International Journal of Computational and Experimental Science and ENgineering (IJCESEN)
Vol. 6-No.1 (2020) pp. 13-18
http://dergipark.org.tr/en/pub/ijcesen

ISSN: 2149-9144

Research Article

Development of Design for Enhancing Trust in Cloud's SPI Stack

Ahmed BENTAJER[1]*, Faouzia ENNAAMA[2], Mustapha HEDABOU[3], Said EL FEZAZI[2]

[1]Abdelmalek Essaadi University, ENSA of Tetouan, SIGL Lab., 93000, Tetouan-Morocco
[2]Cadi Ayyad University, EST Of Safi, LAPSSII Lab., 46000, Safi-Morocco
[3]Cadi Ayyad University, ENSA Of Safi, MTI Lab., 46000, Safi-Morocco
[3]Mohammed VI Polytechnic University, Ben Guerir, Morocco

* **Corresponding Author** : bentajer@outlook.com
ORCID: 0000-0002-1566-8527

Article Info:

DOI: 10.22399/ijcesen.370873
Received : 25 December 2017
Accepted : 09 March 2020

Keywords

Cloud Computing
SPI Stack
TPM
Trustiness
Multisig

Abstract:

Cloud computing defines the SPI model, which is generally agreed upon as providing Software-as-a-Service, Platform-as-a-Service and Infrastructure-as-a-Service. Interest in those service delivery models is growing because the paradigm offers to cloud customers high computational resource on-demand with a low cost. However, trustiness in the cloud services regarding the security and the privacy of the delivered data is the most critical issue in the SPI model. In this paper a trusted SPI model is proposed that gives cloud customer more confidence into SPI services by leveraging a trust in a neutral SPI certifying authority. The proposed model prevents insider attacks from tampering with application service before and after the computational resource was launched and allow cloud customer to verify if its node run in a secure environment.

1. Introduction

The benefits of cloud computing are very known by cloud costumers, notably the capability to use computing and storage resources on-demand with a lower cost. More than 3 million businesses have adopted Google Apps, Google cloud email. This rate is expanding at 3000 users a month [1]. Yet there is still considerable aversion to move to the cloud. Cloud customers are not willing to lose the control of their sensitive data and outsource the trust of computation of critical data to a non-trusted third party. The main concern is about the Confidentiality, Integrity and Availability. A malicious backend administrator may intentionally or accidentally tamper with the hosted data or inject a malicious code onto a Virtual Machine image (VMi), which will be later copied, to dig sensitive information. This can occur without the knowledge of cloud customer. Although many Cloud Providers (CPs) are making efforts in order to strengthen the trustiness of cloud customers by reducing the chances of an internal attacks. For example, CPs limit the access to the critical components of the infrastructure [2]. Nevertheless, malicious insider can still access cloud customers' sensitive data and tamper with them. When Xen is used in the backend, Xenaccess [3] enables backend administrator the access the content of a Virtual Machine's (VM's) memory at runtime. Another alternative for CPs, in order to reinforce the trust with cloud customer, aims to the use of cryptography to prevent confidentiality violations [17-19]. However, cryptography have shown some issues dealing with confidentiality. First, the CP holds both the keying material and encrypted data, given both items backend administrator can get encrypted data. Second, cryptography preserve confidentiality for data at rest and in-transit, and it cannot be applied to a public SaaS or PaaS, since encryption will prevent the indexation and computation of data. To establish more confidence

in their services, CPs have defined the specification of the Trusted Platform Module (TPM) [4]. This approach has been used by Santos et al. [2] to design a trusted cloud computing platform based on TPM attestation chain. Terra [5], a trusted computing platform can prevent malicious backend administrator from tampering with a computation. Terra also provides a remote attestation capability that enables a cloud customer to determine upfront whether the host can run securely the computation on its data. In [9], a trust system with certificate authority (CA) and Trusted Platform Module (TPM) is established.

This paper proposes a model for enhancing trust in cloud SPI (SaaS, PaaS and IaaS) stack, especially for public SaaS and IaaS. The model strengthens the trustiness of the cloud customers. For this purpose, the source code, of a public SaaS, of the application service must be certified by a neutral trusted certifying authority (TCA) in order to prevent the CP from tampering with application service, through to the use of the multisignature mechanism. The CP through the TPM, certifies also the source code. The operation prevents the misuse of service application when it is running in the cloud platform. For the public IaaS, the model will enable the cloud customer to check and authenticate a VMi (Virtual Machine image) created by the CP to ensure that it comes from the CP who claim to be the owner. The operation is done by digitally signing VMi by the TCA. The organization of the paper is as follows: Section 2 provides a background of TPM and Multisig schemes. Section 3 provides an overview of the Trusted SaaS Model. Section 4 presents the Trusted IaaS Module. This paper ends with a conclusion and discussion in section 5.

