A Security Pattern for Cloud service certification

ANTONIO MUÑOZ, University of Malaga
JAVIER LOPEZ, University of Malaga

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 is not trivial. We notice 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 mechanism to implement cloud service certification process based on the usage of Trusted Computing technology, by means of its Trusted Computing Platform (TPM) implementation of its architecture. Among many security security features it is a tamper proof resistance built in device and provides a root of trust to affix our certification mechanism. We present as a security pattern the approach for service certification based on the use TPM.

Categories and Subject Descriptors: D.2.11 [**Software Engineering**] Software Architectures –Patterns; K.6.5 [**Management of Computing and Information Systems**] Security and Protection

General Terms: Security patterns

Additional Key Words and Phrases: Cloud Computing, certification, Security Properties, Trusted Computing, TPM , Security Pattern

ACM Reference Format:

Muñoz, A. and López, J. 2018. A Security Pattern for Cloud service certification. HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. XX (November 2018), 8 pages.

1. INTRODUCTION

Cloud computing has been developed with two main targets; 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 claim that before cloud computing is completely consolidated it is necessary to face some security issues favored cloud nature.

Although security in clouds have improved in recent years, current cloud computing implementations present security challenges still open. Among others, we highlight client's data are available to third party, which implies extra care while storing our important data on the cloud[Security 2016]. Other challenges are derived from poor portability facility provided by the Cloud Service Provider (CSP) derived in locked clients with one specific CSP & depend upon them for all kind of services [Albugmi et al. 2016]. And insecure or incomplete client's data deletion, some data from cloud then it is possible that data may not be deleted because of duplicate data may exist on the cloud [Albugmi et al. 2016]. We conclude that level of security in commercial clouds can be improved, in particular at client side.

We defend that open security challenges are still open since traditional security solutions are difficult to apply in current cloud implementations. Indeed most of them only can be used under barely restricted conditions.

There may be a possibility existing that information belonging to different customers resides on the same data server since different users are sharing a cloud provider platform. This implies that information leakage may arise unintentionally when information for one customer is given to another customer. Prior cloud computing access control mechanisms were based on the assumption that only information from a customer resided on the same data server under own customer's control. However, this assumption is not applicable now. In many cases, security solutions applied in traditional computing are not enough to provide security in cloud computing since new breaches and threats appears within this paradigm. Many research initiatives have faced different aspects of security in these domains, but some of them were unsuccessful since they were based on implementing traditional solutions straightly to the cloud. This fact implies that only partial solutions are applied in some cases, then a necessity of security approaches tailored for cloud computing for particular issues raises. Moreover, hackers are spending substantial time and effort looking for ways to find vulnerabilities in the cloud infrastructure that would allow them to penetrate the cloud [Liu et al. 2012].

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 a 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 is an open problem, at least there is not a satisfactory solution. The approach presented gives an approach for dynamic system verification at runtime using TPM [Group 2005] as a security pattern. TPM stands for Trusted Platform Module, TPM was conceived by a computer industry consortium called Trusted Computing Group (TCG) and is an international standard for a secure cryptoprocessor, a dedicated microcontroller designed to secure hardware through integrated cryptographic keys.

As it was previously pointed out, this paper presents an approach built on a combination of software certification and hardware based certification techniques [Muñoz and Maña 2014]. 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 secure systems based on Trusted Computing tends to be hard to implement in real scenarios, we present a security pattern [Gallego-Nicasio et al. ; Schumacher et al. 2005].

We propose an approach 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. A particular implementation is built using Trusted Computing (TC) technology supporting evidence communication.

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 to 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 isolation.

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 the certification of services.

—Certification is considered a robust mechanism for many security problems.

—Human interaction is required in any certification process that makes unpractical for services in cloud.

—Despite of current certification mechanisms require human interaction at any moment in the whole process.

2.2 Context:

Current cloud environments propose an interesting alternative to migrate corporate data with many functionalities and evident efficiency improvements, and even security offered by cloud providers is enough it can be considerably improved. 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. In terms of efficiency this requirement makes inapplicable in most of cases.

