2016 IEEE Symposium on Service-Oriented System Engineering

Trust Modeling in Cloud Computing

Mary-Jane Sule, Maozhen Li
Dept. of Electronic and Computer Engineering
Brunel University London
Uxbridge, UK

Gareth Taylor
Brunel Institute of Power Systems
Brunel University London
Uxbridge, UK

***Abstract*— *Though, cloud computing continues to gain popularity and has enormous economic gains; the perceived lack of trust from end-users accessing these resources continues to hinder its full deployment, usage and adoption. Tradition security and privacy controls continue to be implemented on cloud but due to its fluid and dynamic nature, research work in the area of end-user attestable trust evaluation of the cloud platform seem to be limited. With the nature of cloud, service level agreements are not enough, the user would want a transparent system with a traceability facility that allows the user to determine the relationship between varying trust relationships across the cloud layers, components, algorithms and applications especially at large scale. This paper presents some security mechanisms that enable cloud service end users to evaluate the trust level of various cloud services and resources. These mechanisms are evaluated based on fuzzy theory on a Eucalyptus cloud platform.*

***Keywords* —** *Security, trust, trusted cloud computing.*

I. INTRODUCTION

Cloud computing continues to offer scalable computing resources to small, medium and large enterprises through virtual, shared, metered, scalable and on demand resources across the Internet or Local Area Network (LAN). [1], [2]. It is projected that by the year 2020, cloud computing could generate up to 2 million jobs in Europe alone [3], [4].

While a lot of research has gone into making cloud computing more secure by enforcing security mechanisms like encryption, firewalls, and security groups, obstacles to trust still hinder potential users from completely adopting "cloud" (as it is sometimes called). End-users are concerned about the "faceless" provider taking control and having access to their resources during processing or storage in the cloud. The end-user wants to be more aware about what happens at the backend of the cloud platform[1] [5]

An end-user's trust concern doesn't lie entirely on technology but also lack of transparency. While an end-user desires a secure platform, the end-user also wants measures that would promote transparency and accountability and this is lacking even among major providers like Amazon EC2 or S3, Microsoft Azure and Google. Insufficient information could lead an end-user to distrust even the most secure system.[6][7].

In general, it may be stated that the overall trustworthiness of any cloud resource may be derived from an algorithmic sum of trust levels (security measures) that are enforced or available across all layers or parts of a computing system[8].

In the case of cloud computing, the ability to probe and test on-demand the trustworthiness of all assigned or consumed computing resources would make end-users confident of the platform as it provides them with additional assurance of data-integrity both during storage and processing.

Fig. 1 shows three layers that popularly make up the cloud platform, these are also known as the delivery models – *Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS)*.

IaaS provides the cloud user with shared storage and computing resources, which include CPU, disk space and memory. *PaaS* provides the cloud user with a platform for testing any application. While, *SaaS* provides the user with software and applications without the user having to worry about licenses and installation, this usually runs on the cloud provider's infrastructure [1].

In cloud service delivery model, IaaS is at the lowest level and is usually the foundation of all cloud computing. It may be stated that all cloud services fundamentally require an underlying cloud infrastructure that is *Infrastructure as a service (IaaS)* layer even when it is not directly offered as a service. The cloud layers are arranged and used in a way that does not prevent a security flaw in one layer affecting the other layers as seen in Fig. 1.

Cloud applications also vary considerably in complexity, while some are simple static web-sites, others may be dynamic, database driven and contain complex sequences of tasks (workflow). Depending on the need of the users, cloud services may be deployed in a number of ways - *Private, Public, Community or Hybrid* cloud deployments with varying security considerations even for major issues like confidentiality, integrity (of data) and availability. Understanding the inter-play of Trust based on individual security-measures implemented in cloud layers and other components is important for the overall trustworthiness of a cloud system.

