Available online at www.sciencedirect.com

ScienceDirect

SugarLoafPLoP 2018 00 (2018) 1–7

SugarLoafPLoP 2018

A Security Pattern for Cloud service certification

Antonio Muñoz, Javier Lopez

Abstract

Cloud computing is interesting from the economic, operational and even energy consumption perspectives but it still raises concerns regarding the security, privacy, governance and compliance of the data and software services offered through it. However, the task of verifying security properties in services running on cloud can be an arduous task. The provision and security of a cloud service is sensitive because of the potential interference between the features and behavior of all the inter-dependent services in all layers of the cloud stack (as well as dynamic changes in them). Besides current cloud models do not include support for trust-focused communication between layers. We present a security pattern that gives a solution for service certification based on the use of a key element trusted computing module.

Keywords: Cloud Computing, certification, Security Properties, Trusted Computing, TPM, Security Pattern

1. Introduction

Cloud computing has been developed with two main targets, that is, reducing IT costs and providing agile services to both users and organizations. The foundations of cloud settle in moving data away from desktops and laptops into large data centers. This fact potentially provides an increasing of innovation in limited devices in the form of innovative methods of business performance. We defend that before cloud computing is completely consolidated it is necessary to face some security issues favored cloud nature.

Cloud computing is not trivial since as a consequence of cloud complexity, a satisfactory level of security cannot be found in commercial clouds. In most cases, existing security solutions are difficult to apply and only can be used under barely restricted conditions. Many research initiatives have faced different aspects of security in these domains. However, most of them were unsuccessful since they based the application of traditional solutions straightly to the cloud. The lack of appropriate solutions implies that only partial solutions exists, it proves the emergent necessity of security approaches tailored for cloud computing idiosyncrasy. From wide spectrum of challenges to face we focus on a subset, for this reason we applied boundaries to our action field, as public clouds.

Certification provides a mechanism to support assurance and compliance, but its adoption to cloud service certification is not as straightforward. Certification has been represented for human beings and not supported for automated processing of certified objects. Besides it is limited to static cases. Current certification schemes do not provide dynamic verification of system at runtime. We claim this interesting feature as essential for dynamic and unpredictable scenario as we found services running in clouds. There are approaches that have addressed the first problem by using computer oriented formats, processes and tools to support the automated validation of certification and selection of services based on their certificates. Nevertheless the second an open problem, at least there is not a satisfactory

[1] This work is part of the CHAPATA project as Marie Curie action from H2020 proposal

Email addresses: amunoz@lcc.uma.es (Antonio Muñoz), jlm@lcc.uma.es (Javier Lopez)

solution. The approach presented provides a solution for dynamic system verification at runtime using TPM [2] as a security pattern.

As it was previously pointed out, this paper presents a solution built on a combination of software certification and hardware based certification techniques [1]. The cornerstone in our model is Trusted Computing technology, we take advantage of its functionalities as secure element. TPM becomes the anchor of our certification chain. Consequently, bringing the gap existing between the software certification and the means for hardware certification becomes as a target. Since the solutions based on Trusted Computing tends to be hardly to implement in real scenarios, we present a security pattern [5, 6].

Evidence communication is supported by Trusted Computing (TC) technology, in particular we implement a mechanism that provides means to establish integrity (authenticity) of evidence, and subsequently verify if the captor integrity holds (can be trusted). Whenever possible, evidence gathering is build upon existing standards and practices (e.g., interaction protocols, representation schemes etc.) regarding the provision of information for the assurance of security in clouds.

We claim the necessity of a binding mechanism as a foundation for service certification as we pointed out. In our binding approach, each service is pledged to operate with a key pair maintaining linked to a pledge. This mechanism implies that service providers can be made legally responsible for using the key pair (only with the pledge service). From security perspective, we define this as one of the strongest points of our approach. Considering key pair resides in TPM and it is bound to pledged configuration of the service. When the service is called, TPM attestation is triggered to measure complete service configuration. This sets up key as available to the service, and allowing attest the integrity of the underlying platform (infrastructure, VM, OS, and every layer involved). Thus, when service status changes a new measurement is taken (new the platform state checking). This will not successfully complete and the key will not be available to preserve the integrity and non repudiation. Every service request is then signed using service private key. To enable to have different configurations, each group of services that share infrastructure is executed in a different virtual machine, this provides encapsulation.

