Attestation for Trusted Computing to Assure Security in Cloud Deployment Services

Udhayakumar Shanmugam, Latha Tamilselvan, Uma Nandhini, and Dhinakaran

***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. This paper proposes an attestation server that defines the functionality and measures the behavioral pattern of hypervisor, BIOS, boot devices and other operating system modules to verify it with good/known databases to determine nodes trustworthiness. Also provides an understanding of various attestation models and standards that justify that attestation as a service is a trustworthy mechanism to enable an ordinary platform to behave as a trusted computing platform.**

***Index Terms*—Attestation, trust, cloud computing, reputation, digital signature, virtual machine.**

I. Introduction

Cloud computing allows better IT resources optimization with virtually unlimited scalability and greater flexibility at a contained cost. As a result, cloud adoption is spreading rapidly and represents a new opportunity with existing advanced technologies like virtualization, service oriented architecture, utility, grid and autonomic computing. The cloud architecture encompasses this varied environment's to outsource the services to external third parties on a pay-per-use basis. This computing paradigm provides advantages both from economic perspective as well as scalability point of view because additional computing resources can be allocated when needed. The problem of data confidentiality and integrity, however prevent many organizations in adopting the cloud, as they are unable to determine where their data is stored, or do not understand how the cloud infrastructure is managed. Thus there are issues and challenges that need to be addressed for an effective implementation of cloud computing. The primary goal of the paper is to enhance the capabilities of security in cloud computing by implementing a suitable model that can enable the trust policies to mitigate the vulnerabilities associated in private and public cloud. This could well be achieved through trusted communication implemented through attestation server, where all the metrics related to virtual machines and it status are maintained and compared with earlier states of successful service deployment. Also, to provide a platform where the services can arbitrate and negotiate using trust metrics to provide maximum security, serviceability and availability for trusted components by setting up various Service Level Agreements (SLA) and privacy assurance strategy.

Manuscript received April 9, 2012; revised June 1, 2012.

Udhayakumar Shanmugam, Latha Tamilselvan, and Uma Nandhini are with the School of Computer and Information Sciences B.S.Abdur Rahman University, Vandalur, Chennai, India (e-mail: mailudhay@yahoo.com).

Dhinakaran is with the Sri Venkateswara College of Engineering, Sriperambadur, Chennai, Inia.

A. 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 [1].

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 [4]. The relational behaviour 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 be-tween the pair of nodes with each defining a trust degree between them through transitivity [5]. Zhang et al., have classified the trust functions based on the following four dimensions [6] 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 neighbours 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.

B. 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 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.

II. 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[3]:

1) *Functional correctness:* The program does what it is supposed to, and it works correctly.
2) *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.
3) *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 other untrusted programs or back to an untrusted caller.
4) *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 [7] describes various areas where trust forms a major alliance in ensuring security.

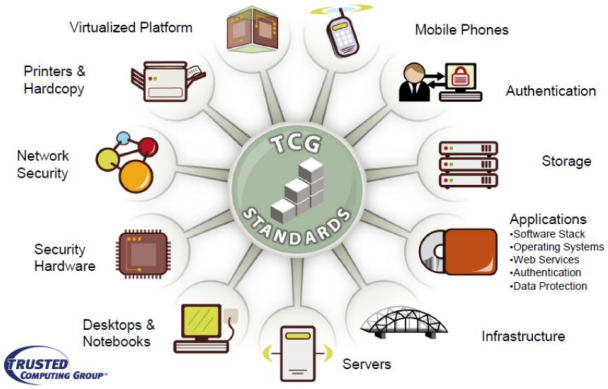

Fig. 1. Trusted computing groups classification of trust domains.

Trust can be classified as below in Table I. The domain for trust has reached in every level from a simple social networking environment to the core of operating system and virtual machines.

TABLE I: Classification of Trust in Various Environments.

Trust in Mobile Adhoc Network	**Trust in Social Networking and Peer-to-Peer Computing**
a. Fairness in Dynamic adhoc network b. Trust evaluation for secure routing c. Cooperation of nodes for identifying shortest path	a. Identity management for users group b. Reputation based social linking c. Peer-to-peer trust for evaluating group strategy
Trust in Operating System and Applications	**Trust in Virtual Machine**
a. User identification and authentication b. Mandatory access control and discretionary access control c. Object reuse protection and complete mediation d. Audit and audit log reduction	a. Intrusion detection b. Allocation of services to VM's c. Trust Migration between Virtual Machines d. Scheduling and resource allocation in VM

A. Risk Associated with Trust

All Today's desktop and laptop computers are essentially open platforms, giving the user-owner total choice about what software runs on them, and the power to read, modify or delete files stored on them. This freedom has led to problems, such as

1) Insecurity for the user, since open platforms are prone to *infection* by viruses, worms, and to inadvertent installation of spyware, denial-of-service attackers, compromised software, keyboard key catchers, etc.
2) *Insecurity* for the network on which the computer is placed, since it may have viruses and worms,

denial-of-service attackers, etc., which threaten other machines on the network.