From Fig. 1., we also see that an end-user who subscribes to a SaaS provider or any of the services may not be concerned about the trustworthiness of the PaaS layer or other layers but maybe concerned with the trustworthiness of the SaaS layer as that is the resource the end-user wants; or the end user maybe subscribing to all the services and maybe concerned with the overall trustworthiness of the platform.

[1] NIST defines Cloud Infrastructure (CI) as the combination of the hardware and software that enable cloud computing

978-1-5090-2253-3/16 $31.00 © 2016 IEEE
DOI 10.1109/SOSE.2016.32

IEEE computer society

Authorized licensed use limited to: Cranfield University. Downloaded on June 19,2025 at 13:45:10 UTC from IEEE Xplore. Restrictions apply.

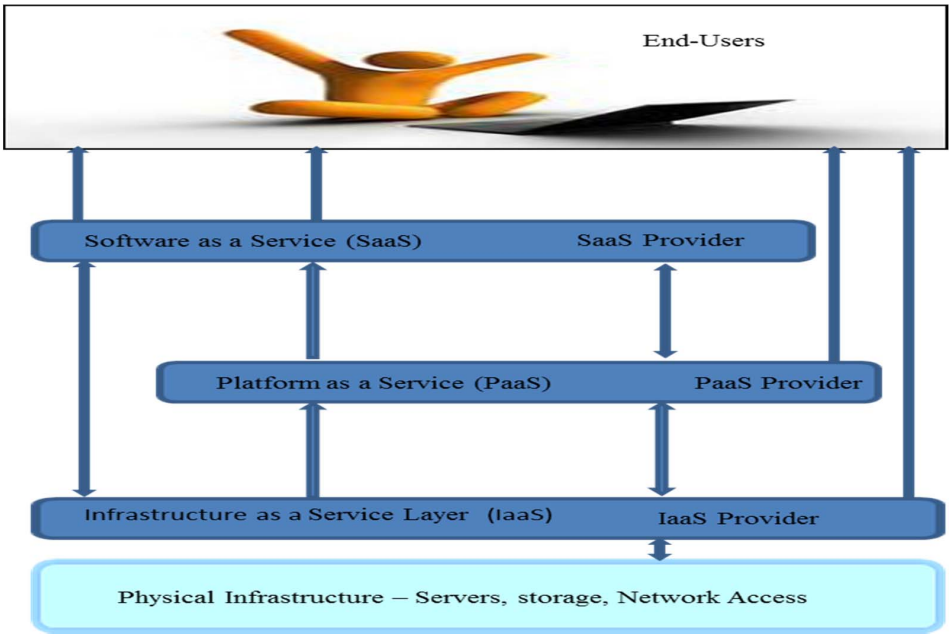

Fig. 1. Cloud Service Delivery Models and Trust Interaction

Whatever the needs of the end-user, the trustworthiness of any cloud service is of paramount concern. In cloud computing, the selection, scheduling, allocation and consumption of resources is typically governed by algorithms that may be influenced by both internal and external factors such as user requirements, current consumption levels, availability requirements and other legal requirements according to the *Service Level Agreement - SLA* documents which may also affect trust directly or in-directly [9], [10].

This paper presents security mechanisms which supports and provides a means to evaluate trust in cloud, these mechanisms may be used to comprehensively model trust across various cloud layers and components, from its physical infrastructure to data storage and processes.

II. Related Work

Various researchers have carried out work on trust computing and some works have been extended to even include trust in cloud computing.

Researcher in [11] presented a model based on fuzzy reputation for trust management in cloud but its focus was on data storage service providers.

While in [12], the authors proposed a cloud computing trust framework based on third party trust providers which means the user is not the verifier and the information is based on second hand communication.

From [6] we know that end-users' trust concerns doesn't entirely lie on technology but also lack of transparency.

Researchers in [13] presented a trust model in cloud computing environment were fuzzy theory based method was used to calculate the trust value of a IaaS cloud provider based on user impressions. The trust model is based on the VMs and had two considerations response time and linear trust chain. This method was evaluated using CloudSim and mathlab.