2. Background

2.1. Trusted platform module (TPM)

Trusted Computing Group (TCG) have defined the specification of TPM [4]. The TPM (Fig. 1) is a chip that contain a cryptographic co-processor, secure memory, I/O components and other components. It also implements a validated FIPS [8] key size generation for computing digital signature key (512 to 2024 bits). The chip can store an Endorsement Key (EK) that identify the TPM (Thus device authentication).

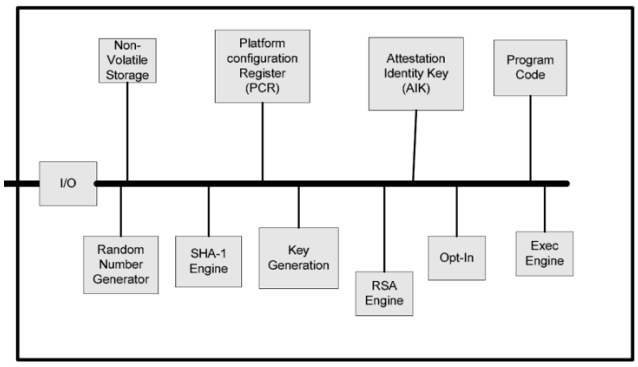

Figure 1. *TPM Architecture*

The chip is tampering resistant through its multiple physical security mechanism. Besides, the chip enables platform integrity; This is done during the boot process, the firmware and operating system components (integrity measurement) are measured (ML) and stored in the TPM. The integrity measurements can be used as evidence for how a system started. Once a platform needs to be attested the platform, a trusted third party sends a token to the platform. The platform takes both the ML and token encrypted by the TPM's EK and send the encrypted information to the third party which decrypt it using EK's corresponding public key (thus ensuring authentication) and checks the integrity of data (ML and Token) to verify that the configuration it deems trusted [4,7]. TPM enables certificate installation/creation on device where it is implemented. Once done, the RSA private key is engraved to the TPM and cannot be exported. Trusted platforms [5,6,9] leverage the features of TPM to enable remote attestation.

2.2. Multi-signature schemes

Multi-signatures schemes are a digital signature scheme that allows two or more signers to sign a single document as a group, yielding a multi-signature with the same size as a standard signature [21]. This allow them to jointly authenticate a document using a single compact signature [20]. By transmitting the joint signature instead of ***n*** individual signatures, multi-signature schemes help saving on communication costs. However, one still needs the public keys of all cosigners in order to verify the validity of such a multi-signature. The concept is used in the field of cryptocurrency for additional security in transactions. In a bitcoin

wallet, multiple signature is required in order to authorize a bitcoin transaction.