The pattern presented in this paper gives functionality for the generation of hybrid certificates based on the combination of different types of evidences (including testing and monitoring data, and trusted computing platform proofs). It leads to cover security properties to an unprecedented extent and increase the overall confidence in the use of cloud services.

2. A Security Pattern for cloud service certification model using TPM

2.1. Intent

A recurrent problem in cloud environments is the security of services that can be faced using certification mechanisms. However, current certification mechanisms require human interaction at any moment in the whole process. We propose a security pattern that provides a solution for service certification using cryptographic hardware as anchor.

2.2. Context:

It describes a recurrent environment in which services running on clouds require a certification mechanism that is not covered with existing rigid certification schemes.

2.3. Problem description:

Certification provides a mechanism to support assurance and compliance, but its adoption to cloud service certification is not as straightforward. Certification has been represented for human beings and not supported for automated processing of certified objects and limited to static cases. Current certification schemes do not provide dynamic verification of system status at runtime. In these terms our solution should gather the following forces:

- A certification stack that defines the process of engineering and developing systems (services and applications) is needed.

- Certification approach that includes analyzed software to certify, identifies and specify runtime proofs to generate the certificate is required.

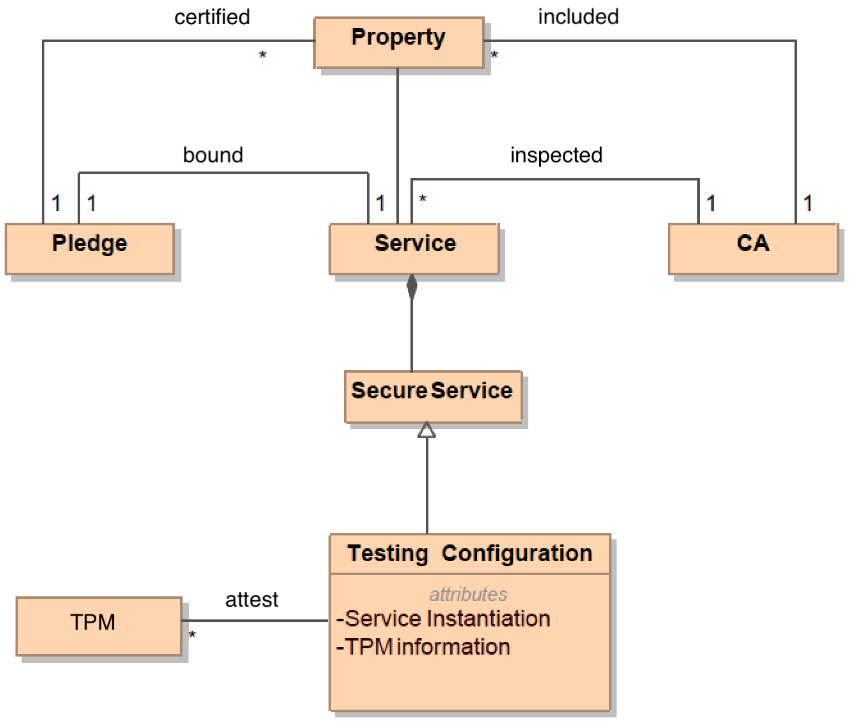

Figure 1. Cloud Certification Class Diagram

- It is needed the pledge generation process that authorize the certificate and generate pledge.
- The process that defines how to use pledge by the clients should be included.

2.4. Solution:

Certification stack is shown in figure 3. This defines the process of engineering and developing systems (services and applications). This process includes some elements as security aspects (not only traditional based on functional aspects) and certificates to establish trust relationships. This security pattern provides an alternative to service certification in clouds to traditional mechanisms where human interaction is required.