The research work presented in this paper involves the ability of a user to evaluate the trustworthiness of the various resources on a cloud platform using fuzzy logic. This ability is based on the security mechanisms that have been configured across all layers of the platform.

III. Trusted Computing and Fuzzy Theory

According to [6], [8][14], trust in computing is modeled after human relationships and as such is strongly associated with security; that is, generically the concept of trust (and security) may be applied to other areas, for example, a party may "trust" another party to deliver secure quality service, in which case trust becomes a measure of the "security" of service-availability. Authors in [15] classified trust into objective and subjective trust. Objective trust is concerned with the impressions directly between two parties; subjective trust involves the impressions obtained from third parties.

As shown in Figure 1, trust in cloud should be examined at 4 different layers (Physical, IaaS, PaaS and Saas).

In this work, trust at the physical layer is derived from the defense capability of individual physical devices. A single cloud host (physical server) which is an IaaS component may be secured using the TPM based Trusted Computing (TC) industry standards [16]. As shown in Fig. 2 [13], a chain of transitive trust is established from hardware (core Root of Trust) to the software layers (virtualization) across the BIOS device, the boot-loader and operating system.

Authorized licensed use limited to: Cranfield University. Downloaded on June 19,2025 at 13:45:10 UTC from IEEE Xplore. Restrictions apply.

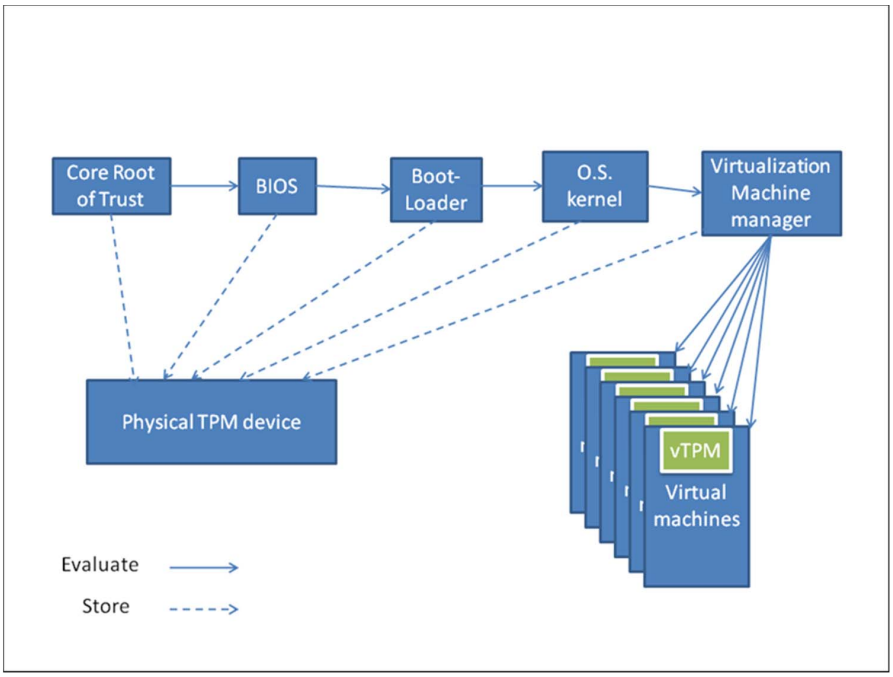

Fig. 2: Linear and tree based on trusted computing of a single host

In Figure 2, the chain of trust that exists between the core root of trust to the virtualization machine manager layer may be modeled based on the linear transitive trust principle where the final trust is the minimum offered by each intermediary object that is part of the chain. Each stage of the TC chain (cRMT to VMM) shown in Figure 2 may be represented as a node in an N node graph system where the trust level between two adjacent nodes would be given by the expression:

$$T(A,B) = \min\{T(A,B), T(B,C), T(C,D), \ldots T(N-1,N)\} \quad (1)$$

Similarly, the trust relationship between the virtualization machine manager and the virtual machines is a one-to-many relationship of direct trust and may be modeled by a tree graph where the trust relationship is the maximum trust level obtained from each directly connected object. That is, assuming a one-to-many node (VMM to vmi) system where T denotes the Trust level between two adjacent levels.

$$\mathrm{T}(VMM, vm_1) = \max \{\mathrm{T}(VMM, vm_1), (VMM, vm_2), \ldots, (VMM, vm_i)\} \quad (2)$$

While, the virtual machines may include a virtual TPM device as shown in Figure 2, their virtual nature precludes their consideration in the overall security and trust of the IaaS layer.