3. Trusted SaaS Platform

Earlier implementations of a multisignature σ on a message m is obtained by setting $(\sigma_i : i \in L)$ where σ_i is the signature of the i^{th} signer on the message m [11, 12]. This multisignature is however large, of size proportional to the number $|L|$ of signers. Moreover, Earlier multisignature schemes require a set up process between all the signers which make their use impracticable especially for devices with small resources such as PDAs, cell phones and TPM chips. Researches efforts have led to recent multisignature schemes [13, 14] that do not require nothing more than that each signer has a certified public key. Furthermore, these multisignature schemes have become as efficient as others signature schemes in both signing and verification process. The proposed SaaS platform aims to enhance trustiness in today's cloud computing services. It allows users of these services to have an independent insight into their behavior. The main components of the proposed platform are: a cloud provider (CP) that hosts application services on an infrastructure and platform that it controls, a service provider (SP) that launches the application service as an instance hosted on a cloud platform owned by CP and a neutral source code certifying authority (TCA) responsible for certifying a source code of an application service owned by SP. In our design the TCA uses a validated RSA key pair in accordance with FIPS-140 standards in order to issue certificates. The CP uses as private key the EK of the TPM. Related to the EK are Attestation Identity Keys (AIKs). An AIK is created by the TPM and linked to the local platform through a certificate for that AIK. This certificate is created and signed by a certificate authority (CA). A privacy CA allows a platform to present different AIKs to different remote parties, so that it is impossible for these parties to determine that the AIKs are coming from the same platform. In our model, the used AIK key represents the public key of the TPM regarding multisignature schemes [13,14].The code source certification of the application service by a TCA and the CP addresses the specific need to prevent the misuse of the application before and after it was launched. The use of multisignature schemas establishes a mechanism of double protection since it prevents attackers from SP and CP to act jointly or separately in order to misrepresent the behavior of the application. The attestation provided by the TCA ensures the integrity of the application when it was under the monitoring of the cloud provider. After launching the application, the SP assumes it's the control but it still the cloud provider that controls the server platform and the launch procedure. Thus, the cloud provider can be responsible for guaranteeing the application's integrity after it was launched by using the TPM certification. In various TPM-based platform, such as Intel's Trusted Technologies [16], the proposed functionalities have been extended to post-launch checks. This makes the task of the cloud provider more affordable. The TPM attestation consists of several steps of cryptographic authentication by which the specification for each layer of the platform is checked from the hardware up to the operating system and application code. At a high level, the TPM attests the source code of service application by signing its hash with an attestation identity key (AIK). This will be done by following the trust chain *TPM → BIOS → Bootloader → OS → Application*. Direct Anonymous Attestation (DAA) [15] can be used to protect the privacy of the TPM in such a way that a user will able to verify the validity of attestation without linking it with the platform that contains the TPM. The SP first (1) requests the CP to certify the source code of its application service (P). Once received, the SP (2) verifies whether the signature issued by the CP is valid or not. If it is valid then it forwards (P) sealed with the digital signature to the TCA (3), otherwise the service provider requests another signature. Once received, the TCA (4) first checks the signature provided by the CP. If it was valid, the TCA cheeks the source code of the application and then certifies it. This will be done by signing the digital signature issued by the CP via the multisignature mechanism. Otherwise, the TCA rejects the request. If all these steps had gone smoothly, the TCA (5) forward the result of the multisignature mechanism to the CP who made it available for the cloud users (Fig. 2).

4. Trusted IaaS Platform

The proposed IaaS platform aims to enhance trustiness of leased VMi. In their manual [10],

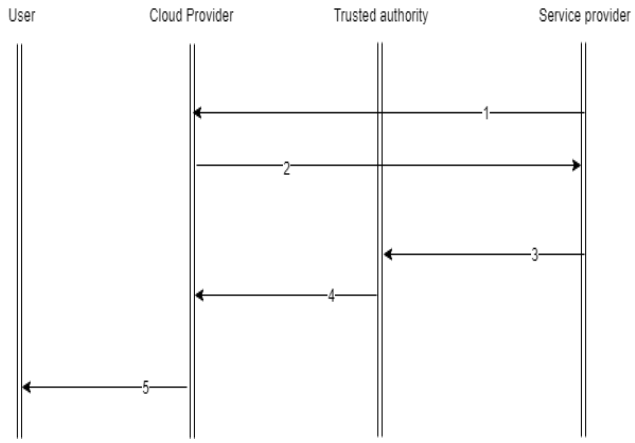

***Figure 2**. Trusted SaaS Model*

NIST have defined cryptographic key challenges tied to VMi authentication and deployment. When a cloud customer leases a VMi from the CP, he could be worried about its authenticity. To improve confidence with the cloud customer, the CP may digitally sign the VMi, and makes the public corresponding key available for cloud customer in order to check the authenticity. In [5,6] solutions focus more on secure computation of data and a closed box execution. In [9] authors propose a collaborative design to make effective use of TPM. Thus, the solutions suppose that cloud customer have already deployed the VM. Trust in IaaS platform start with the authentication and check of the pre-built VMi before its first deployment. Our design proposes the authentication and check functions and work as [6] or [9]. When the cloud customer leases a VMi from the CP, he could be worried about its authenticity. To enhance the trustiness with him, the CP may digitally sign the VMi and makes to corresponding public available for customer in order to check the authenticity. The problem with digital signature as mentioned by NIST [10] is that: (i) the cloud customer cannot have assurance that communication with CP are secure when exchanging data and results, (ii) Cloud customer does not have enough assurance about the credibility of verification engine running in the CP.

Fig. 3 presents the added value by the TCA to ensure the VMi check and authentication. Our design assumes that the part of TCA responsible of digitally signing the VMi is implemented as private cloud at the CP. Once the CP generates the VMi, the TCA compute the digest and store it securely. Every time that a cloud customer checks out the VMi, he can verify the digital signature by the EK's corresponding key, which he supplies to the verification engine, as illustrated in Fig. 3.

