UNIVERSITÉ DE MONTRÉAL

SECURITY IN CLOUD COMPUTING: EVALUATION AND INTEGRATION

TALAL HALABI
DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
AOÛT 2018

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

SECURITY IN CLOUD COMPUTING: EVALUATION AND INTEGRATION

présentée par : HALABI Talal
en vue de l'obtention du diplôme de : Philosophiæ Doctor
a été dûment acceptée par le jury d'examen constitué de :

M. PIERRE Samuel, Ph. D., président
Mme BELLAÏCHE Martine, Ph. D., membre et directrice de recherche
M. QUINTERO Alejandro, Doctorat, membre
M. GRÉGOIRE Jean-Charles, Doctorat, membre externe

DEDICATION

To my parents,
I couldn't have done it without your love and support...

To my brothers and sisters,
for believing in me...

To the memory of my grandmother,
you taught me to reach for the stars,
I wish you were here today to see me touching them...
...I miss you deeply.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my research adviser Professor Martine Bellaïche for her persistent support, encouragement, and patience during my Ph.D. studies. Thank you for the confidence you have placed in me throughout the last four years and for the countless things you taught me in research, as well as in real life.

I would also like to thank Professor Samuel Pierre, Professor Alejandro Quintero, and Professor Jean-Charles Grégoire for showing interest in my work and accepting to participate in this jury.

I am grateful to Professor Michel Dagenais for his financial support during the last year.

I am especially thankful to my fellow colleagues for the stimulating and productive discussions. In particular, I thank my dear colleague and friend Adel Abusitta for having created an inspiring and dynamic research environment. The lab would have been boring without you.

My thanks also go to my dear friends who have shared with me my ups and downs and were very patient towards my research-driven mood changes. Thank you Alexandre, Rym, Lolwa, and Mahmoud for always being there for me.

Last but not least, I would like to thank my family : my parents, brothers and sisters for continuously supporting me overseas throughout my challenging journey. Thank you for believing in me.

RÉSUMÉ

Au cours de la dernière décennie, le paradigme du Cloud Computing a révolutionné la manière dont nous percevons les services de la Technologie de l'Information (TI). Celui-ci nous a donné l'opportunité de répondre à la demande constamment croissante liée aux besoins informatiques des usagers en introduisant la notion d'externalisation des services et des données. Les consommateurs du Cloud ont généralement accès, sur demande, à un large éventail bien réparti d'infrastructures de TI offrant une pléthore de services. Ils sont à même de configurer dynamiquement les ressources du Cloud en fonction des exigences de leurs applications, sans toutefois devenir partie intégrante de l'infrastructure du Cloud. Cela leur permet d'atteindre un degré optimal d'utilisation des ressources tout en réduisant leurs coûts d'investissement en TI. Toutefois, la migration des services au Cloud intensifie malgré elle les menaces existantes à la sécurité des TI et en crée de nouvelles qui sont intrinsèques à l'architecture du Cloud Computing. C'est pourquoi il existe un réel besoin d'évaluation des risques liés à la sécurité du Cloud durant le procédé de la sélection et du déploiement des services. Au cours des dernières années, l'impact d'une efficace gestion de la satisfaction des besoins en sécurité des services a été pris avec un sérieux croissant de la part des fournisseurs et des consommateurs. Toutefois, l'intégration réussie de l'élément de sécurité dans les opérations de la gestion des ressources du Cloud ne requiert pas seulement une recherche méthodique, mais aussi une modélisation méticuleuse des exigences du Cloud en termes de sécurité.

C'est en considérant ces facteurs que nous adressons dans cette thèse les défis liés à l'évaluation de la sécurité et à son intégration dans les environnements indépendants et interconnectés du Cloud Computing. D'une part, nous sommes motivés à offrir aux consommateurs du Cloud un ensemble de méthodes qui leur permettront d'optimiser la sécurité de leurs services et, d'autre part, nous offrons aux fournisseurs un éventail de stratégies qui leur permettront de mieux sécuriser leurs services d'hébergements du Cloud. L'originalité de cette thèse porte sur deux aspects : 1) la description innovatrice des exigences des applications du Cloud relativement à la sécurité ; et 2) la conception de modèles mathématiques rigoureux qui intègrent le facteur de sécurité dans les problèmes traditionnels du déploiement des applications, d'approvisionnement des ressources et de la gestion de la charge de travail au cœur des infrastructures actuelles du Cloud Computing. Le travail au sein de cette thèse est réalisé en trois phases.

Dans le premier volet de notre travail, nous proposons une méthodologie systématique de l'évaluation de la sécurité intégrant un ensemble de métriques qualitatives et quantitatives pour décrire le niveau de sécurité des systèmes du Cloud Computing en se basant sur une

analyse rigoureuse de leurs exigences. Cette évaluation permet l'examen relatif du statut de sécurité des fournisseurs de services et leur permet de se situer dans le marché du Cloud en termes de niveau de sécurité. De plus, celle-ci peut jouer le rôle d'une feuille de route que les fournisseurs peuvent exploiter afin d'améliorer la sécurité de leurs services. L'ensemble des métriques de sécurité est en outre élaboré pour définir les Security Service Level objectives (SSLO), que nous incorporons dans la conception d'un Security Service Level Agreement (Security-SLA). Le Security-SLA désigné met en lumière la représentation quantitative des offres de sécurité dans le Cloud, ainsi que les besoins des clients en matière de sécurité. Pour permettre une évaluation pratique et réaliste des Security-SLAs du Cloud, nous proposons une approche de sélection des services reposant sur l'importance relative des besoins de sécurité des clients en termes des trois attributs du modèle de sécurité « CIA triad » : la confidentialité, l'intégrité et la disponibilité.

