A Methodology for Management of Cloud Computing using Security Criteria

Carlos Alberto da Silva
Depto of Computer Systems
Institute of Computing - UNICAMP
Campinas, Brazil
beto@las.ic.unicamp.br

Anderson Soares Ferreira
Depto of Computer Systems
Institute of Computing - UNICAMP
Campinas, Brazil
anderson@las.ic.unicamp.br

Paulo Licio de Geus
Depto of Computer Systems
Institute of Computing - UNICAMP
Campinas, Brazil
paulo@las.ic.unicamp.br

***Abstract*—Desirable requirements of cloud computing are to avoid wasting underused resources and increasing response time due to shortage of resources. We notice that recent literature in the field prioritizes the administration of resource provisioning and the allocation algorithms for an energy-efficient management of cloud computing environments. Security metrics can be seen as tools for providing information about the security status of a certain environment. With that in mind, we tackle the management of cloud computing security by using GQM methodology to develop a cloud computing security metrics hierarchy. The main goal of the proposed hierarchy is to produce a security index that describes the security level accomplished by an evaluated cloud computing environment. In a second step, this security index is used to compute an allocation index that helps in setting management priorities with a security bias. We also present a methodology for cloud computing management using security as a criterion.**

I. Introduction

Cloud computing provides on-demand access to a pool of computational resources, e.g. network, storage, services. These resources can be promptly provided or released with little management effort, since the environment is dynamic and scalable [1].

As pointed in Lindner et al. [2], cloud computing can deliver a choice of computing infrastructure, software development and deployment platform, or web applications as services, made available to consumers in a pay-as-you-go model. In the industry these services are referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), respectively.

The operational model and the technology used to provide services in cloud computing environments have different levels of risk when compared with traditional information technology environments [3]. Despite the attractive benefits of this technology, there is a growing concern about the security of information and applications for cloud environments, making security issues the main obstacle to the adoption of cloud computing [4].

Due to its elastic characteristic, cloud computing allows for dynamic resource allocation/deallocation with no human intervention. In Andrew et al. [5], the criteria used for that were: i) maximizing the allocation of resources, avoiding underutilization; ii) minimize the response time, avoiding overuse of the resources; iii) minimize power consumption migrating VMs from one node to another.

In order to improve security in cloud environments, this article presents a methodology for cloud management based on security criteria obtained from security metrics and security agreements. It is organized as follows: Section II presents the theoretical aspects related to the topic and related work; Section III describes the proposed methodology for management of cloud computing using security criteria; Section IV discusses the allocation policy; and Section V presents conclusions and future work.

II. Theoretical Aspects

According to Landwehr [6], protecting a computer system requires establishing the sensitivity of the data manipulated by the application. Furthermore, the security of computer systems involves issues such as security policies and risk management. Security policies specify a set of rules for: protecting the physical level; disaster containment and recovery; backup management; media preservation and destruction; user training; event logging policy; use of cryptography and its parameters; system and resource access control; preventing violation of laws and ethics etc. Risk management involves the systematic and continuous evaluation of computer security levels, as well as system and application assessment to identify threats and vulnerabilities for further mitigation or correction [7].

A. Security Metrics

Security metrics are measurements from which to monitor and compare the level of security and privacy attained, as well as the current security status of a computing environment. The use of security metrics promotes transparency, decision making, predictability and proactive planning [8]. Metric is a measurement standard, defining both what is being measured (the attribute) and how it is measured (the unit of measure) [9].

Measurement is the process of metric collection which, through pre-established rules, will allow the interpretation of results [9]. Metrics can be composed of sub-elements that are referred to as primitive metrics or sub-metrics. Any restrictions or controls relating to the primitives are defined in

the measurement process. A metric can be expressed in one of the following ways:

i) # - "number", expressing an absolute value of any element measured;
ii) % - "percentage", expressing a percentage of an element measured in relation to the total number of elements;
iii) "logic value", expressing *Yes* or *No* for an event.

B. Related Work

Security and privacy are among the most discussed topics when migrating information from traditional systems to cloud computing.