2.4.1. Structure:

Figure 2 shows some dynamic aspects shown as sequence diagrams for pledge generation use case. Also figure 1 shows the class diagram for a pledge generation use case. CA inspects a particular service and extracts particular properties. This process adds testing configuration as relevant information for secure services.

Certification Authority (CA) inspects a service to certify and includes those properties that are required. Pledge can then be used to certify properties inspected by CA and maintain bound to the service. Henceforth TPM can be used to attest a particular Testing Configuration (instantiation of service and TPM information).

2.5. Dynamics:

Figure 2 describes a sequence diagram for one of our use cases, this shows aspects from pledge generation use case. The sequence should be started by service provider, who pretends initiates the certification of his services in cloud infrastructure (Infrastructure As A Service). CA inspect the service instance together with available properties to update pledge content. CA is to fill the pledge structure, checking and matching properties and service inspection feedback. TPM resources are used to take measures of service current state. Public key, migratable key and signature are generated using TPM functionalities and included in the pledge.

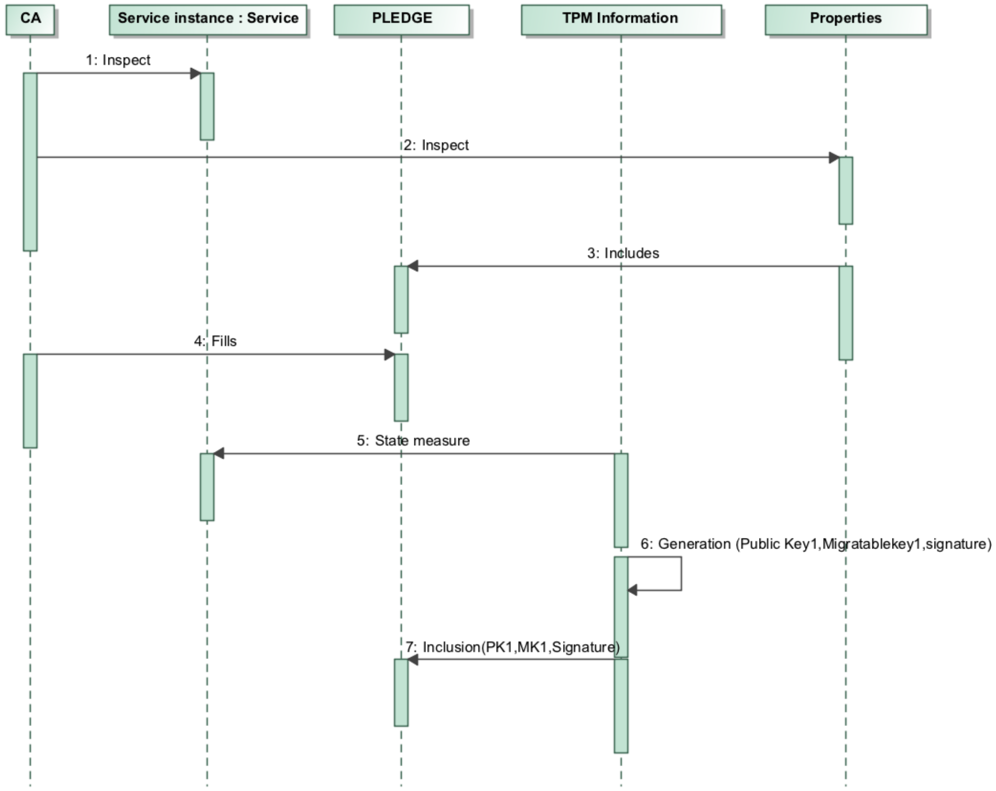
CA
Service instance : Service
PLEDGE
TPM Information
Properties
1: Inspect
2: Inspect
3: Includes
4: Fills
5: State measure
6: Generation (Public Key1,Migratablekey1,signature)
7: Inclusion(PK1,MK1,Signature)

Figure 2. Cloud Certification Sequence Diagram

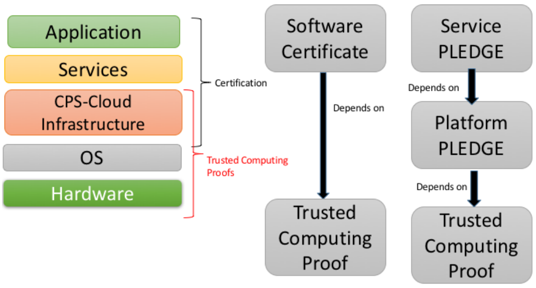

Figure 3. Certification Stack.