2.3 Problem description:

Current certification schemes do not provide dynamic verification of system status at runtime. In these terms our approach 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.

—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 1. 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. Different colors in figure 1 for stack match with boxes in figure 2 where an improved certification process is described. In this line, Analyse Software green boxes are conducted at Application level from figure 1 stack in green color, so on and so forth.

2.4.1 *Structure:.* Figure 3 shows the class diagram for a certification and proof binding use case. CA inspects a particular service (Analyse Software from figure 2) and extracts particular properties (Identify Runtime Proofs from figure 2). Proof specifications are produced using TPM functionalities. Pledges bind certified properties to services and proof specifications (Generating certificate from figure 2). This process adds testing configuration as relevant information for secure services.

Certification Authority (CA) is the entity that conducts the inspection of a service to be certified. CA includes required properties as part of the Pledge. A Pledge is a complement to the certificate that using semantics together

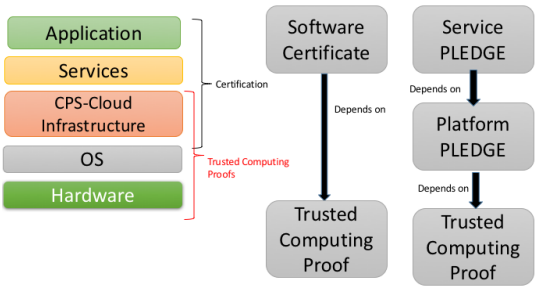

Fig. 1. Certification Stack

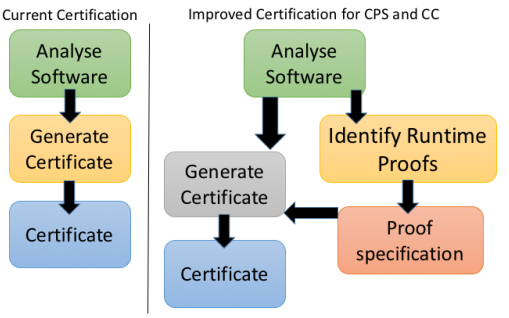

Fig. 2. Certification Process

are used to implement service binding. Pledges can then be used to certify properties inspected by CA and maintain bound to the service. Henceforth, TPM (stands for Trusted Computing Plaftorm) can be used to attest a particular Testing Configuration (instantiation of service and TPM information).

2.4.2 *Dynamics:.* Figure 4 describes a sequence diagram for one of our use cases, this shows aspects from binding use case initiates by Cloud Provider not included to simplify the figure. 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