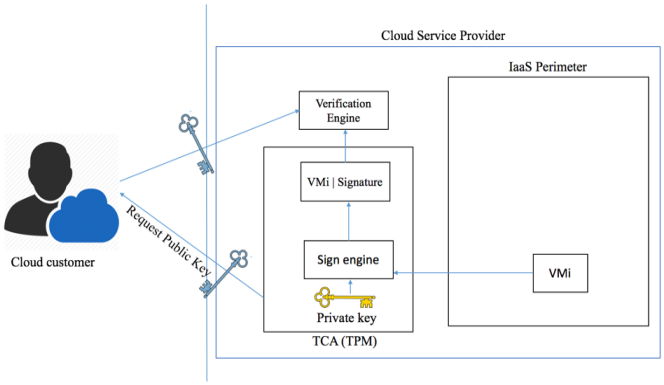

***Figure 3**. VMi Authentication and Check*

5. Discussion

Trusting a CP is mainly based on its reputation; therefore, it has no interest in the behavior of the service instance it hosts. It can strength assurances of trustworthiness to its customers by issuing attestations of its own software stack based on a hardware root of trust. The CP is also responsible for granting the privileges access (admin interface) to the SP. Thus, it can limit the control of the SP on the application after it was launched to only the legitimate operations (launch, stop, ...). For all these reasons, the CP itself can be seen as a trusted platform that run a well-tested software stack and offers hosting platform for multi tenants' users. This means that the CP in our model can serve as a guarantor for preventing the SP from tampering with application service after it was launched even if it was under its control. The mere fact that the CP acts as a root of trust in our model does not mean that it is completely trusted since it can tamper with application service before it was launched. This issue was addressed by incorporating a TCA source code authority that ensures the application's integrity when it was under the control of the CP. The TCA is responsible for certifying the source code of the application service jointly with the CP. This process prevents any attempt for subverting the instance before it was hosted in the cloud platform.
The VMi digital signature, enable cloud customer to check and authenticate the pre-built VMi to ensure that it comes from trusted source and have not been tampered with. To reinforce the trust with cloud customer the TCA digitally sign the VMi through the TPM. The operation has the advantage that the private key (EK) is securely stored, well protected

from unauthorized use or disclosure and protected while in use through TPM's facilities. Also, the design enables authentication of the signing authority and thus the authentication of the VMi.

6. Conclusion

In this paper we argue that concerns about trust computing is a major deterrent for enterprises looking to embrace cloud computing especially public SaaS and IaaS solutions. We surveyed the security concerns unique to the SPI delivery model. Also a new trusted SaaS model that addresses these security issues and reinforces the trustiness of users in SaaS services and enables authentication of leased VMi in IaaS model is proposed. In our SaaS model, the source code of the service application is certified by a neutral TCA and the CP using the multisignature mechanism. For the IaaS leased VMi, the VMi is digitally signed by the TCA and the cloud customer needs only to verify the digital signature using the AIK key of the TCA in order to check out the VMi. In the future, we plan to implement the full functionalities of our model and evaluate its performance and perform a comprehensive security analysis.

References

[1] Ron Zalkind. Protecting Your Data In Google Docs Compliance In The Cloud. http://hosteddocs.ittoolbox.com/protecting-your-data-in-google-docs.pdf.

[2] Santos et al. (2009) Towards trusted cloud computing. In Proceedings of the Workshop on Hot Topics in Cloud Computing, HotCloud'09. USENIX Association, 2009. http://portal.acm.org/citation.cfm?id=1855533.1855536.

[3] Lina Jia et al. (2017). T-VMI: Trusted Virtual Machine Introspection in Cloud Environments. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (CCGrid '17). IEEE Press, Piscataway, NJ, USA, 478-487. DOI: https://doi.org/10.1109/CCGRID.2017.48

[4] https://trustedcomputinggroup.org/tpm-main-specification/

[5] Tal Garfinkel et al. (2003). Terra: a virtual machine-based platform for trusted computing. In *Proceedings of the nineteenth ACM symposium on Operating systems principles* (SOSP '03). ACM, New York, NY, USA, 193-206. DOI=http://dx.doi.org/10.1145/945445.945464