Dans la deuxième partie du travail, nous mettons l'emphase sur l'importance de la satisfaction quant à la sécurité au sein des processus de placement des services et de leur fédération dans le Cloud Computing. Pour supporter le déploiement sécuritaire des applications du Cloud, nous concevons un modèle d'optimisation multiobjectif de la sécurité afin de composer des services sécurisés dans un environnement Multi-Cloud au niveau de granularité des serveurs. Le modèle fait état du degré de satisfaction quant à la sécurité comme étant l'élément primaire dans le processus d'approvisionnement des services dans le Cloud et permet aux clients de délibérer avec succès du compromis entre la sécurité et la performance. Le problème est résolu de manière optimale grâce à l'utilisation d'un solveur mathématique, et sa pertinence quant à la satisfaction des besoins est étudiée en termes de la violation des Security-SLAs. En général, les fournisseurs de services fédèrent l'exécution de leur charge de travail entre eux avec pour objectif d'améliorer la Qualité de Service (QdS). Toutefois, dans leurs activités de fédération, ceux-ci prennent rarement en compte la sécurité des environnements hôtes et les conséquences sérieuses que cette fédération peut apporter à la sécurité de la charge de travail. Pour permettre aux fournisseurs d'intégrer l'élément de sécurité dans le processus de fédération, nous concevons deux cadres différents de fédération sécuritaires en ligne. Dans le premier, le risque de sécurité des infrastructures du Cloud Computing est jaugé selon un ensemble de critères et incorporé dans un modèle de jeux coopératif dirigeant le processus de formation de la fédération. Dans cette approche, la réputation des fournisseurs, estimée selon la perception de leurs services par les clients, joue un rôle fondamental dans l'évaluation de leurs préférences lorsqu'ils rejoignent les fédérations. Le modèle de jeux est analysé au regard des propriétés de stabilité définies dans la littérature, et son habilité à assurer une fédération sécurisée des charges de travail sur demande est démontrée. Dans la seconde approche, nous évaluons le niveau de sécurité des fournisseurs par rapport à un

niveau de référence des besoins de sécurité qui est défini en se basant sur le Security-SLA. Le cadre adopte un modèle de jeux hédonique coalitionnel dans lequel les fournisseurs spécifient leurs préférences en termes de niveau de sécurité des fédérations formées. Le modèle proposé démontre l'efficience computationnelle, et prouve son habilité à réduire le taux et la sévérité des violations du Security-SLA dans le processus de fédération entre les centres de données du Cloud Computing. En y combinant les exigences de QdS et une analyse adéquate de profitabilité, l'approche peut être mise en place sur les infrastructures actuelles du Cloud Computing afin d'assurer une fédération plus sécurisée.

Au cours de la dernière étape de notre thèse, nous abordons les défis liés à l'intégration de la sécurité dans l'approvisionnement et de l'allocation des ressources dans le Cloud. Premièrement, pour encourager les fournisseurs de services à réduire le risque de sécurité de leurs infrastructures, nous concevons un modèle d'allocation des ressources basé sur l'évaluation du risque de sécurité des consommateurs à l'échelle d'un InterCloud. Ce risque est évalué selon le niveau de sécurité déclaré dans les requêtes d'approvisionnement des ressources. L'objectif de ce modèle est d'optimiser l'utilisation des ressources, tout en gardant le niveau du risque de sécurité des infrastructures du Cloud sous une valeur critique. Le problème d'allocation des ressources que nous définissons dans un contexte de sécurité est modélisé en tant que problème d'optimisation linéaire et résolu grâce à deux différentes approches métaheuristiques appartenant à la famille du calcul évolutionnaire : l'algorithme génétique ou Genetic Algorithm (GA) et l'Artificial Bee Colony (ABC). Ces deux approches démontrent leur habilité à accomplir une approximation acceptable de la solution optimale et à être implémentées en ligne. Ensuite, on se penche sur le problème d'approvisionnement des ressources en prenant en considération la satisfaction des demandes des requêtes en terme de sécurité. Afin de répondre efficacement aux exigences des clients en matière de sécurité durant le processus d'allocation des ressources en ligne, nous proposons un modèle dans lequel le Broker aspire à maximiser la satisfaction liée à la sécurité lors de l'allocation des demandes des clients sur les centres de données du Cloud. Le problème est résolu en utilisant l'algorithme génétique, et la performance du modèle est évaluée en termes de la qualité de la solution générée et du temps d'exécution. Finalement, nous abordons le problème d'intégration de la sécurité dans le Cloud d'un point de vue financier. Nous modélisons mathématiquement le processus d'allocation des ressources sécurisées dans un contexte d'enchère. Dans le modèle, on demande aux clients de divulguer leur appréciation de la sécurité des ressources du Cloud dans leurs requêtes d'approvisionnement des ressources. Pour résoudre ce problème d'optimisation linéaire tout en assurant la véracité des renseignements fournis par les soumissionnaires, nous proposons un mécanisme d'allocation en ligne appartenant à la famille des mécanismes « Dominant-Strategy Incentive-Compatible » (DSIC). Le mécanisme conçu aspire à maximi-

ser le bien être social (Social Welfare) des clients et à réduire le coût de l'investissement en sécurité en allouant les ressources aux clients qui valorisent le plus leur sécurité. Celui-ci instaure une procédure d'allocation basée sur l'algorithme gourmand et une règle de paiement favorisant l'honnêteté des clients. Il démontre son habileté à trouver en un temps polynomial des solutions proches de la solution optimale généralement réalisée à l'aide du mécanisme de Vickrey-Clarke-Groves (VCG) qui est normalement exécuté hors-ligne.