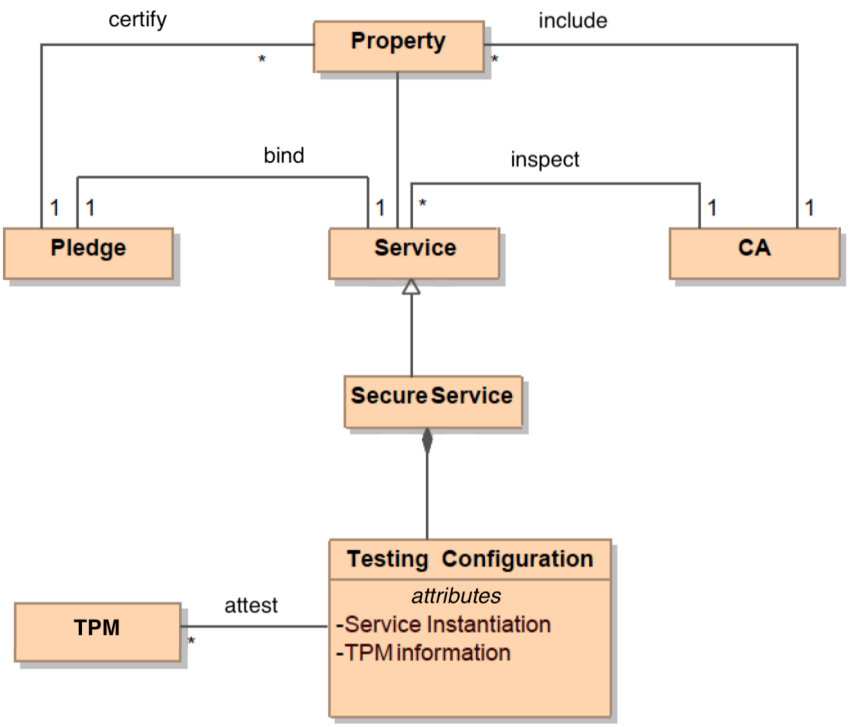

Fig. 3. Cloud Certification Class Diagram

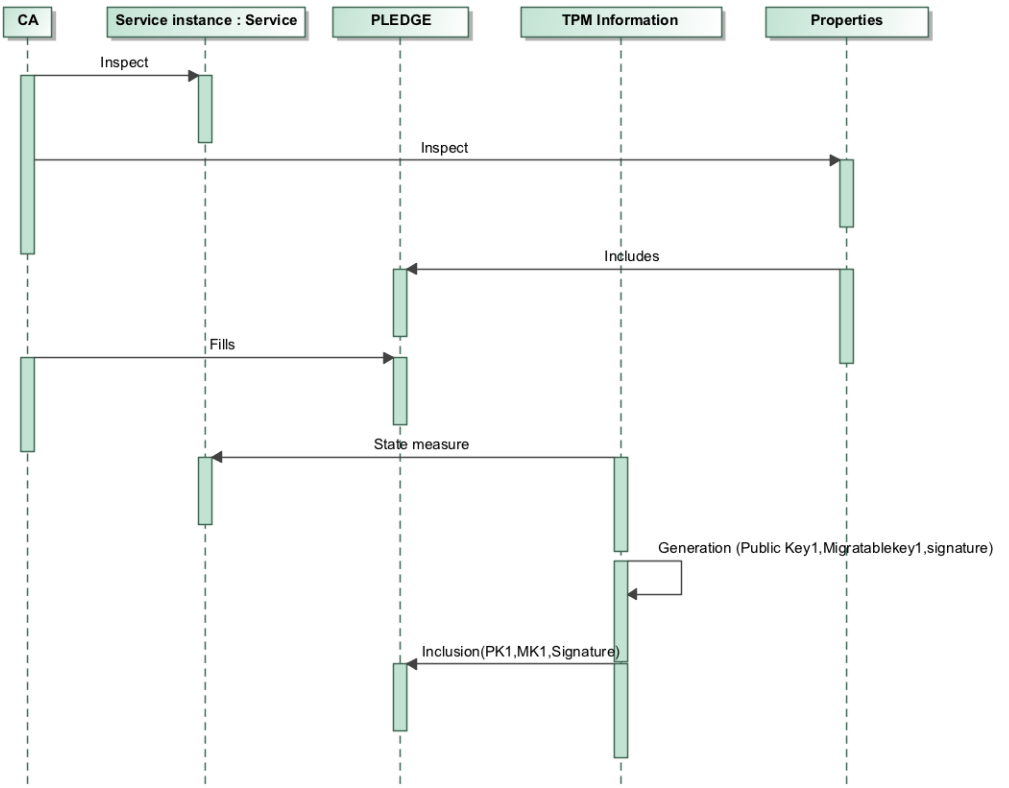

Fig. 4. Cloud Certification Sequence Diagram

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.

2.5 Implementation:

Our approach 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 design 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 conducted the design of our approach considering 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 approach 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 binding 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 1 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 [Muñoz and Maña 2011; Muñoz et al. 2010]. 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 approach implies some restrictions, among them the hardest one is TPM equipped hardware (or even virtualized [Berger et al.]).