[6] Krautheim F.J. et al. (2010) Introducing the Trusted Virtual Environment Module: A New Mechanism for Rooting Trust in Cloud Computing. In: Acquisti A., Smith S.W., Sadeghi AR. (eds) Trust and Trustworthy Computing. Trust 2010. Lecture Notes in Computer Science, vol 6101. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-642-13869-0_14

[7] Li B. et al. (2014) The Application of Trusted Computing Technology in the Cloud Security. In: Wong W.E., Zhu T. (eds) Computer Engineering and Networking. Lecture Notes in Electrical Engineering, vol 277. Springer, Cham. DOI : https://doi.org/10.1007/978-3-319-01766-2_99

[8] National Institute of Standards and Technology (NIST) FIPS PUB 186-4, Digital Signature Standard (DSS). DOI: http://dx.doi.org/10.6028/NIST.FIPS.186.4

[9] Yu, Z. et al. (2017). A Trusted Architecture for Virtual Machines on Cloud Servers with Trusted Platform Module and Certificate Authority. Journal of Signal Processing System vol:86. https://doi.org/10.1007/s11265-016-1130-9

[10] Chandramouli , R. et al. (2013). NIST IR 7956: Cryptographic Key Management Issues & Challenges in Cloud Services. DOI : http://dx.doi.org/10.6028/NIST.IR.7956

[11] Itakura, K. and Nakamura, K. (1983). A public-key cryptosystem suitable for digital multisignatures. In NEC Res. Development 71, pp. 1-8.

[12] Okamoto, T. (1988). A digital multisignature scheme using bijective public-key cryptosystems. *ACM Trans. Comput. Syst.* 6, 4 ,432-441. DOI: http://dx.doi.org/10.1145/48012.48246

[13] Bellare, M. and Neven, G. (2006) New multi-signatures and a general forking lemma. in CCS06. DOI: https://doi.org/10.1145/1180405.1180453

[14] Bellare, M. and Neven, G. (2007). Identity-based multi-signatures from RSA. In CT-RSA. DOI: https://doi.org/10.1007/11967668_10

[15] Brickell , E. et al. (2004). Direct Anonymous Attestation. In ACM Conference on Computer and Communications Security, pp. 132-145.

[16] Brown A., and Chase J. S. (2011) Trusted Platform-as-a-Service: A Foundation for Trustworthy Cloud-Hosted Applications. In: Proc. of CCSW. pp. 15-20 . DOI: https://doi.org/10.1145/2046660.2046665

[17] Tang Y. et al. (2012), Secure Overlay Cloud Storage with Access Control and Assured Deletion, IEEE Transactions on Dependable and Secure Computing, Vol 9, p. 903-916. DOI: https://doi.org/10.1109/TDSC.2012.49

[18] Saurabh S. et al. (2016) A survey on cloud computing security: Issues, threats, and solutions, Journal of Network and Computer Applications, Volume 75, November 2016, Pages 200-222, ISSN 1084-8045. DOI: http://dx.doi.org/10.1016/j.jnca.2016.09.002.

[19] Khalil, I., et al. (2014), Cloud Computing Security: A Survey, Computers, 3(1), 1-35; DOI: https://dx.doi.org/10.3390/computers3010001.

[20] Bellare, M. and Neven, G. (2006). Identity-Based Multi-signatures from RSA. Topics in Cryptology – CT-RSA. Lecture Notes in Computer Science. 4377. pp. 145-

162. CiteSeerX 10.1.1.207.2329. doi:10.1007/11967668_10. ISBN 978-3-540-69327-7.

[21] Itakura, K. and Nakamura, K. (1983) A public-key cryptosystem suitable for digital multisignatures. NEC Research & Development 71, 1–8.

Nomenclature

ABBREVIATION	SIGNIFICATION
AIK	**ATTESTATION IDENTITY KEYS**
CA	**CETIFICATION AUTHORITY**
CP	**CLOUD PROVIDER**
EK	**ENDORSEMENT KEY**
FIPS	**FEDERAL INFORMAITON PROCESSING STANDARDS**
IAAS	**INFRASTRUCTURE AS A SERVICE**
ML	**MEASUREMENTS LIST**
NIST	**NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY**
PAAS	**PLATFORM AS A SERVICE**
SAAS	**SOFTWARE AS A SERVICE**
SP	**SERVICE PROVIDER**
SPI	**SOFTWARE-PLATFORM-INFRASTRUCTURE**
TCA	**TRUSTED CERTIFYING AUHTORITY**
TCG	**TRUSTED COMPUTING GROUP**
TPM	**TRUSTED PLATFORM MODUL**