A Security Pattern for Cloud Service Certification

ANTONIO MUÑOZ, University of Malaga
JAVIER LÓPEZ, University of Malaga
SIGRID GÜRGENS, Fraunhofer SIT

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. The 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). Also current cloud models do not include support for trust-focused communication between layers. 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. We present a mechanism to implement a cloud service certification process based on the use of Trusted Computing technology, by means of a Trusted Computing Platform (TPM) implementation of its architecture. Tamper proof resistance built in device is a requirement that 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 of a TPM.

Categories and Subject Descriptors: D.2.11 [**Software Engineering**] Software Architectures –Patterns; D.5.1 [**Security and Protection**] ; D.4.6 [**Authentication, Authorization, Logging**] ; 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. et al., 2018. A Security Pattern for Cloud Service Certification. HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. XX (November 2018), 9 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 has as objective data away from desktops and laptops into large data centers. This fact potentially provides an increase 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 related to the nature of the cloud.

Although security in clouds has improved in recent years, current cloud computing implementations present many security challenges. Among others, we highlight client's data are available to third party which cloud imply extra care while storing our important data on the cloud. Other challenges starts from the low portability facilities provided by the Cloud Service Provider (CSP) derived in locked clients with one specific CSP depend upon them for all kind of services [A. AlbugmiA. Albugmi2016]. Also 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

Author's address: A. Muñoz, J. López. Computer Science Department, University of Malaga, Ada Byron building, 29071 Malaga, Spain; email: {amunoz,jlm}@lcc.uma.es; S. Gürgens. Fraunhofer – Institute for Secure Information Technology SIT, Rheinstrasse 75, 64295 Darmstadt, Germany; email: Sigrid.Guergens@sit.fraunhofer.de

cloud [A. AlbugmiA. Albugmi2016]. We consider that the level of security in commercial clouds can be improved, especially at the client side.

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 restricted conditions. There may be a possibility that information belonging to different customers resides on the same data server since different users are sharing a cloud provider platform. Thus 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 for every cloud environment.

In many cases, traditional security solutions are not enough to provide security in cloud computing since new breaches and threats appear within this paradigm. Many research initiatives have faced different aspects of security in these domains. Even though some of them were unsuccessful since they were based on traditional security solutions straightly applied to the cloud. Thus only partial solutions are successfully applied in some cases, arising the necessity of additional security approaches tailored for cloud computing. 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 [F. LiuF. Liu2012].

Certification provides a mechanism to support assurance and compliance, but its adoption to certify cloud services is not 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 believe this interesting feature is essential for dynamic and unpredictable scenario as met in clouds. Existing approaches 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 attempts dynamic system verification at runtime using a TPM [GroupGroup2005] (Trusted Platform Module). TPM was conceived by a computer industry consortium called Trusted Computing Group (TCG) and is an international standard for a secure crypto-processor, a dedicated micro-controller 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 [A. MuñozA. Muñoz2014] as a security pattern. 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 software certification and the means for hardware certification becomes a target. Since the secure systems based on Trusted Computing tends to be hard to implement in real scenarios, we present a security pattern [B. GallegoB. Gallego2009, EduardoEduardo2013].

Our approach provides means to establish integrity (authenticity) of evidence, and subsequently verify if the service host integrity holds (can be trusted). Whenever possible, evidence gathering is built 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.

As indicated earlier, we claim the necessity of a binding mechanism as a foundation for service certification. In our binding approach, each service is pledged to operate with a key pair 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 a security perspective, we define this as one of the strongest points of our approach. Considering that a key pair resides in TPM and it is bound to pledged configuration of the service these are use to preserve pledge integrity and confidentiality. Thus 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. 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 are executed in a different virtual machine, which provides isolation.

The pattern presented in this paper provides 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. Attempt this is an original idea, we consider it a pattern because of its structured description and generic use.

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.

—Provides means to establish integrity (authenticity) of evidence.

—Mechanisms to verify if the captor integrity holds (can be trusted) are provided.

2.2 Context:

Current cloud environments propose an interesting alternative to migrate corporate data with many functionalities and evident efficiency improvements, and even when the 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 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 it inapplicable in most cases.

2.3 Problem description:

Current certification schemes do not provide dynamic verification of system status at runtime. For this reason, it is required human interaction to implement current certification schemes. Cloud services may vary slightly and they need a new certification with the inherent delay of human interaction that decrease the dynamism expected for cloud services. The forces associated with this problem are as follows:

2.3.1 *Forces:*

—Level of security increased: We want to increase the level of security of web services in clouds by means of integrating a tailored solution based on certification. Data confidentiality, service confidentiality, data integrity and service integrity are granted using our certification model.