Les solutions présentées serviront de guide tant pour les utilisateurs du Cloud que pour les fournisseurs de services en misant sur la réalisation d'un service d'hébergement Cloud sécuritaire. Garder à l'esprit l'élément de sécurité au cours des opérations de sélection des services du Cloud, du déploiement des applications, de l'allocation des ressources et de la gestion de la mobilité de la charge de travail permettra aux services du Cloud de jouir d'un niveau de sécurité jusqu'ici difficilement atteignable, et rendra leur adoption entièrement réalisable.

ABSTRACT

Over the past decade, the Cloud Computing paradigm has revolutionized the way we envision IT services. It has provided an opportunity to respond to the ever increasing computing needs of the users by introducing the notion of service and data outsourcing. Cloud consumers usually have online and on-demand access to a large and distributed IT infrastructure providing a plethora of services. They can dynamically configure and scale the Cloud resources according to the requirements of their applications without becoming part of the Cloud infrastructure, which allows them to reduce their IT investment cost and achieve optimal resource utilization. However, the migration of services to the Cloud increases the vulnerability to existing IT security threats and creates new ones that are intrinsic to the Cloud Computing architecture, thus the need for a thorough assessment of Cloud security risks during the process of service selection and deployment. Recently, the impact of effective management of service security satisfaction has been taken with greater seriousness by the Cloud Service Providers (CSP) and stakeholders. Nevertheless, the successful integration of the security element into the Cloud resource management operations does not only require methodical research, but also necessitates the meticulous modeling of the Cloud security requirements.

To this end, we address throughout this thesis the challenges to security evaluation and integration in independent and interconnected Cloud Computing environments. We are interested in providing the Cloud consumers with a set of methods that allow them to optimize the security of their services and the CSPs with a set of strategies that enable them to provide security-aware Cloud-based service hosting. The originality of this thesis lies within two aspects: 1) the innovative description of the Cloud applications' security requirements, which paved the way for an effective quantification and evaluation of the security of Cloud infrastructures; and 2) the design of rigorous mathematical models that integrate the security factor into the traditional problems of application deployment, resource provisioning, and workload management within current Cloud Computing infrastructures. The work in this thesis is carried out in three phases.

In the first phase of our work, we propose a systematic security evaluation methodology that integrates a set of qualitative and quantitative metrics to describe the security level of a Cloud Computing system based on a rigorous analysis of the security requirements of Cloud services. This evaluation enables the relative assessment of the security status of CSPs and allows them to situate themselves in the Cloud market from a security perspective. It can also plays the role of a Cloud security roadmap which CSPs can exploit to improve the secu-

rity of their services. The set of security metrics is further elaborated to define the Security Service Level Objectives (SSLO) which we incorporate into the design of a standard Security Service Level Agreement (Security-SLA). The designed Security-SLA highlights the quantitative representation of Cloud security offerings and customers' security requirements. To enable a realistic and practical evaluation of the Cloud Security-SLA, we propose a service selection approach that relies on the relative significance of customer security requirements in terms of the three attributes of the CIA triad security model: confidentiality, integrity, and availability.

In the second phase, we emphasize the importance of security satisfaction in the processes of service placement and workload federation in Cloud Computing. To support the secure deployment of Cloud applications, we design a multi-objective security optimization model for service assignment in a Multi-Cloud setting at the granularity level of the server. The model exploits security satisfaction as the aspect of primary importance in the provisioning of Cloud services and enables customers to effectively deliberate over the trade-off between security and performance. The linear security-aware service placement optimization problem is solved optimally using a mathematical solver, and its relevance to service security satisfaction is studied in terms of Security-SLA violations. In general, CSPs usually federate the execution of their workload to each other with the objective of improving Quality of Service (QoS). However, in their federation activities, CSPs rarely take into account the security of Cloud hosting environments and the potentially dire consequences that this federation can bring to the security of the workload. To allow CSPs to integrate the security element into the federation process, we design two different security-aware online federation formation frameworks. In the first one, the security risk of Cloud Computing infrastructures is assessed according to a set of criteria and incorporated into a cooperative game model that drives the federation formation process. In this approach, the reputations of CSPs, which are estimated based on customers' feedback, play a fundamental role in the evaluation of CSPs' preferences while joining federations. The game model is analyzed with respect to the stability properties defined in the literature, and its ability to provide secure workload federation on the fly is demonstrated. In the second framework, we evaluate the security level of CSPs with respect to a security baseline that we define based on the Security-SLA. The framework adopts a hedonic coalitional game model in which CSPs specify their preference relationships in terms of the security level of formed federations. The proposed model shows computational efficiency, and proves its power in reducing the rate and severity of Security-SLA violations when federating service workload between Cloud Computing data centers. With further combination of QoS requirements and adequate pricing analysis, the framework can be implemented on current Cloud Computing infrastructures to ensure service protection during the process of

federation.