In Foster et al.[10], the authors present the basic characteristics of cloud computing and make a comparison between computational grids and clouds, through the analysis of aspects of their architectures, business models, management and security. The work also presents the main problems of cloud computing, including the lack of standardization among cloud solutions.

New methodologies that describe the integration of security policies with Security-SLA are presented by Buyya et al. [11], Hayden [8], Lamin et al. [12] and Righi et al. [13].

A formal specification model for abstract security properties is presented by Mana et al. [14], and a formal approach to specification and rigorous analysis of security metrics is presented by Krautsevich et al. [15].

A review of the methodology to describe multilevel security policies through a tool to measure the quality of protection (QoP) for the information flow and the risks involved in the problem of multilevel security in computer networks is presented by Foley et al. [16].

A comparative analysis model and taxonomy of security metrics are presented by Savola [17], [18], [19].

In Halonen et al. [20], the authors presented an overview of how to manage security in complex systems such as cloud computing, focusing on the technical aspects of security and comparing the various taxonomies of security metrics.

The security criteria that should be present in a tool for security management for cloud computing is presented by Halonen et al. [21].

Studies like Zhang et al. [22], Younge et al. [5], Yazr et al. [23] present solutions to the problem of allocation of computing resources of the clouds based on a number of criteria: maximal use of resources, minimizing the response time for the user or reduction of power consumption. This problem is generally defined as a knapsack problem [24], or a specific variant called Vector Bin Packing [25]; both problems are known to be NP-hard problems.

As pointed in Arshad et al. [26], the scheduler can prevent random migration of a virtual machine to a less secure host than the current one using the QoS requirements of the contract security SLA in the evaluation of candidate nodes for the migration process.

III. Management of Cloud Computing

Figure 1 represents the proposed life cycle of security management for cloud computing environments.

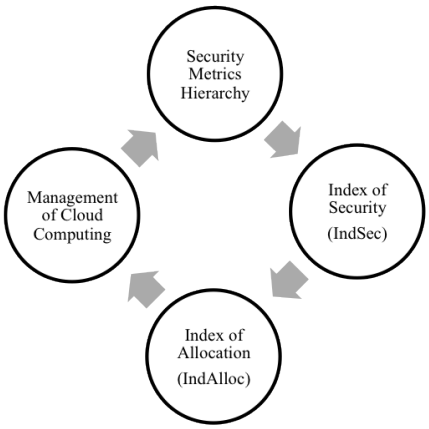

Fig. 1. Life cycle of Security Management

The proposed methodology for security management in cloud computing is based on the following components: i) security metrics hierarchy; ii) security index (IndSec); iii) allocation index (IndAlloc); iv) management of cloud computing.

A security metrics hierarchy is derived from the GQM methodology. A security index ($IndSec$) will be computed using the security metrics hierarchy, which in turn allows for the calculation of the allocation index ($IndAlloc$). Finally, the cloud management scheduler will use the allocation index as a reference for the the resource allocation process.

In the context of the life cycle of security management (Fig. 1), a security metrics hierarchy is presented as a new form of visualization of security-related information that is collected from the cloud computing environment.

A. GQM Methodology

In the 1970s, the GQM method (Goal-Question-Metric) [27] was designed to move testing for software defects from the qualitative and subjective state it was currently in to an empirical model, in which defects would be measured against defined goals and objectives that could then be linked to results.

The GQM methodology defines a measurement model on three levels: i) Conceptual level (goal) - a goal is defined for an object for a variety of reasons, with respect to various models of quality, from several points of view and relative to a particular environment; ii) Operational level (question) - a set of questions is used to define models of the object under study and then attention is focused on that object to characterize the assessment or achievement of a specific goal; iii) Quantitative level (metric) - a set of metrics, based on the models, is associated with every question in order to answer it in a measurable way.

In our methodology, the security metrics hierarchy is generated directly from the GQM definition process, during which stage security features are mapped to corresponding security metrics. Table I shows the relationship between the GQM methodology and the security metrics hierarchy (SMH).