—Flexibility and usability: We want to increase usability using a certification approach that includes analyzed software to certify, identifies and specify runtime proofs to generate the certificate, but avoiding the necessity of human interaction to complete the certification process. In the same line, system flexibility increases since it is not required the human interaction at some point to resume the process, but it is automated.

2.4 Solution:

The certification stack is shown in figure 1, which defines the process of engineering and developing systems (services and applications). This process includes some elements as security aspects (not only traditional based

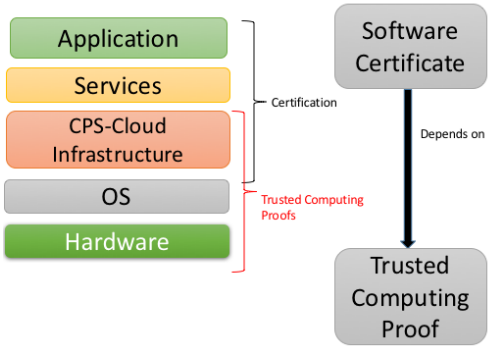

Fig. 1. Certification Stack

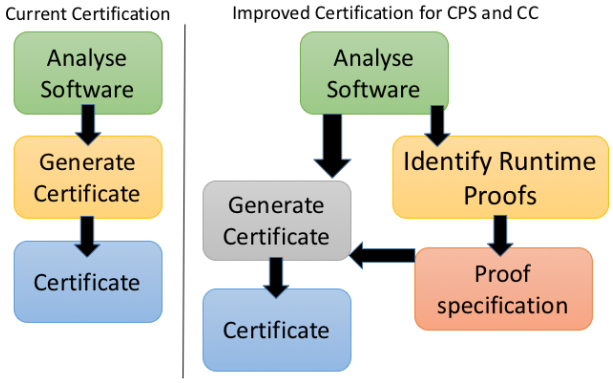

Fig. 2. Certification Process

on functional aspects) and certificates to establish trust relationships. The different colors in figure 1 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.

Figure 1 shows how software certificates are used to certify applications, services and some parts of CPS-Cloud infrastructures, in some cases. Trusted Computing proofs are used to provide certification of hardware, OS and other parts of CPS-Cloud infrastructures. In figure 2 we distinguish between current certification models that are based on generating a certificate; And our enriched certification that includes the identification of runtime proofs and proof specifications since this solution makes a combined use of the software certificate and TC proofs.

2.4.1 *Structure:.* Figure 2.4.1 shows the class diagram of our approach. This diagram shows describes Service as a Class, which has certain security properties (Property class) included by the Certification Authority (CA) after inspecting the service. Properties are used to certify Pledges to bind to Services for selected properties. A pledge is a new element that complements to the certificate that using semantics together are used to implement service binding. Pledges can then be used to certify properties inspected by CA and maintain bound to the service. Secure Service Class inherits of Service and is compound of Testing Configuration (class with two key attributes, Service instantiation and TPM information). TPM class describes a TPM interface in charge to attest Testing Configuration attributes.

Diagram 2.4.1 shows a certification and proof binding are conducted. CA (Certification Authority) inspects a particular service (Analyze 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. Testing configuration class is not included in diagram 2.4.1 since data provided by this class are required to set up the service and TPM, but cryptographic operations (key issuing, encryption, decryption, etc.) and measurement states are conducted within TPM.

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. Henceforth, TPM (Trusted Computing Platform) can be used to attest a particular Testing Configuration (instantiation of service and TPM information).

2.4.2 *Dynamics:.* Figure 2.4.1 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 initiates the certification of his services in the cloud infrastructure (Infrastructure As A Service). CA inspects the service instance together with its available properties to update the pledge