In the last phase of our work, we address the challenges to security integration at the level of resource provisioning and allocation in the Cloud. First, to support the CSPs in reducing the security risk on their infrastructures, we design a resource allocation model based on the assessment of customers' security risk which we evaluate in terms of the required security implementation specified in their resource provisioning requests. The objective of the model is to optimize resource utilization while keeping the security risk level of the Cloud infrastructure below the critical value. The security risk-aware resource allocation problem is modeled as a linear optimization problem and solved using two different metaheuristics approaches from the family of evolutionary computation: the Genetic Algorithm (GA) and Artificial Bee Colony (ABC). Both approaches demonstrate their ability to achieve an acceptable approximation of the optimal solution and to be implementable in online mode. Afterwards, the resource provisioning problem is tackled from a security satisfaction perspective. To effectively respond to customers' security requirements during the process of online resource allocation, we propose a broker-based model that aims at maximizing security satisfaction when allocating customers' requests to the Cloud data centers. The linear security-based resource allocation optimization problem is solved using the GA, and the performance of the model is evaluated in terms of solution quality and scalability. Finally, we approach the problem of security integration in the Cloud from a financial point of view: we mathematically model the process of secure resource allocation in an auction-based context. In the model, customers are asked to show their valuation of the security of the Cloud resources in their resource provisioning requests. To solve the linear optimization problem while ensuring the truthfulness of the bidders, we propose a Dominant-Strategy Incentive-Compatible (DSIC) online mechanism that aims at maximizing customers' social welfare and reducing the cost of security investment by allocating the secure resources to the customers who valuate their security the most. The mechanism implements a greedy-based allocation procedure and a truth-inducing payment rule. The mechanism is computationally efficient and achieves acceptable approximation of the optimal solution usually achieved through the offline Vickrey-Clarke-Groves (VCG) mechanism.

The proposed solutions will serve as a guide to both Cloud customers and service providers towards the achievement of secure Cloud-based service hosting. Bearing in mind the security element during the operations of Cloud service selection, application deployment, resource provisioning and allocation, and workload mobility management will elevate Cloud Computing services to a whole new level, at which their full adoption could finally be realizable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

IT	Information Technology
CSP	Cloud Service Provider
CIP	Cloud Infrastructure Provider
SaaS	Software-as-a-Service
PaaS	Platform-as-a-Service
IaaS	Infrastructure-as-a-Service
AWS	Amazon Web Service
EC2	Elastic Cloud Compute
VM	Virtual Machine
SLA	Service Level Agreement
Security-SLA	Security Service Level Agreement
CPU	Central Processing Unit
QoS	Quality of Service
CSA	Cloud Security Alliance
API	Application Programming Interface
DoS	Denial of Service
DDoS	Distributed Denial of Service
SLO	Service Level Objective
SSLO	Security Service Level Objective
EA	Evolutionary Algorithm
GA	Genetic Algorithm
ABC	Artificial Bee Colony
DSIC	Dominant Strategy Incentive-Compatible
IEEE	Institute of Electrical and Electronics Engineers
CSMIC	Cloud Services Measurement Initiative Consortium
SMI	Service Measurement Index
AHP	Analytical Hierarchy Process
MCDA	Multiple Criteria Decision Analysis
NIST	National Institute of Standards and Technology
GRC	Governance, Risk Management and Compliance
CCM	Cloud Controls Matrix
CAIQ	Consensus Assessment Initiative Questionnaire
ISO	International Organization for Standardization

IEC	International Electrotechnical Commission
ANP	Analytic Network Process
GQM	Goal, Question, Metric
HHM	Hierarchical Holographic Modeling
ENISA	European Network and Information Security Agency
MUSA	Multi-cloud Secure Applications
CMfg	Cloud Manufacturing
SCOS	Service Composition and Optimal Selection
VMPP	Virtual Machine Placement Problem
VCG	Vickrey–Clarke–Groves
MCA	Multi-Criteria Analysis
XSS	Cross Site Scripting
MITM	Man in the Middle
SQL	Structured Query Language
ROSI	Return On Security Investment
OS	Operating System
IAM	Identity and Access Management
SAML	Security Assertion Markup Language
SSO	Single Sign-On
LDAP	Lightweight Directory Access Protocol
OWASP	Open Web Application Security Project
URL	Uniform Resource Locator
SSL	Secure Sockets Layer
TLS	Transport Layer Security
PDP	Provable Data Possession
VPN	Virtual Private Network
SIEM	Security Incident and Event Management
SSO	Single Sign On
ACL	Access Control List
HTTP	HyperText Transfer Protocol
HSTS	HTTP Strict Transport Security
PKI	Public Key Infrastructure
SSH	Secure Shell
DLP	Data Loss Prevention
FRC	Fraudulent Resource Consumption
CV	Coefficient of Variation

EDoS	Economic Denial of Sustainability
OAuth	Open Authentication
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
XACML	eXtensible Access Control Markup Language
MOCILP	Multi-Objective Constrained Integer Linear Programming
VMPP	VM Placement Problem
MKP	Multiple-Knapsack Problem
BPP	Bin Packing problem
MCN	Maximum Cycle Number
CILP	Constrained Integer Linear Programming
TU	Transferable Utility
NTU	Non-Transferable Utility

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

The adoption of Cloud Computing technology continues to rise at an accelerated pace. Recent statistics show that the total worldwide Cloud IT infrastructure revenue has almost tripled in the last four years [1]. Many organizations have decided to shift their businesses to the Cloud to benefit from its attractive features and deliver an enhanced user experience to their customers. These benefits include higher availability, wider geographic reach, increased cost savings, and improved business continuity. Cloud Computing forms a new business model that enables on-demand access to an IT environment of shared, distributed, and scalable resources such as applications, servers, and storage nodes. It is based on the idea of resource democratization through the abstraction of IT infrastructures, where physical resources become decoupled from the delivered service or application. With the possibility of rapid provision and release of computing resources and highly flexible configuration of applications, the Cloud has created an opportunity for IT consumers to optimize the management of their workload and enjoy high performance services while reducing the cost of IT investment.