2.6 Example Resolved:

We provide an approach for the certification of services running on cloud. Figure 1 shows our certification model stack combines Trusted Computing Proofs with software certificates. As a bridge to the gap between both elements we included a new concept, that is pledge. As we show in figure 2 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.

2.7 Consequences:

This pattern provides the following advantages:

—A certification approach for services in cloud computing.

—Identification and specification at runtime of proofs and certificate generation.

—Hardware proofs combined with software certificate to conduct a comprehensive process.

—A binding mechanism to combine proofs and certificate within pledge concept.

—Mechanisms for analyzing software (service) to certify.

Liabilities include:

—Hardware requirements restrict the usage of this patterns, such as a TPM embedded is needed.

—Pledge issuing and management entail additional steps that makes certification more complex.

2.8 Known Uses:

We pretended to achieve an approach that includes a certification stack that defines the process of engineering and developing systems. Figure 1 shows that certification stack vs the infrastructure stack of a service. Figure 2 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.

Some authors define that to accept our approach 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 approach 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 approach makes easy to include real examples in which a direct implementation of our pattern can take place.

2.9 Related Patterns

—Some patterns related to provide security for cloud environments as the misuse patterns for cloud computing [Hashizume et al. 2011].

—Cloud Resource Access Control pattern [Erl et al. 2015] is related to the pattern proposed in this paper.

3. CONCLUSIONS & ONGOING WORK

This paper proposed a pattern that provides a approach 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.

REFERENCES

A. Albugmi, M. O. Alassafi, R. J. Walters, and G. Wills. 2016. Data security in cloud computing. In *Fifth International Conference on Future Communication Technologies, FGCT 2016, London, United Kingdom, August 17-19, 2016*. 55–59. DOI:http://dx.doi.org/10.1109/FGCT.2016.7605062

S. Berger, R. Cáceres, K. Goldman, R. Perez, R. Sailer, and L. van Doorn. vTPM: virtualizing the trusted platform module. In *In Proceedings of the 15th conference on USENIX Security Symposium*.

T. Erl, R. Cope, and A. Naserpour. 2015. *Cloud Computing Design Patterns*. Prentice Hall. https://books.google.es/books?id=MjHYoQEACAAJ

B. Gallego-Nicasio, A. Muñoz, A. Maña, and D. Serrano. Security patterns, towards a further level. In *In Proceedings of the SECRYPT 2009*.

Trusted Computing Group. 2005. TCG Specifications. (2005). http://www.trustedcomputinggroup.org/home/

K. Hashizume, N. Yoshioka, and E.B. Fernandez. 2011. Misuse Patterns for Cloud Computing. In *Proceedings of the 2Nd Asian Conference on Pattern Languages of Programs (AsianPLoP '11)*. ACM, New York, NY, USA, Article 12, 6 pages. DOI:http://dx.doi.org/10.1145/2524629.2524644

F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf. 2012. *NIST Cloud Computing Reference Architecture: Recommendations of the National Institute of Standards and Technology (Special Publication 500-292)*. CreateSpace Independent Publishing Platform, USA.

A. Muñoz, A. Maña, and P. Antón. 2010. In the Track of the Agent Protection: A Solution Based on Cryptographic Hardware. In *Computer Network Security*, Igor Kotenko and Victor Skormin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 284–297.

A. Muñoz and A. Maña. 2011. TPM-based protection for mobile agents. *Security and communication networks* 4, 1 (2011), 45–60.

A. Muñoz and A. Maña. 2014. Software and Hardware Certification Techniques in a Combined Certification Model. In *International Conference on Security and Cryptography (SECRYPT)*. 405–410. DOI:http://dx.doi.org/10.5220/0005029102380243

M. Schumacher, E. Fernandez, D. Hybertson, and F. Buschmann. 2005. *Security Patterns: Integrating Security and Systems Engineering*. John Wiley & Sons, Inc., USA.

Calyptix Security. 2016. Top 5 Risks of Cloud Computing. (2016). https://www.calyptix.com/research-2/top-5-risks-of-cloud-computing