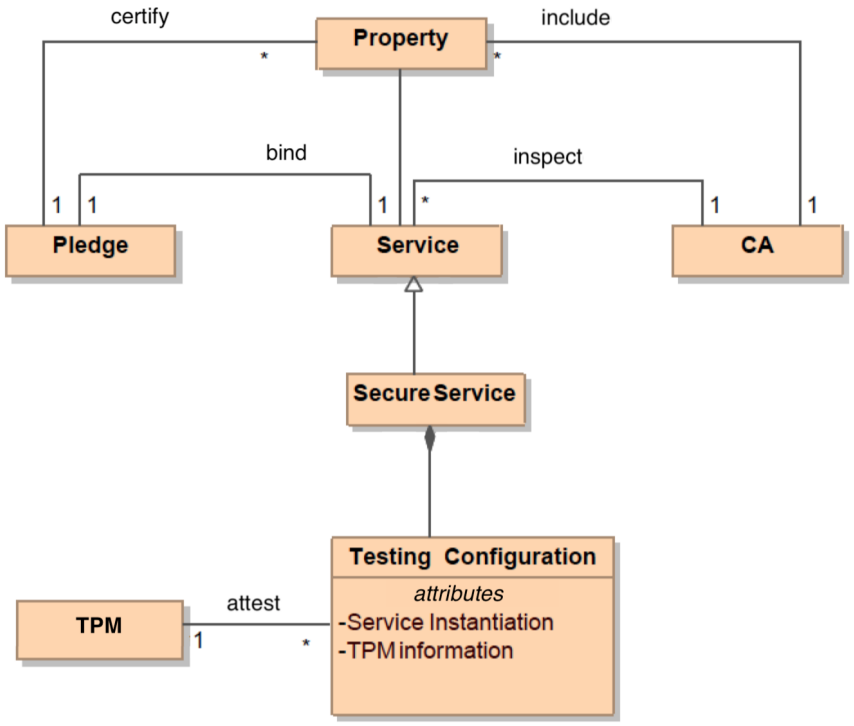

Fig. 3. Cloud Certification Class Diagram

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.

2.5 Implementation:

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.

Our approach implies the usage of different technologies within a mediation layer. Among these, we highlight the role of TPM as essential to attest both the hardware and the native OS. Also TPM is used to attest software certificates used for higher leveled applications and services. Our approach 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

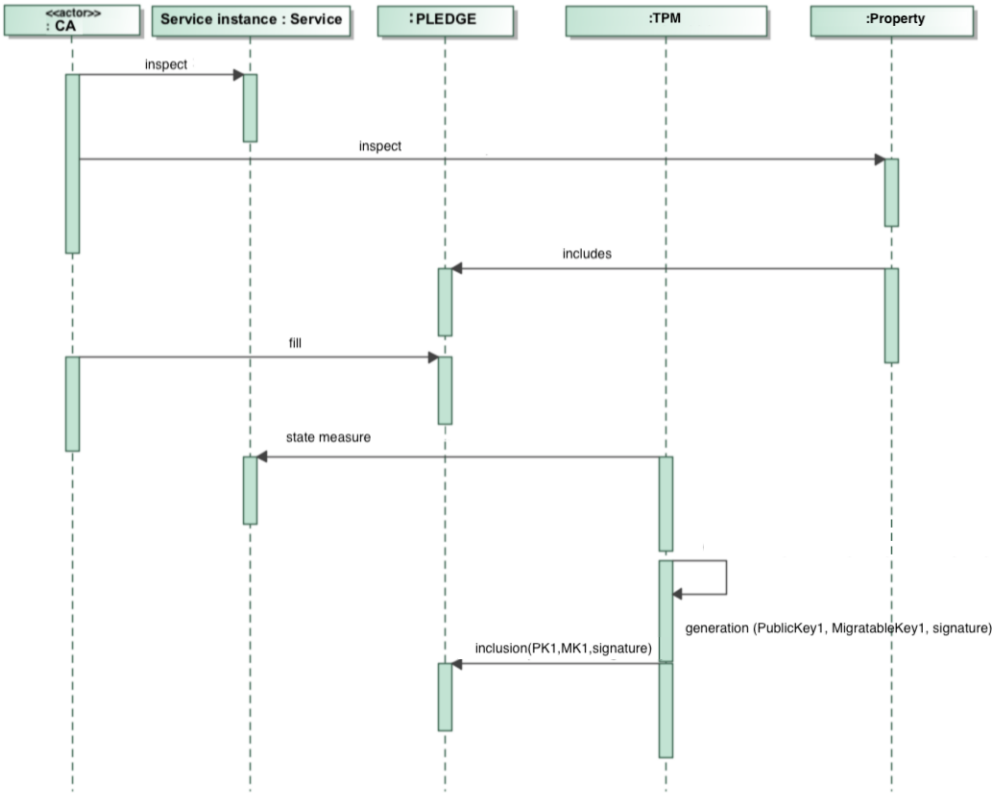

Fig. 4. Cloud Certification Sequence Diagram

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.

A list of implementation hints is following given:

—An implementation of the proposed certification approach recommends the use of TPM2 that includes support for additional cryptographic algorithms, enhancements to the availability of the TPM to applications, enhanced authorization mechanisms, simplified TPM management and some additional capabilities to enhance the security of platform services among others.

—Service security properties should be described using a formalism to be inspected by the certification authority in a automated procedure.

—Testing Configuration class included in diagram from figure 2.4.1 includes TPM information (TPM device-owner configuration) and information needed for instantiating related service.