TABLE I
RELATIONSHIP BETWEEN THE GQM METHODOLOGY AND SMH

GQM Levels	SMH Levels
Conceptual level	Group Metric
Operational level	Metric
Quantitative level	Sub-Metric

For each goal statement identified in the conceptual level, a group metric will be defined. The operational level identifies which objects or activities must be observed or collected to measure the individual components of the goal statement. Lastly, the quantitative level defines which metrics remains explicitly aligned with the higher level goal statement.

B. Security Metrics Hierarchy

The security metrics hierarchy (Fig. 2) is derived from the GQM methodology. Its components are: i) Security Index ($IndSec$); ii) Group Metrics (Met_i); iii) Metrics ($Met_{i.j}$); iv) Sub-Metrics ($Met_{i.j.k}$).

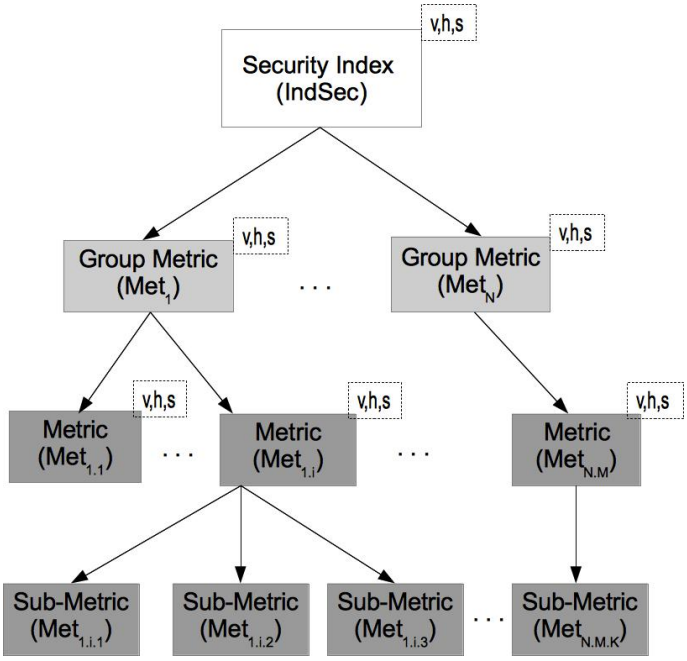

Fig. 2. Security Metrics Hierarchy

The Security Index (IndSec^{v,h,s}) is defined as the highest value in a set of security items:

$$IndSec^{v,h,s} = \max(Met_1^{v,h,s}, Met_2^{v,h,s}, ..., Met_N^{v,h,s})$$

The value of a metric group ($Met_i^{v,h,s}$) is defined as the highest value from a set of metrics:

$$Met_i^{v,h,s} = \max(Met_{i.1}^{v,h,s}, Met_{i.2}^{v,h,s}, ..., Met_{i.M}^{v,h,s})$$

The value of a metric ($Met_{i.j}^{v,h,s}$) is defined as the highest value from a set of sub-metrics:

$$Met_{i.j}^{v,h,s} = \max(Met_{i.j.1}^{v,h,s}, Met_{i.j.2}^{v,h,s}, ..., Met_{i.j.K}^{v,h,s})$$

The sub-metric represents a sub-part of a metric; it is used when a metric can be specialized in several ways, with each one having a different contribution to the overall metric. An example of the metric hierarchy is shown in Table II; the values in the column *Type* are: G = Group Metric, M = Metric and S = Sub-metric.