However, the migration of services and data to the Cloud empowers existing IT security vulnerabilities and introduces new ones that are specific to Cloud Computing such as consumers' loss of physical control over their data. These vulnerabilities create a fertile environment for potential security threats like data loss and Denial of Service (DoS). Due to its embedded functional and technical characteristics, Cloud Computing induces more security concerns than traditional IT infrastructures, thus the need for a thorough assessment of its associated risks and benefits during the migration process. The lack of effective management of security satisfaction in current Cloud infrastructures and limited transparency of Cloud Service Providers (CSP) with respect to their security implementation are among the main factors that are slowing the process of Cloud adoption, especially by the organizations that deal with sensitive information and provide highly reliable services. Security is considered to be a major driving factor of the Cloud market today, hence increasing security investments and emphasizing security satisfaction have apparently become indispensable to boost the migration towards the Cloud business model.

CSPs are in need for the right strategies to increase their public trustworthiness and improve their reputations. These strategies will not only involve the deployment of appropriate security measures within their data centers, but also the adoption of new security-oriented infrastructure management frameworks. These frameworks will be based on incorporating the security element into every aspect of Cloud management and operation, including applica-

tion deployment, resource provisioning and allocation, and workload mobility management, to enable security-aware Cloud-based service hosting. In this context, customers should be provided with a trustworthy platform that can assess the security of CSPs with respect to the security requirements of their applications, and grant them the possibility to perform an optimized security-driven Cloud service selection.

In this thesis, we address the challenges to security evaluation in Cloud Computing and tackle the problem of security satisfaction in independent and interconnected Cloud environments. We are interested in providing the Cloud consumers with a set of methods that allow them to optimize the security of their services and the CSPs with a set of solutions that enable them to perform security-aware service management on their infrastructures. In our research, the security element is investigated from multiple angles and for different purposes to enable the creation of rigorous optimization models and the design of plausible and practical solutions.

This introductory chapter is divided as follows. First, the basic concepts and technical terminology related to our research are defined and explained to allow for a better understanding of the foundations of our research problem. Then, the addressed problem is described and research objectives are defined. Afterwards, the originality of this thesis is emphasized through a detailed description of the main research contributions. Finally, the structure of the thesis is outlined.

1.1 Definitions and basic concepts

This section aims at defining the terminology and concepts that will be used in the rest of the thesis, which will help the reader in better grasping the context of our work. We start by introducing the Cloud Computing and describing its main characteristics and models. Then, the process of resource allocation in the Cloud is explained. Finally, a description of the principal terms related to the Cloud security environment is elaborated.

1.1.1 Cloud Computing

In the last decade, the Cloud Computing paradigm has transformed the way we think about computing. It has created an opportunity to respond to the ever increasing computing needs of the users by introducing the concept of service and data outsourcing. Cloud consumers usually have online and on-demand access to a large and distributed IT infrastructure providing a plethora of services according to a pay-per-use model. In this service-oriented model, consumers can dynamically configure and scale the Cloud resources according to the requirements of their applications without becoming part of the Cloud infrastructure. This allows

them to reduce their IT investment cost and achieve optimal resource utilization. Cloud Computing presents a resilient computing model that is based on large-scale distributed storage and parallel programming and execution of tasks (e.g., MapReduce framework [2]). It provides flexible workload management as an essential feature that promotes better performance and improved risk isolation. In the following, some important Cloud-related concepts are explained to smoothen readers' understanding of the rest of the thesis.

1.1.1.1 Service delivery models

In general, Cloud Computing services are provisioned according to three different models :

- Software-as-a-Service (SaaS), where software applications are provided to users over the internet in an on-demand fashion to eliminate the complexity of installing and maintaining the software. In this case, the service provider is in charge of managing and securing the software. Google Pack is an example of such services and includes a set of web accessible tools such as Gmail and Google Docs.
- Platform-as-a-Service (PaaS), where consumers have access to a platform of language-specific development tools (e.g., .Net) and an execution environment of operating systems and databases that permit them to deploy and manage their applications and websites without the need to install any software on their systems or to manage the underlying infrastructure. Examples of such services include the Google App Engine and the Elastic Beanstalk from Amazon Web Service (AWS).
- Infrastructure-as-a-Service (IaaS), where virtualized resources including servers, networking equipment, and storage systems are rented to consumers to allow them to flexibly manage their workload without the need to deploy a computing infrastructure on their premises. According to this model, users have the control over their Virtual Machines (VM) and service providers are responsible of operating and securing the underlying infrastructure. A popular service in this category is the Elastic Cloud Compute (EC2) provided by AWS [3].

SaaS and PaaS delivery models are usually deployed on top of the IaaS model, hence they are predisposed to inherit all its security vulnerabilities. In a multi-tiered Cloud architecture, SaaS and PaaS providers rent the IaaS resources to offer their Cloud services, hence performance and security responsibilities start to overlap and are usually specified through adequate Service Level Agreements (SLA).

1.1.1.2 Deployment models

The following deployment models are currently the most common in Cloud Computing :

— Public Cloud, which is owned and managed by the CSP, and provides online and on-demand access to a pool of IT resources (e.g., CPU power, data storage, etc) and web services to the public. Although this model provides the users with highly scalable and inexpensive computing resources, it introduces several risks including security and vendor lock-in.

— Private Cloud, which is usually owned by a private organization to provide Cloud services to its internal users. The infrastructure may exist outside the organization's premises and might be operated by a third party.