3) Insecurity for software authors and media content providers, *since* open platforms allow programs, music files, images etc. to be copied without limit and without loss of quality.

B. Vulnerabilities in Cloud

In recent years cloud computing has become a growing interest for organizations looking to reduce their IT costs by offloading infrastructure and software costs onto 3rd party organizations who offer software-as-a-service (SaaS) (e.g. Google Apps), platform-as-a-service (PaaS) (e.g. Google App Engine), and infrastructure-as-a-service (IaaS) (e.g. Amazon EC2). However, due to the relative infancy of cloud based computing services, there exists uncertainty about the level of information security offered by these services. IaaS cloud services are largely reliant on virtualization technology, which is seen as providing all the security and process isolation a customer might want. While virtualization offers some potential security, there are drawbacks and complexities of which cloud providers and customers should be aware. In 2011, out of a set of 1201 publically reported vulnerabilities 855 had cloud based security implications [9].

Side Channel Attacks: The co-location of multiple VMs on a single piece of hardware presents malicious VM owners with the opportunity to glean potentially sensitive information from victim VMs sharing the same hardware resource.

Scheduler Vulnerability Attacks: In hardware virtualization a hypervisor provides multiple Virtual Machines (VMs) on a single physical system, each executing a separate operating system instance. The hypervisor schedules execution of these VMs much as the scheduler in an operating system does, balancing factors such as fairness and I/O performance. As in an operating system, the scheduler may be vulnerable to malicious behavior on the part of users seeking to deny service to others or maximize their own resource usage. Recently, publically available cloud computing services such as Amazon EC2 have used virtualization to provide customers with virtual machines running on the provider's hardware, typically charging by wall clock time rather than resources consumed. Under this business model, manipulation of the scheduler may allow theft of service at the expense of other customers, rather than merely reallocating resources within the same administrative domain. The flaw is in the Xen scheduler allowing virtual machines to consume almost all CPU time, in preference to other users [8].

Cross-site scripting (XSS): It is a web-application vulnerability that allow attackers to bypass client-side security mechanisms normally imposed on web content by modern web browsers. By finding ways of injecting malicious scripts into web pages, an attacker can gain elevated access-privileges to sensitive page content, session cookies, and a variety of other information maintained by the browser on behalf of the user. Cross-site scripting attacks are therefore a special case of code injection.

Similarly, DOS attack, Cross Site Request Forgery (CSRF), Remote File Include (RFI) SQL injection, XML injection, arbitrary file access and memory corruption are some of the other attacks and vulnerabilities that are more prevalent. These attacks have clearly indicated that Trust, Privacy and Security (TPS) are the major threshold areas in Cloud Security.

III. Trusted Cloud Models : Attestation

Attestation is the means by which a trusted computer assures a remote computer of its trustworthy status. The platform is manufactured with a public/private key pair built into the hardware. The public part of the hardware key is certified by an appropriate CA. Each individual platform has a unique hardware key. Using the private part of its hardware key, the system can guarantee assertions about the platform state. A remote computer can verify that those assertions have been guaranteed by a trusted computer.

An Application Key (AK) is a key pair created during attestation, for use by a particular application. At creation time, it gets tied to the platform's root key. Using an AK instead of using the platform root key directly has several benefits: (a) it reduces the load on the hardware, since only the hardware can use the root key but the CPU will use the AK; (b) helps prevent cryptanalysis of the root key; (c) somewhat addresses the privacy issues, since the AK is not directly associated with the hardware. The AK in this protocol is signed by the hardware key, which means any verifier (service provider) can link the session with the unique identity of the PC. This means that all the activities of the PC user can be linked, and a profile of the activities can be built.

A. Attestation Protocol

The hardware has a public/private key pair, PK_h and SK_h, called the hardware root key.