TABLE II
EXAMPLE OF A SECURITY METRICS HIERARCHY

Level	Description	Type	Metric
1	**Security Policy**	G	Met_1
1.1	Current Level of Enforcement of the Security Policy	M	$Met_{1.1}$
1.1.1	Number of reported security policy violations in the previous 12 months	S	$Met_{1.1.1}$
1.1.2	Number of enforcement actions taken against policy violations in the previous 12 months	S	$Met_{1.1.2}$
1.2	Current Structure of the Security Policy	M	$Met_{1.2}$
1.2.1	Number of documents that make up the corporate security policy	S	$Met_{1.2.1}$
1.2.2	Format(s) of security policy documents	S	$Met_{1.2.2}$
1.2.3	Location(s) of security policy documents (content management system, static web page, three-ring binder)	S	$Met_{1.2.3}$
1.2.4	Types of policy acknowledgement mechanisms (e-mail notification of users, electronic acknowledgement of policy access or review, hard copy signoff sheet)	S	$Met_{1.2.4}$
1.2.5	Length of time since the last security policy review by management	S	$Met_{1.2.5}$
2	**Security-Related Downtime**	G	Met_2
2.1	System down (failure)	M	$Met_{2.1}$
2.1.1	Time between failures	S	$Met_{2.1.1}$
2.1.2	Failure duration	S	$Met_{2.1.2}$
2.1.3	Mean system availability	S	$Met_{2.1.3}$
2.2	System down (maintenance)	M	$Met_{2.2}$
2.2.1	Time between maintenance	S	$Met_{2.2.1}$
2.2.2	Maintenance duration	S	$Met_{2.2.2}$
2.2.3	Mean system availability	S	$Met_{2.2.3}$
2.3	Downtime resulting from a security event	M	$Met_{2.3}$
2.3.1	Number of security events in a time period, Duration of event remediation	S	$Met_{2.3.1}$
3	**Vulnerabilities found on hosts**	G	Met_3
3.1	Total number of hosts scanned	M	$Met_{3.1}$
3.1.1	Total number of registered hosts	S	$Met_{3.1.1}$
3.1.2	Total number of unregistered hosts	S	$Met_{3.1.1}$
3.2	Total number of hosts vulnerable	M	$Met_{3.2}$
3.2.1	Total number of registered hosts vulnerable	S	$Met_{3.2.1}$
3.2.2	Total number of unregistered hosts vulnerable	S	$Met_{3.2.2}$
4	**Internal Vulnerability Assessment**	G	Met_4
4.1	Vulnerabilities on the internal servers	M	$Met_{4.1}$
4.1.1	Security vulnerability counts for assessed internal servers (from scanning)	S	$Met_{4.1.1}$
4.1.2	Ratios of vulnerabilities by type, OS, owner, and so on	S	$Met_{4.1.2}$
4.2	Severe vulnerabilities found on the internal servers	M	$Met_{4.2}$
4.2.2	CVSS scores for all identified vulnerabilities present on internal servers	S	$Met_{4.2.1}$

The use of the function max at each level of hierarchy causes the largest measured metric value to be passed on to the level immediately above, i.e. the highest measured value will be the only significant one.

The nomenclature for the security metrics is $Met_{i.j.k}^{v,h,s}$, measured in the cloud computing environment for the the

virtual machine (v) on host (h) and service (s), for layers IaaS, PaaS and SaaS. The reference $i.j.k$ identifies the location of the metric in the hierarchy, where: i refers to the security item, j refers to the group metric, and k refers to the primitive metric. The representation of a security metric in the hierarchy is described as follow: A value $Met_{i.j.k}^{v,h,s}$ is a measurement for the user (u), virtual machine (v) on host (h) and service (s), for the group metric i, metric j and sub-metric k.

C. Conversion of Security Metrics

The motivation behind value conversion is: i) to extract a meaning for the values measured by the primitive metrics; ii) to allow sorting them by their absolute value; iii) to prevent the value domains of security metrics from having instances that are difficult to be compared with each other, and to simplify the computational model using a method to converge the values of each primitive metric measured to a common scale of values.

On the proposed scale, the highest value (4) represents a security level less reliable and/or presenting a serious security problem. The lowest value (0) represents a security level that is safer and/or that does not present any security issue.