— Community Cloud, which is normally owned and managed by a group of organizations that share the same interests or have similar requirements. This model usually enjoys higher levels of security.

— Hybrid Cloud, which involves a combination of the previous three models. For instance, by deploying a hybrid Cloud, an organization can grasp the benefits of a public Cloud and continue to use its own data center to secure its sensitive data.

— Multi-Cloud, where an organization can use different Cloud services mostly offered by independent CSPs to achieve optimal performance and reduce the risk of dependence on one CSP. This model is becoming the most preferred strategy of Cloud deployment by organizations since it allows them to take advantage of each CSP's quality capabilities through a mix and match service delivery model according to their requirements [4].

Among these deployment models, public clouds usually introduce the highest security risk due to their distributed and shared resources. In this thesis, the security of public Cloud infrastructures is primarily highlighted to reflect the generality of our work.

1.1.2 InterCloud and federation

The Cloud Computing paradigm presents several concerns and challenges for CSPs, such as Quality of Service (QoS) guarantee, resource limitation, disaster-recovery planning, regional distribution of workload, and legal issues. To address these concerns, a CSP might choose to provide her services through the deployment of multiple data centers that are geographically distributed but securely interconnected. This concept is commonly known as InterCloud. Another model that usually helps CSPs in reducing resource provisioning costs and impro-

ving the performance of their services is Cloud federation. It allows a CSP to flexibly and transparently outsource a portion of her users' requests to other independent CSPs. Federation usually occurs along two different dimensions : horizontal, taking place at matching layers of the Cloud stack or for similar service delivery models (e.g., under the form of shared VM instances), and vertical, spanning multiple layers in order to service the additional requests on one specific layer through delegation [5]. By interconnecting their Cloud infrastructures and sharing their resources and capabilities, federated CSPs can optimize the management of their workload, maintain higher performance and QoS levels, and improve cost-effectiveness and energy efficiency. Moreover, their objectives can involve enhancing resilience against failures or unexpected situations and redundancy implementation by replicating computations and data among multiple infrastructures.

1.1.3 Virtualization

Virtualization is a key supporting technology in Cloud Computing, through which hardware resource abstraction is enabled. It allows multiple operating systems to exist on the same physical server and share its resources, under the control of the hypervisor, with the objective of improving hardware utilization and reducing the cost of IT investment. In an IaaS delivery model, virtualized resources are normally presented in the form of VMs having different computing and storage capabilities, on which users can run their software and applications.

1.1.4 Resource allocation in the Cloud

In general, the process of resource allocation in Cloud Computing involves three major actors : the Cloud customer, broker, and service provider, and includes three main phases :

Phase 1 : customers specify the amounts of resources such as CPU power, memory and storage capacities, and bandwidth, that are required to deploy their applications into the Cloud along with special constraints related to performance, QoS, and data localization. These demands are transferred to a Cloud broker and will eventually be provisioned in the form of VMs.

Phase 2 : the broker takes responsibility of finding an adequate CSP or multiple CSPs that can satisfy the customer's resource demands and constraints and transfers her request to the selected CSPs.

Phase 3 : CSPs provision the resources on their data centers according to resource availability in a way that optimizes infrastructure utilization and satisfies customers' needs for QoS. The cost of allocating the resources and running the Cloud application will finally be returned to

the customer.

In this thesis, the process of resource allocation is tackled from a security perspective. Our work aims at integrating the security element into the three phases of service deployment. First, security requirements will be modeled and combined with customers' requests. Second, service evaluation and selection that is usually performed by the Cloud broker will be based on the security level of CSPs and their ability to satisfy customers' security needs. Finally, the security aspect will be incorporated into the processes of resource provisioning and workload management within the Cloud data centers.

1.1.5 Service Level Agreement

Service and data outsourcing lies at the heart of Cloud Computing. In this context, Service Level Agreements (SLA) become critical to guarantee service availability and performance. A SLA is a negotiated contract between customers and CSPs, usually through a broker, where service levels, QoS parameters (e.g., availability, response time, throughput, etc), and resource consumption as well as fail-over policies and data backup requirements are specified and enforced via the application of penalties and compensations in case of violations. The contract can be monitored through several high-level and low-level metrics and is usually used to evaluate customers' satisfaction. Recently, security has attracted a lot of attention in the research on SLAs in Cloud Computing, leading to the establishment of so called Security-SLA, to which are attached some of the main contributions of this thesis.

1.1.6 CIA triad security model

The CIA triad security model serves as a tool for evaluating information security and developing security measures and solutions. The model focuses on the following three main goals to achieve the security of an information technology :

— Confidentiality, which concerns with protecting sensitive information from unauthorized disclosure.

— Integrity, which pertains to accuracy, completeness and validity of information during processing and storage.

— Availability, which concerns with information being operational and accessible whenever required.

The relative significance of each goal can vary from a situation to another according to the technology and use scenario. In a Cloud Computing system, this model is usually extended

to cover other aspects that are critical to service security, such as accountability and privacy. Accountability or compliance can be defined as the awareness and adherence to obligations (e.g., corporate social responsibility, applicable laws, ethical guidelines), including the assessment and prioritization of corrective actions deemed necessary and appropriate. This element concerns with keeping track of actions that are related to security responsibilities and violation of regulations, laws and Security-SLAs. It also stands for the ability of Cloud customers to detect computational faults and ensure that their workload is being correctly managed by the CSP. Privacy on the other hand is crucial to the Cloud security ecosystem due to the fact that customers' data are no more residing on the organization's premises but on the Cloud servers, which are operated by the CSP. In general, a breach to confidentiality or integrity will lead to privacy violation [6]. Privacy-preservability in the Cloud presents complicated technical challenges due to data distribution and customers' unawareness of their location.