Typically, the IaaS layer is a collection of physical resources (such as hosts) that may be co-located in a single data-center or distributed across geographically separated data-centers and the net trust expected from the IaaS service layer is obtained from the multiplicative sum of the trust levels of individual physical hosts and resources. It is clear that the presence of a single compromised or unsecured host would negatively affect the trust of all hosts in a data-center or the IaaS layer.

$$T_{IaaS}\,\text{layer} = T_{R1} * T_{R2} * \ldots * T_{Rn} \quad (3)$$

In many cloud implementations, the cloud PaaS layer is mainly secured using either zones or firewalls and in some cases both are implemented at the Cloud middle-ware and the trust replications may thus be modeled by the equation with the number of virtual machines limited to the finite set of vm(s) sharing the same zone. In addition, trust at the PaaS layer may be complemented by a defence capability such as the inclusion of an Intrusion Detection System (IDS) engine directly into the PaaS software-environment. In which case, the IDS is used to provide a baseline control against which the contents (files and data) of the PaaS system may be compared to detect unauthorised modifications or updates to the PaaS software-environment caused by Trojan horses, viruses or other malicious activities. The IDS engine is typically a separate application that runs within a vm, consumes resources and provides a measure of the one-to-many (tree) direct trust of other applications within the vm. That is, the IDS trust may be modeled by:

$$T(IDS, app_1) = \max\{T(IDS, app_1), T(IDS, app_2), \ldots, T(IDS, app_n)\} \quad (4)$$

Where $T(IDS, app_n)$,represents the trust measure provided by the IDS engine for a particular application.

Authorized licensed use limited to: Cranfield University. Downloaded on June 19,2025 at 13:45:10 UTC from IEEE Xplore. Restrictions apply.

The PaaS environment is typically created from a software hierarchy consisting of a boot-loader, operating system kernel, shell and libraries. The final security of individual applications therefore depends on the security of this underlying software stack. That is:

$$T\ (IDS, app_x) = \min\{T(IDS, bootloader), T(IDS, O.S - Kernel), (IDS, O.S - shell), (IDS, O.S - libraries)\} \quad (5)$$

PaaS layers are typically provided by virtual machines from the underlying IaaS layers and the trust relationship between the IaaS and PaaS layers may be modeled as a trust fusion or quasi-summative or average process.

$$T\ (IaaS, PaaS_1) = \max\{\ T(IaaS, PaaS_1), T(IaaS, PaaS_2), .., T(IaaS, PaaS_n\} \quad (6)$$

And for each PaaS

$$T\ (PaaS_X) = T\ (IaaS, PaaS_x) + T\ (IDS_x, app_x) \quad (7)$$

Where the + sign represents the trust fusion process and $T\ (IDS_x, app_x)$represents the added component provided by the internal IDS engine.

In cloud implementations, the SaaS layer is typically protected by securing the communication layer against man-in-the-middle attacks using a suitable mechanism such as secure socket layer (SSL) connections or secure shell (SSH) connections. Both SSL or SSH represent identity Trust relationships based on possession of matched certificates/credentials, however, while SSL is more in-direct requiring a 3rd party certification authority, SSH connections represent direct-trust between two parties. SSH connection may be modeled as simple vector, where an unknown host is not trusted while a known/matching host is fully trusted. It is noted that a known but not-matching host is completely untrusted.