—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. Currently there are not guidelines about how to implement and handle pledges, but ongoing work will provide some recommendations.

—We make use of a certification mechanism based on the underlying remote attestation to achieve a semantic attestation model appropriate for cloud computing to achieve a semantic remote attestation[HaldarHaldar2004].

—Support for relating certifications with trusted computing proofs in order to provide a comprehensive trust chain covering the full cloud stack should be provided. Certification on TC platforms, where certificates of the higher levels (e.g., services) should include conditions on the low levels based on TC, will provide scope for exploiting TC in complex execution tasks and provide a basis for completing the trust chain.

—Existing certification solutions do not provide reliable means for assessing the trustworthiness of a composite application, since certificates are intended for human use. Additionally, the lack machine readable format, lack of explicit and precise formulation of security properties, highlighted that it cannot be used for runtime security assessment. TC Proofs, extracted from TPM execution, are used to certify the lower layers (hardware, native OS and software infrastructure). We recommend a model based on three levels of certification, to know the Service/Application, Platform and TC Proof. Our definition of binding corresponds to any means used to guarantee that a pledge corresponds to a service in the cloud. Then software certificate will be used for the higher levels (applications, services, software infrastructure).

—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 [A. MuñozA. Muñoz2011, Muñoz A.Muñoz A.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 the service's 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 [S. BergerS. Berger2006]).

2.6 Consequences:

The proposed pattern provides a number of advantages, which are described below.

—A service certification approach to cloud computing that avoids the recurring need for human interaction in traditional certification models, implying a significant improvement in the usability of certification models for numerous use cases of daily interaction with certification-based systems.

—The existence of certified services allows for improved flexibility. Since it allows systems to be built considering the efficiency of the system as opposed to other traditional certification models that required human interaction with the restrictions of use that this implies.

—The use of certificates grants data and service confidentiality and data an service integrity as a consequence it increases security level.

Liabilities include:

—The use of this pattern is restricted by the indispensable requirement of having a TPM hardware device available in the system.

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

2.7 Known Uses:

In this work we achieve a proposal that includes a certification stack, with the definition of the related systems engineering and development process itself. Figure 1 shows that certification stack versus the infrastructure stack of a service. The figure 2 represents the proposed certification approach, which includes the steps that make up the process such as the analysis of the software to be certified, the identification and specification of the runtime tests and the generation of the certificate. The pledges are issued in such a way that they delegate the authorization by the certificate; a complete description of the definition of the pledges is not necessary to understand the operation of our models and a deeper description is beyond the scope of this document. Moreover, the process describing the use of pledges by clients is a work in progress.

2.8 Related Patterns

—Among the related patterns, those aimed at providing security for cloud environments include patterns of misuse for cloud computing such as the one presented in [K. HashizumeK. Hashizume2011].

—Cloud Resource Access Control pattern [ErlErl2015] is related to the security pattern proposed.

3. CONCLUSIONS & ONGOING WORK

This paper proposes a security pattern that provides a solution for the certification of cloud services. The solution makes use of secure elements in its design, in particular trusted hardware modules are used. The proposed scheme successfully bridges the gap between hardware-based certification and software certification. Combining the best of both worlds and overcoming their respective limitations results in a potentially useful solution. The concept of pledge as a form of IT-oriented certification is an essential key to improving the flexibility and practical applicability of TC mechanisms, as well as opening up possibilities for exploring future applications of trusted computing technology.

REFERENCES

R. J. Walters G. Wills A. Albugmi, M. O. Alassafi. 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

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

A. Maña A. Muñoz. 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

A. Maña D. Serrano B. Gallego, A. Muñoz. 2009. Security patterns, towards a further level. In *In Proceedings of the SECRYPT 2009*. SECRYPT INSTICC Press., 349–356.

Fernandez-Buglioni Eduardo. 2013. *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns* (1st ed.). Wiley Publishing.

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

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

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

Chandra Deepak Franz Michael Haldar, Vivek. 2004. Semantic Remote Attestation: A Virtual Machine Directed Approach to Trusted Computing. In *Proceedings of the 3rd Conference on Virtual Machine Research And Technology Symposium - Volume 3 (VM'04)*. USENIX Association, Berkeley, CA, USA, 3–3. http://dl.acm.org/citation.cfm?id=1267242.1267245

E.B. Fernandez K. Hashizume, N. Yoshioka. 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

Antón P. Muñoz A., Maña A. 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, 284–297.

K. Goldman R. Perez R. Sailer L. van Doorn S. Berger, R. Cáceres. 2006. vTPM: virtualizing the trusted platform module. In *In Proceedings of the 15th conference on USENIX Security Symposium*, Vol. 15. USENIX Association.