1.1.7 Security vulnerabilities in Cloud Computing

Although it presents many technical, operational, and financial advantages, the Cloud Computing technology adds a new level of vulnerabilities that makes security one of the main challenges to its adoption. The two major security vulnerabilities [7] that arise when deploying services to the Cloud are :

— Third-party control, which is due to service and data outsourcing. When using Cloud services, customers' control over their data will be reduced, and potential threats to confidentiality, integrity, and availability might materialize (e.g., if the access rights to these data are abused by a malicious Cloud insider). These threats can lead to serious financial and technical damages, especially if the outsourced data are sensitive.

— Multi-tenancy, which is a Cloud characteristic related to resource sharing. In a Cloud infrastructure, virtualized resources including computational power, storage, databases, and applications are shared among customers to achieve optimal utilization. This characteristic entails a high risk Cloud environment that is susceptible to potential threats to data confidentiality and service availability.

Other vulnerabilities related to virtualization also exist, and are discussed in detail in [8], along with their associated threat vector. For instance, VMs' co-location on the same physical server can lead to data leakage via covert channels that could form without the awareness of the hypervisor. Moreover, VM mobility, which is an essential feature of dynamic and flexible workload management in Cloud data centers, can also be the origin of a security compromise. Securing VMs during creation, replication, execution, migration, and deletion is critical to the security of a Cloud service.

1.1.8 Major security threats in the Cloud

Cloud Computing services are accessed via the internet and using the web, so they are prone to the traditional security threats associated with these technologies from flooding attacks to browser-related attacks. Moreover, the damage that these threats could cause in a Cloud environment is usually augmented comparing to a traditional infrastructure. For instance, a DoS attack on a Cloud application might not only affect the victim service, but also other services that are co-hosted on the same server due to the property of resource sharing. Data hosted in the Cloud could be compromised during storage, transmission, and processing. The Cloud Security Alliance (CSA) [9] specified the top threats to Cloud Computing security as : abuse and nefarious use, insecure Application Programming Interfaces (API), malicious insiders, shared technology, data loss or leakage, account or service hijacking, and unknown risk profiles [10]. We will elaborate more on these threats throughout the thesis.

1.2 Problem definition

The Cloud Computing model involves many potential security threats inherited from its architectural model and technical properties like multi-tenancy and data distribution. These threats create security and trustworthiness concerns for customers and hold back the migration of services towards the Cloud. With the abundant emergence of Cloud Computing companies, security becomes one of the main driving factors of the Cloud market today, especially for businesses that deal with sensitive information and confidential data. CSPs are expected to maintain the security of the Cloud service and guarantee its availability. According to the study in [11], the rate of organizations that completely trust the public Cloud infrastructures today to protect their data is only at 23%. Every year, the Cloud market giants witness severe incidents and security breaches causing Cloud outages and failures, which affect the production process and result in lost data and revenue. For instance, in 2017, Microsoft Skype Europe users suffered from connectivity problems due to an apparent Distributed Denial of Service attack (DDoS) that affected the whole communications platform [12]. Similarly, in 2016, a number of AWS VM instances hosting critical workload for big companies subsequently failed because of a power outage in the region of Sydney, Australia, resulting in a serious service disruption [13]. Customers blamed AWS, the world's largest CSP, for not being sufficiently prepared for such incidents.

CSPs are in need for the right strategies to increase their public trustworthiness and improve their reputations. To speed up the migration of services to the Cloud and allow the customers to unrestrictedly benefit from the advantages of this technology, security offerings need to

be fully transparent to customers and built on clear terms that reflect the reality of a Cloud environment, with the possibility of violation detection and compensation. Cloud Computing services should be evaluated in a security context to demonstrate their ability to cope with the security requirements of customers' applications. Security satisfaction is a key concept to the success of a Cloud-hosted service. To be able to measure this satisfaction, security needs should be modeled and represented in a way that facilitates their interpretation and evaluation. Modeling security requirements and offerings is a tough task due to the lack of standard vocabularies and performance indicators that describe the implementation of security mechanisms in current clouds. The standardization of security terms will promote the usability of Security-SLAs as a strategy to protect both customers and CSPs and enable the existence of thorough, fair, and practical methods for evaluating and comparing Cloud services. Rendering possible the relative evaluation of CSPs' security levels will also allow the Cloud stakeholders to add a flavor of security-oriented competitiveness to current service pricing schemes.

CSPs need to be able to explicitly describe to customers the security level of their infrastructures using adequate and comprehensible terms, and customers want to be able to satisfactorily assess this level with respect to their security requirements. Thus, the need for suitable metrics and tools to effectively measure and monitor the quality of provided security solutions in order to predict failures and reduce the possible damages. The lack of these appropriate security terms and assessment methodologies is the first motivation to our work in this thesis. On the other hand, a rigorous security evaluation methodology should not solely be based on well-defined evaluation criteria, but also be aware of the trade-offs that need to be deliberated among them in the evaluation process. For instance, it might not be sufficient to assess a security mechanism by only measuring the security strength that it provides without studying its effect on application performance and QoS. State-of-the-art Cloud service evaluation techniques mostly aim at ranking Cloud services according to several factors, without emphasizing the relative significance of these factors to customers' applications. Moreover, these evaluation factors are predominantly qualitative, and rarely highlight the security aspect in the evaluation. With the growing number of CSPs, and the increasing interdependence between customers' security objectives, the complexity of the decision making process is amplified. This creates the need for a formal decision model that integrates a set of evaluation techniques and specific selection criteria.