T (SSH) = {0 if identity-trusts exists but is not matched
{1 if identity-trust is matched (8)

Although trust in SSL is different from SSH, the simplest forms of SSL connections may similarly be modeled by:

T (SSL) = {0 if identity-trust is NOT verified
{0 if identity-trusts is revoked
{1 if identity-trust is verified (9)

However, the ability to revoke certificates suggest that SSL connections may be better modeled by more complex algorithms such as EigenTrust or PeerTrust where the group of peers is limited to a well-known set of peers or community.

At the application level, data at the SaaS layer may be protected against theft or loss using a suitable mechanism such as hashing, data-coloring or encryption or obfuscation. Data-coloring could be an optimal technique for cloud platforms and applications [17]. This is so, as colored data may still be processed without overhead (the coloring is transparent) while also providing the ability to detect if the data has been tampered with, identify and report data-loss. In essence, colored-data (output of a data-coloring process) may be viewed as being able to maintain integrity and availability of data without a processing overhead.

Therefore, considering security related metrics of confidentiality, integrity and availability in a cloud context, hashing as a data-protection technique can only provide integrity, while obfuscation provides only limited confidentiality; data-coloring provides integrity and availability; while encryption provides confidentiality and integrity but not availability.

$$T\ (DP) = \sum(0.15T_{hash} + 0.35T_{dc} + 0.15T_{ob} + 0.35T_{enc}) \quad (10)$$

In equation (10), the assigned weights are based on the relative contribution by the individual technique. Equation (10) may be extended to include other techniques and the absence of an individual technique does not adversely affect the derived Trust.

The SaaS layer trust may be modeled as a summation of the fusion of Trusts derived from the connections of security and data-protection.

$$T(SaaS) = \frac{\big(T(conn) + T(DP)\big)}{2} \quad (11)$$

Typically, a SaaS service requires an underlying PaaS. The trust relationship between a PaaS and a SaaS application is typically a 1-to-1 nature which may be modeled as a discount trust process, that is

$$T(SaaS) = T(PaaS, SaaS) * \left(\frac{\big(T(conn)+T(DP)\big)}{2}\right) \quad (12)$$

The simulation for this research would be implemented to accommodate different scenarios. These scenarios would be deployed to accommodate different levels of trust, this is achieved by deploying different security techniques across different layers and this in turn can be evaluated to reflect the different levels of trust in the platform.

The users depending on their needs and applications would be provided a VM based on the provisioning policy. The provision policy is further enhanced by fuzzy logic algorithms (type 1 and type 2). The trust measurements values are expected to be between 0 and 1.

Since no computing system can be completely trusted as every system is “only as secure as its weakest link”, the same can be said of the cloud system; therefore categorically stating a system is completely trusted or not may not be a true reflection of the system.

Fuzzy logic which is a computing approach that is based on “degrees of truth” includes cases of trust based on 0 and 1 (low

Authorized licensed use limited to: Cranfield University. Downloaded on June 19,2025 at 13:45:10 UTC from IEEE Xplore. Restrictions apply.

or high) but it allows the inclusion of various cases in between such as low to medium, medium to high, etc.

Fuzzy logic makes an aggregate of various parameters and based on certain thresholds makes a decision, it is suitable for modeling uncertainties. Type 1 fuzzy set or type 2 fuzzy set can be applied to the different cases for decision making[18], [19].

In this paper, using fuzzy logic evaluation, the configured security mechanisms are used to calculate cloud trustworthiness based on the user's needs and satisfaction. The user may only want to know the trust state of the IaaS and doesn't mind about PaaS and SaaS as the user doesn't need that service or maybe only concerned with the trust state of IaaS and PaaS and not SaaS or vice versa. Using fuzzy logic approach, trust values are combined to enable the user decide the trustworthiness of the platform or services.

IV. Conclusion

The mechanisms used and presented in this research to model cloud trustworthiness are unique. This paper presents how the techniques are used to evaluate cloud trustworthiness. Further work would include implementing and evaluating the techniques. The evaluation would include using fuzzy logic algorithms to select trustworthy cloud platform.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Draft) Recommendations of the National Institute of Standards and Technology," *Nist Spec. Publ.*, vol. 145, no. 6, p. 7, 2011.