1) Logical metric: A metric of type *logic* must return a logical value measured from an event, e.g. "Anti-virus installed?". After the conversion (Tab. III) one gets:

TABLE III
CONVERSION OF LOGICAL METRIC

Index	logical value (x)
0	Yes
1	
2	
3	
4	No

The conversion function is described as $y = f(x)$, where x can be a measured logic value *Yes* or *No*:

$$y = \begin{cases} 0 & \text{if x } = \textit{Yes} \\ 4 & \text{if x } = \textit{No} \end{cases}$$

2) Numerical metric: It is a metric that returns a numerical value representing an event, e.g. "number of security patches not installed". After the conversion (Tab. IV) one gets:

TABLE IV
CONVERSION OF NUMERICAL METRIC

Index	# - Numeric Metric (x)
0	$]\ -\infty\ ,\ 0\ ]$
1	$]\ 0\ ,\ a_1\ ]$
2	$]\ a_1\ ,\ b_1\ ]$
3	$]\ b_1\ ,\ c_1\ ]$
4	$]\ c_1\ ,\ +\infty\ [$

With respect to the start and end boundaries of each interval, we have: $0 < a_1 \leq b_1 \leq c_1 < +\infty$

The conversion function is described as $y = f(x)$, where x is the value measured by the numerical metric:

$$y = \begin{cases} 0 & \text{if x } \in\ ]\ -\infty\ ,\ 0\ ] \\ 1 & \text{if x } \in\ ]\ 0\ ,\ a_1\ ] \\ 2 & \text{if x } \in\ ]\ a_1\ ,\ b_1\ ] \\ 3 & \text{if x } \in\ ]\ b_1\ ,\ c_1\ ] \\ 4 & \text{if x } \in\ ]\ c_1\ ,\ +\infty\ [\end{cases}$$

IV. ALLOCATION POLICY

The allocation index ($IndAlloc$) is calculated from the security index, and its value represents the resource allocation percentage that will be supplied to the user:

$$IndAlloc = \left(1 - \frac{IndSec}{5}\right) * 100$$

Table V shows the resource allocation percentage calculated as a function of the security index.

TABLE V
RESOURCE ALLOCATION USING THE SECURITY INDEX

Allocation Index table			
$IndSec^{v,h,s}$	$IndAlloc^{v}$	**Priority**	**Impact**
0	100 %	maximal	zero
1	80 %	high	minimal
2	60 %	medium	medium
3	40 %	low	high
4	20 %	minimal	critical

In the next two sections we describe two alternative strategies to be chosen by the user for the implementation of the allocation index.

A. Strategy A: Apply in All

In the Apply in All (AA) strategy, the physical structure of the Cloud Computing is seen as a single logical unit for resource allocation management. Therefore, this results in an allocation index ranging from 20% to 100% of the original allocation factor. Figure 3 depicts this strategy.

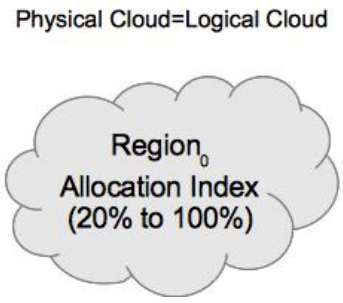

Fig. 3. Computing the Allocation Index through *Apply in All* (AA).

As an example of this strategy, a client who has a security index that equals 2 will be allocated 60% of his/her resource requests.

B. Strategy B: Apply in Regions

In the Apply in Regions strategy (AR), the physical structure of the Cloud Computing is divided into five logical regions that represent the calculated security levels (0, 1, 2, 3 and 4), where $Region_i \Leftrightarrow IndSec = i$. Each region will provide a specific resource percentage, as shown in Table V. Figure 4 shows this strategy, which may be exemplified by, say, a client who owns a security index that equals 2, that will be located in $Region_2$ and allocated 60% of his/her resource requests.