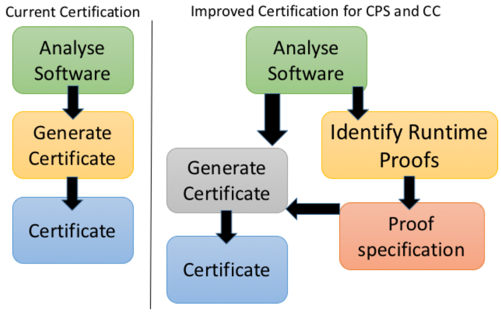

Figure 4. Certification Approach

2.6. Implementation:

Our solution can be defined as an orchestration of different technologies within a mediation layer. Among these, we highlight the role of Trusted computing (TC) technology. TC is essential to attest both the hardware and the native OS. Also TC attests software certificates used for higher leveled applications and services. Our solution relies on the sealed bind key functionality provided by trusted computing technology.

Assurance of cloud services allows service consumers and providers to ascertain that the service properties provided in the certificates guarantee continuous compliance with their own requirements. This enhanced mechanism increases the confidence of both consumers and providers that their required level of assurance is being kept, before becoming involved in service design, deployment, and access on cloud.

An overview of the workflow is following described as; a sealed bind key is used to encrypt part of the code of the service. This mechanism enables that it can be only used when platform state is preserved unchanged. We have designed this solution being aware of their restrictions, but it provides a high level of security allowing to establish a limited execution environment. In spite of limitations, which should hinder its integration in real world scenarios, but a tailored solution based on this scheme can be suitable for particular cases. We propose after a previous study of the case. We propose relaxing initial restricted conditions, which enables achieving valuable secure levels. Besides a relaxed version could be adapted for cases with lower security requirements with positive prediction outcomes.

We have included of two use cases descriptions to a better understanding how our pattern works; pledge generation (where the certificate authorization is involved) and pledge usage use cases. We introduce our pledge concept as a semantic description for certifying services. This is composed by two well differentiated parts, the standalone and SAML container. The pledge standalone representation and SAML container specific representation. Resuming use cases, the first step is service evaluation, which includes CA checking. This inspects a list of properties that must be fulfilled and inspecting the service. This triggers that CA fills pledge form with feedback information sent. Binding platform implies the creation of a key pair (using a sealed key as seed), this sealed key is bind to the state of the platform preserving platform integrity.

The TPM infrastructure is used to supply a foundation where all cloud certification chains of trust can be grounded, adding a major trusted capabilities to certification location attestation. TPM based certificates provide services the possibilities to show proofs, including a variety of strong authentication and data security mechanisms, demonstrating compliance with numerous regulations.

Certification models should also provide means to combine several evidences in an integrated framework, establishing the foundations for the definition of hybrid and incremental certificates. We highlight the incremental certification as particularly important features. When the evidence from a certificate is enough to verify the security property related to it (as determined by the certification model). The a certificate is issued as an instance of this type. Novelty of this approach is that even after certificate is issued, it can be updated subject to changes in the operational conditions of the platform. As systems are being composed not only based on functional aspects, but also on security aspects (properties, threats, risks, etc.), trust is established by means of certificates. Also components of a system may change without the knowledge or control of other components.