[2] D. Sun, G. Chang, L. Sun, and X. Wang, "Surveying and analyzing security, privacy and trust issues in cloud computing environments," in *Procedia Engineering*, 2011, vol. 15, pp. 2852–2856.

[3] M. R. Nelson, "OECD Briefing Paper for the ICCP Technology Foresight Forum: Cloud Computing and Public Policy | Publications | Georgetown University," 2009. [Online]. Available: http://explore.georgetown.edu/publications/47143/. [Accessed: 28-Oct-2015].

[4] E. C. P. S. European Cloud Partnership SteeringBoard, "Establishing a Trusted Cloud Europe," 2012.

[5] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "TrustCloud: A Framework for Accountability and Trust in Cloud Computing," in *2011 IEEE World Congress on Services*, 2011, pp. 584–588.

[6] K. M. Khan and Q. Malluhi, "Establishing Trust in Cloud Computing," *IT Prof.*, vol. 12, no. 5, pp. 20–27, Sep. 2010.

[7] S. Pearson and A. Benameur, "Privacy, Security and Trust Issues Arising from Cloud Computing," in *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, 2010, pp. 693–702.

[8] A. Nagarajan and V. Varadharajan, "Dynamic trust enhanced security model for trusted platform based services," *Futur. Gener. Comput. Syst.*, vol. 27, no. 5, pp. 564–573, May 2011.

[9] M. Armbrust, I. Stoica, M. Zaharia, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, p. 50, Apr. 2010.

[10] Z. Chen and J. Yoon, "IT Auditing to Assure a Secure Cloud Computing," in *2010 6th World Congress on Services*, 2010, pp. 253–259.

[11] Wu Xu, "A Fuzzy Reputation-based Trust Management Scheme for Cloud Computing," *Int. J. Digit. Content Technol. its Appl.*, vol. 6, no. 17, pp. 437–445, Sep. 2012.

[12] W. Fan and H. Perros, "A novel trust management framework for multi-cloud environments based on trust service providers," *Knowledge-Based Syst.*, vol. 70, pp. 392–406, Nov. 2014.

[13] L. Gu, C. Wang, Y. Zhang, J. Zhong, and Z. Ni, "Trust Model in Cloud Computing Environment Based on Fuzzy Theory," *Int. J. Comput. Commun. Control*, vol. 9, no. 5, p. 570, Aug. 2014.

[14] E. D. Canedo and R. R. de C. and R. de O. A. Rafael Timóteo de Sousa Junior, "TRUST MEASUREMENTS YELD DISTRIBUTED DECISION SUPPORT IN CLOUD COMPUTING," *Int. J. Cyber-Security Digit. Forensics*, vol. 1, no. 2, pp. 140–151, 2012.

[15] M. B. B. K. Thomas Beth, "Valuation of Trust in Open Networks."

[16] M.-J. Sule, M. Li, G. A. Taylor, and S. Furber, "Deploying trusted cloud computing for data intensive power system applications," in *2015 50th International Universities Power Engineering Conference (UPEC)*, 2015, pp. 1–5.

Authorized licensed use limited to: Cranfield University. Downloaded on June 19,2025 at 13:45:10 UTC from IEEE Xplore. Restrictions apply.

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

[18] N. B. N. N. P. N. L. M. N. V. Prasath, “Fuzzy Logic In Cloud Computing,” vol. Vol.2 - Is, no. Vol.2 - Issue 3 (March - 2013), Mar. 2013.

[19] H. Liao, Q. Wang, and G. Li, “A Fuzzy Logic-Based Trust Model in Grid,” in *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2009, vol. 1, pp. 608–614.

Authorized licensed use limited to: Cranfield University. Downloaded on June 19,2025 at 13:45:10 UTC from IEEE Xplore. Restrictions apply.