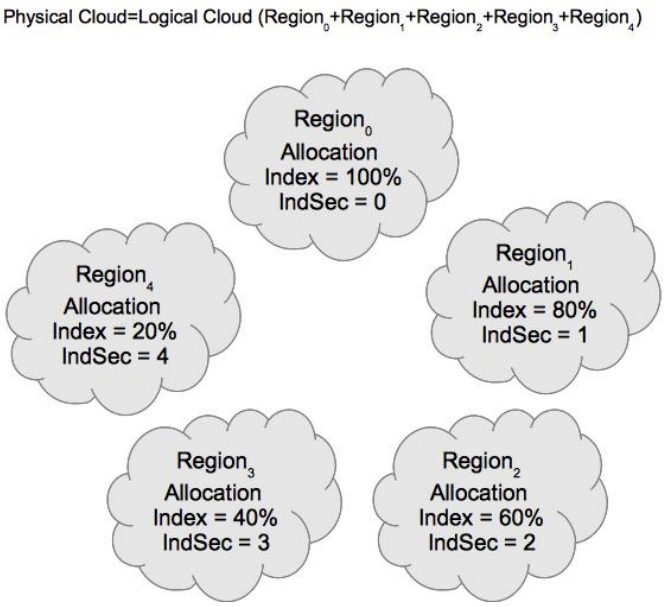

Fig. 4. Computing the Allocation Index through *Apply in Regions* (AR).

C. Strategy Comparison

Strategy A has a lower cost to create and maintain (single region), therefore presents a lower security level, since all clients continue to share the same infrastructure. A security incident on a client's domain could have consequences on other client's. In contrast, strategy B has a higher creation and maintenance cost but due to its intrinsic confinement guarantees, to a certain degree, a more desirable security level among different client services. Figure 5 shows an estimate of this comparison.

V. Conclusion and Future Work

In this article we proposed a methodology for management of cloud computing using security criteria. We presented two strategies for resource management that addresses scalability and granularity in cloud computing. The security index ($IndSec$) transparently conveys the security level measured in the cloud computing environment for the various security features modeled in the metrics hierarchy. Moreover, this approach has the advantage of supporting hierarchical decomposition, which allows the model to be more scalable and distributed.

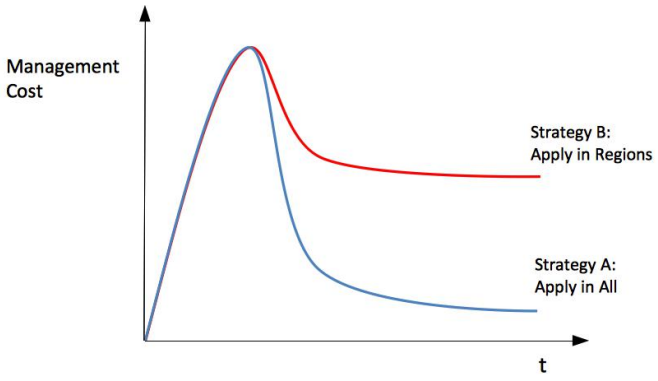

Fig. 5. Strategy Comparison

As for future work, we currently use security metrics that can be measured automatically from the environment, but the process still requires experts to set up limiting values for the ranges, which means that our model is highly dependent on human intervention. Another formulation for calculating the security index can be obtained by combining a weight value for each metric, where each weight value represents the degree of importance among metrics toward composing the metrics set. The security metrics at an upper level could be calculated as a weighted average of the metrics of the level immediately below it. Also, we plan to extend the comparison of strategies for management of cloud computing that were presented (AA and AR), in relation to overhead and performance, for a preliminary set of 180 metrics derived from accepted GQM methodology.

Acknowledgment

References

[1] P. Mell and T. Grance. (2011, May) The nist definition of cloud computing. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdflastaccessin28/11/2011.

[2] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, January 2009.

[3] CSA. (2009) Security guidance for critical areas of focus in cloud computing, cloud security alliance. V2.1. Cloud Security Alliance. [Online]. Available: http://www.cloudsecurityalliance.org/guidance/csaguide.v2.1.pdf

[4] G. B., W. T., and S. E., "Understanding cloud computing vulnerabilities," *IEEE Security and Privacy*, 2010.

[5] A. J. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for cloud computing environments," *IEEE*, no. 978-1-4244-7614-5/10, 2010.

[6] C. Landwehr, "Computer security," *International Journal of Information Security*, vol. 1, pp. 3–13, 2001.