1) When an application A is started, it first generates a *public*/private key pair PK_A and SK_A, called the application key (AK). The application requests the hardware to certify its public key. The certificate CA = $\{PK_A, \#A\}SK_h$ returned by the hardware includes a hash of the executable A.
2) When the application wants to attest its validity to a remote server, it sends the certificate chain (PK_h, C_A) to the server. The server checks:
   - The signatures are valid, and PK_h is not revoked.
   - The application hash embedded in CA is on the server's list of applications it trusts.

The application now authenticates itself by proving knowledge of SK_A. For example, the application and the server can run a key exchange to generate a session key.

B. Limitations of Attestation

We emphasize that attestation must result in a shared secret between the application and remote party, otherwise the platform is vulnerable to session hijacking—an attacker could wait for attestation to complete, reboot the machine into untrusted mode, and masquerade as an authorized application.

Leveraging attestation requires the presence of software that allows the remote party to meaningfully interpret the

state of the system. This takes place through a multi-step process whereby the hardware will attest to what operating system it booted, the operating system will in turn attest what application it requires a key for, and will only allow the use of that key by that given application.

1) It is important to realize that software attestation only tells a remote party exactly what executable code was launched on a platform and establishes a session key for future interaction with that software component on the platform.
2) The software component could be buggy and produce incorrect results. The onus is on the remote party to choose who to trust.
3) Attestation provides no information about the current state of the running system.
4) For example, attestation does not show whether the software component has been compromised by a buffer overflow attack, infected by a virus, etc.
5) Future behaviour can only be ensured for authenticated interactions via a shared secret.
6) A platform is only as trusted as the tamper resistance of hardware and level of assurance of its trusted OS.
7) Attestation tells a remote party exactly what executable code was launched on a platform and establishes a session key for future interaction with that software component on the platform.

C. Classifications of Attestation

Attestation as Services (AaaS) would be the next service being provided in the cloud architecture similar to Software as a Service and Infrastructure as a Service etc,.

Remote Attestation: Remote attestation allows changes to the user's computer to be detected by authorized parties. For example, software companies can identify unauthorized changes to software, including users tampering with their software to circumvent technological protection measures. It works by having the hardware generate a certificate stating what software is currently running. The computer can then present this certificate to a remote party to show that unaltered software is currently executing. Remote attestation is usually combined with public-key encryption so that the information sent can only be read by the programs that presented and requested the attestation, and not by an eavesdropper.

Property Based Remote Attestation: Property-based remote attestation method oriented to cloud computing is designed based on the characteristics of cloud computing. In this method, through the attestation proxy, the remote attestation of the computing platform's security property is realized without disclosing the platform's configuration, and users can validate the security property of the actual computing platform in the virtual cloud computing environment.

Hash Based Attestation: The key primitive provided by secure coprocessors is *hash-based attestation*, whereby the platform generates a certificate that captures the binary launch-time hash of all components comprising the software stack. Hash-based attestation forces all trust decisions to be *axiomatic*.

Logical Attestation: Logical attestation is based on attributable, unforgeable statements about program properties, expressed in logic property descriptions represented as logical formulas. It builds on much past work that uses logical inference for authorization, known as credentials-based authorization.

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...

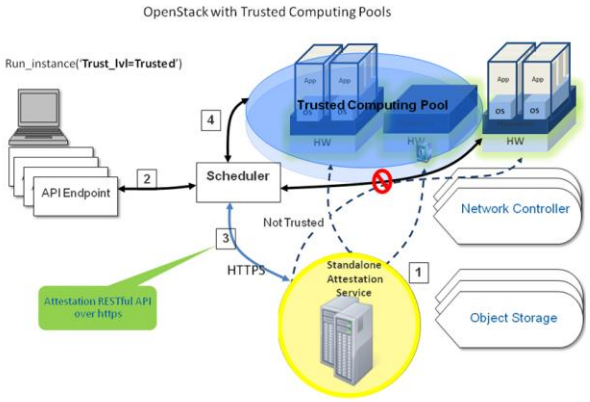

Fig. 2. Attestation server for trusted cloud platform (Courtesy open stack project of apache 2.0)

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 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. 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.

References

[1] S. Udhayakumar, S. Chandrasekar, 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

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] trustedcomputinggroup [Online]. Available: http://www. org

[8] F. F. Zhou, M. Goel, P. Desnoyers, and R. Sundaram, "Scheduler Vulnerabilities and Attacks in Cloud Computing," *College of Computing and Information Science*, Boston, 2011.

[9] [Online]. Available: http://info.Cenzic.com/2012-Applicaiton-Security-Trends-Report.html