic contributors, trusted computing had an advantage of being thoroughly specified & documented in detail. Security concerns that hamper increased adoption of cloud computing abound, so this paper has focused on establishing trust in VM launch stage in a cloud computing environment. Till now all environments are being installed & network established & also Clod environment been created. In future work intrusion alarm system will be established.

References

[1] Banirostam, Hamid., Hedayati, Alireza. (2013). A Trust Based Approach to Increasing Security in Cloud Computing Infrastructure, *In:* 2013 UKSim 15th International Conference on Computer Modelling & Simulation, IEEE

[2] Krautheim, John., F., Dhananjay, Phatak, S., Alan, Sherman., T. (2010). Introducing Trusted Virtual Environment Module: A New Mechanism of Rooting Trust in Cloud Computing, A. Acquisti, S.W. Smith, & A. -R. Sadeghi (Eds.): TRUST 2010, LNCS 6101, p. 211–227, 2010. © Springer-Verlag Berlin Heidelberg.

[3] Krautheim, John., F., Dhananjay, Phatak, S., Alan, Sherman., T. Private Virtual Infrastructure: A Model of Trustworthy Utility Cloud Computing UMBC Computer Science Technical Report Number TR-CS-10-04, Krautheim & Sherman were supported in part by the Department of Defense under Information Assurance Scholarship Program grants H98230-08-1-0334 & H98230-09-1-0404

[4] Shaikh, Rizwana., Sasikumar, M. (2015). Trust Model of Measuring Security Strength of Cloud Computing Service, In: International Conference on Advanced Computing Technologies & Applications (ICACTA-2015), *ScienceDirect Procedia Computer Science* 45 (2015) 380 – 389.

[5] Parag, K., Shelke, Sneha Sontakke, A. D. Gawande (2012). Intrusion Detection System of Cloud Computing, *International Journal of Scientific & Technology Research*, 1 (4) May 2012.

[6] Khaled, M., Khan, Malluhi, Qutaibah. (2010). Qatar University, Establishing Trust in Cloud Computing, IT Pro September/ October 2010 . IEEE.

[7] Krautheim, F. J. (2009). PVI for Cloud Computing. **In**: Workshop on Hot Topics in Cloud Computing, San Diego, CA.

[8] Trusted Plateform Module Specified Version 1.2 Revision 103. Trusted Computing Group(TCG) (2007)

[9] Berger, S., Cáceres, R., Goldman, K. A., Perez, R., Sailer, R., van Doorn, L. (2006). virtual TPM: Virtualizing TPM. *In*: Proceedings of 15th USENIX Sec. Symposium, Vancouver, BC.

[10] Loeser, P., England, Para-Virtual TPM Sharing. *In*: Lipp, P., Sadeghi, A.-R., Koch, K. -M. (Eds.) Trust 2008. LNCS, (4968, pp. 119–132. Springer, Heidelberg (2008)

[11] Dragovic, Barham, P. B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A. (2003). Xen & Art of Virtualization. ACM SIGOPS OP Sys. Review 37, 164–177.