Certification stack shown in figure 3 proposes the novelty of an engineering. New systems are composed not only based on functional aspects, but also on security aspects (properties,threats, risks,etc). A key element is used to establish trust relationships, that is, certificate that enables that components of a system may change without the knowledge or control of other components. TPM provides secure storage and key pair resides in TPM; and is bound to the pledged configuration of the service. When the service is called, the TPM attestation functionality is used to attest the (complete) service configuration, this has been applied in different scenarios i.e. mobile agents [4, 7]. At this execution point, key sets are available to the service. If service changes TPM functionality is used to attest the state, then the checking fails and we can assume that the key will not be available anymore. Every response to a service call is signed using service private key. We consider that it is important to provide the capability of grouping services in terms of functionality. For this purpose, each group of services (sharing infrastructure) is executed in a virtual machine. This solution implies some restrictions, among them the hardest one is TPM equipped hardware (or even virtualized [3]).

2.7. Example Resolved:

We were able to design a solution for the certification of services running on cloud. Figure 3 shows our certification model stack proposing a solution that combines Trusted Computing Proofs with software certificates. It includes pledge concept as a certification element that bridges the gap between hardware proof and software certificate. As we show in figure 4 this enhanced service certification mechanism includes two essential steps before the certificate generation actually takes place. Runtime proofs have to be identified and related proofs properly specified.

2.8. Consequences:

We pretended to achieve an approach that includes a certification stack that defines the process of engineering and developing systems. Figure 3 shows that certification stack vs the infrastructure stack of a service. Figure 4 depicts our certification approach including steps as analysis of software to certify, identification and specification at runtime of proofs and certificate generation. Pledge are issued authorizing the certificate, a complete description of pledge definition is not needed to understand how our patterns works and is out of the scope of this paper. Likewise, the pledge usage process by the clients is out of the boundaries of this paper and currently is ongoing work.

2.9. Known Uses:

Some authors define that to accept our solution as a pattern, we should find at least three examples of its use in real systems. However, there are some exceptions to this rule when the solution is clearly generic, as our pattern obviously is. This model faces the problem of certification of services in cloud avoiding the human inspection in every step. Generic nature of our solution makes easy to include real examples in which a direct implementation of our pattern can take place.

3. Conclusions & Ongoing Work

This paper proposed a pattern that provides a solution for service certification in cloud computing using a trusted hardware module. The proposed scheme can successfully bridge the gap between Trusted Computing and Software Certification by combining the best of both worlds and overcoming their respective limitations. The concept of pledge as a computer oriented form of certification is an essential key for improving the flexibility and practical applicability of TC mechanisms. Besides opening possibilities to explore future applications for Trusted Computing technology.

We propose a discussion of a generic life cycle model including a variety of possible updates with other key changes throughout life cycle of incremental certificates. Ongoing work includes the description of trusted computing technology as a security pattern itself as a complement to the approach presented throughout this paper and the composition of both patterns as an unified solution. We based on our initial definition of pledge concept, a further definition of pledge extent and its language is currently under study and being addressed as part of CHAPATA project.

References

[1] Muñoz A., Maña A. Software and Hardware Certification Techniques in a Combined Certification Model. International Conference on Security and Cryptography (SECRYPT). pp 405-410. 2014.

[2] Trusted Computing Group: TCG Specifications.

[3] Stefan Berger, Ramón Cáceres, Kenneth A. Goldman, Ronald Perez, Reiner Sailer, and Leendert van Doorn. 2006. vTPM: virtualizing the trusted platform module. In Proceedings of the 15th conference on USENIX Security Symposium - Volume 15 (USENIX-SS'06), Vol. 15. USENIX Association, Berkeley, CA, USA, pages.

[4] Muñoz, Antonio; Maña, Antonio. TPM-based protection for mobile agents.Security and communication networks. SI 4(1):45-60, 2011.

[5] B. Gallego-Nicasio, A. Muñoz, A. Maña, D. Serrano, Security patterns, towards a further level, in: E. Fernández-Medina, M. Malek, J. Hernando (Eds.), SECRYPT, INSTICC Press, 2009, pp. 349-356.

[6] Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., Sommerlad, P. (2006). Security Patterns: Integrating Security and Systems Engineering. Systems Engineering.

[7] Muñoz, Antonio; Maña, Antonio, Anton, Pablo. In the Track of the Agent Protection A Solution Based on Cryptographic Hardware. 5th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security Ubicacin: St Petersburg, Russia, sept. 08-10, 2010.