[7] ISO/IEC. (2009) Iso/iec 27000 - information technology - security techniques - information security management systems - requirements. international organization for standartization.

[8] H. L., *IT Security Metrics: A Practical Framework for Measuring Security & Protecting Data*. McGraw-Hill Osborne, 2010.

[9] D. S. Herrmann, *Complete guide to security and privacy metrics*. Auerbach Publications, 2007, iSBN: 0-8493-5402-1.

[10] I. Foster, Y. Zhao, I. Raicu, and L. Shiyong, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008*, ser. GCE '08, November 2008, pp. 1–10.

[11] R. Buyya, S. K. Garg, and R. N. Calheiros, "Sla-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions," *Proceedings of the 2011 IEEE, International Conference on Cloud and Service Computing (CSC 2011, IEEE Press, USA)*, 2011.

[12] F. R. Lamin, C. B. Westphall, and S. A. de Chaves, "Sla perspective in security management for cloud computing," in *Sixth International Conference on Networking and Services*, ser. ICNS'10. IEEE Computer Society, March 2010, pp. 212–217.

[13] R. R. Righi, F. R. Pellissari, and C. B. Westphall, "Sec-sla: specification and validation of metrics for service level agreements oriented to security," in *IV Workshop in Computer Security System*. Porto Alegre, RS: SBC, 2004, pp. 199–210.

[14] A. Mana and G. Pujol, "Towards formal specification of abstract security properties," *The Third International Conference on Availability, Reliability and Security (ARES 08) - IEEE*, pp. 80–87, 2008.

[15] L. Krautsevich, F. Martinelli, and A. Yautsiukhin, "Formal approach to security metrics.: what does "more secure" mean for you?" *Proceedings of the Fourth European Conference on Software Architecture - ECSA '10*, vol. Companion Volume, pp. 162–169, 2010.

[16] S. N. Foley, S. Bistarelli, B. O'Sullivan, J. Herbert, and G. Swart, "Multi-level security and quality of protection," in *QUALITY OF PROTECTION - Security Measurements and Metrics*, ser. Part 3, vol. 23. Springer US, 2006, pp. 93–105.

[17] R. M. Savola, "Towards a security metrics taxonomy for the information and communication technology industry," *International Conference on Software Engineering Advances (ICSEA) - IEEE*, p. 60, 2007.

[18] ——, "Towards a taxonomy for information security metrics," *Proceeding QoP '07 Proceedings of the 2007 ACM workshop on Quality of protection - ACM*, pp. 28–30, 2007.

[19] ——, "A security metrics taxonomization model for software-intensive systems," *Journal of Information Processing Systems*, vol. 5, p. 197, 2009.

[20] P. Halonen and K. Hatonen, "Towards holistic security management through coherent measuring," 2010, pp. 155–161.

[21] O. Rebollo, D. Mellado, and E. Fernndez-Medina, "A systematic review of information security governance frameworks in the cloud computing environment," *Journal of Universal Computer Science*, vol. 18, pp. 798–815, 2012.

[22] Y. Zhang, G. Huang, X. Liu, and H. Mei, "Integrating resource consumption and allocation for infrastructure resources on-demand," *IEEE 3rd International Conference on Cloud Computing*, 2010.

[23] Y. O. Yazr, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, and Y. Coady, "Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis," *3rd International Conference on Cloud Computing - IEEE*, no. 978-0-7695-4130-3/10, pp. 91–98, 2010.

[24] W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das, "Utility functions in autonomic systems in icac 04," *Proceedings of the first international conference on autonomic computing, IEEE Computer Society*, p. 7077, 2004.

[25] F. Hermenier, X. Lorca, J. M. Menaud, G. Muller, and J. Lawall, "Entropy: A consolidation manager for clusters, in vee09: Proceedings of the 2009," *ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, p. 4150, 2009.

[26] J. Arshad, P. Townend, and J. Xu, "Quantification of security for compute intensive workloads in clouds," *15th International Conference on Parallel and Distributed Systems - IEEE*, 2009.

[27] V. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric approach," 1994.