Studying security satisfaction in the context of Cloud service provisioning is normally challenging, especially in the presence of multiple interrelated services with different security requirements, which can not be always satisfied by a single CSP. This fact entails the need

for appropriate modeling of the requirements and the whole service composition process in a security-oriented fashion. Exploiting the Multi-Cloud service deployment model to optimize the application's security satisfaction could be the beginning of the rope towards a solid solution. However, this optimization process has a multi-objective flavor that can hardly be disguised, since satisfying security rarely consists in satisfying a single factor or criterion, but mostly involves satisfying several competing criteria in parallel. Another complication that adds to the problem of optimization of security satisfaction is the granularity level of security implementation within the Cloud infrastructure. Security solutions might not be uniformly deployed across the whole data center. In fact, the physical servers of the infrastructure can provide different levels of security integration according to the constraints on resource utilization and QoS, and the security requirements of the managed workload.

After deploying the service to the Cloud in a way that satisfies its security requirements, the CSP should ensure security satisfaction during the whole service lifetime and avoid Security-SLA violation. This includes the execution of customers' workload on her own infrastructure and on other infrastructures in case of a federation scenario. CSPs usually exploit the concept of federation for performance-related or financial reasons. For instance, a CSP might federate her workload to another CSP whose data center is geographically closer to the application end users in order to deliver higher QoS. Also, she might federate her workload to other CSPs only for the sake of escaping the legal requirements imposed by the jurisdiction of the region where her data center is established. However, the security element needs to be considered in this process as federating a workload to another CSP might not always satisfy the security constraints that are critical to its protection. To reduce the violation of Security-SLA during service execution, the problem of security satisfaction will require deep integration into the tasks of workload management. For instance, the ability of CSPs in achieving the security level that is required to manage the federated services should be effectively evaluated before a federation can form. Along with the difficulty of security level quantification introduced by this evaluation, the problem of federation formation itself presents some challenges. It belongs to the family of coalition formation problems which are very popular in multiagent systems. For these problems, finding the optimal coalition is NP-complete, hence the need to adopt a computationally efficient solution, since Cloud federation formation usually occurs in online mode.

The problem of avoiding or limiting Security-SLA violations in a Cloud Computing infrastructure is naturally tricky. The challenge does not only reside in effectively implementing the suitable security solutions and satisfying customers' security demands, but also in closely managing the highly intermingled factors that drive the dynamic and unpredictable beha-

vior of a Cloud Computing system. CSPs should be able to possess high level of control over the security of their infrastructures and be able to thoroughly anticipate and counter security threats and limit their propagation within their data centers. Security risk in the Cloud Computing environment, which is usually defined in terms of existing vulnerabilities and possible threats, must be continuously assessed in real-time and tightly coupled with the process of Cloud resource management. Indeed, a resource provisioning framework that is primarily based on the evaluation of security risk will provide the CSP with the opportunity of proactively controlling the security of her resources and eliminating potential security threats before they even exist on the infrastructure. In general, security plays a minor role in the process of resource allocation and management in today's Cloud data centers, which objectives are usually focused on performance and QoS, energy-efficiency, or profit maximization. The problem mainly lies in the security evaluation phase. If security risk assessment is to be integrated in an optimization problem, it must follow the path towards suitable mathematical modeling.

Increasing security integration within the Cloud also has a financial challenge. The CSP is usually responsible for securing her Cloud services and providing basic security capabilities. However, the implementation of advanced security solutions is not necessarily a priority for the CSPs, since it introduces additional costs to their budgets, and does not necessarily generate higher profit. These costs include the price of security infrastructure (e.g., firewalls, antivirus software, Intrusion Detection Systems (IDS), etc), salaries of security architects, and the expenses of security training programs. The return on security investment in an IT infrastructure is not usually measured in terms of the achieved revenue, but as a function of damage reduction, i.e, the decrease in loss expectancy due to security incidents and the rate of threat occurrence. According to the study in [14], the average total cost of a data breach is around 4 million US$. This gives an idea about how crucial is to secure a Cloud infrastructure that hosts thousands of services and huge amounts of users' sensitive data. To alleviate the costs of security investments, the operations of security integration could be executed in cooperation with the consumers, who will mostly be ready to participate in this process since they are the primary beneficent.

These problems have led to the elaboration of the following research questions that we addressed thoroughly throughout the thesis :

— How to characterize the security level of a Cloud Computing infrastructure in quantitative terms ?

— How to exploit the concept of Security-SLAs for suitable security modeling and effective evaluation of security satisfaction ?

— How to enable Cloud customers to perform rigorous and optimized security-based service selection ?

— How to perform security-aware application deployment and service placement in the Cloud ?

— How to ensure the satisfaction of security requirements during workload execution and federation ?

— How to enable CSPs to increase their control over the security of their infrastructures through the process of resource allocation and provisioning ?

— How to support CSPs in the process of security integration from a financial perspective ?

1.3 Research objectives

This thesis essentially aims at improving the comprehensibility of security in Cloud Computing through the design of advanced quantification and evaluation methodologies. The goal is to increase the integration of the security element into the definition of the fundamental problems in the Cloud such as, service selection, resource provisioning and allocation, and workload management, which can hardly be done without the ability to effectively quantify it. More specifically, the objectives of the thesis are :

— Propose a security evaluation methodology for Cloud Computing based on the definition of quantitative security metrics.

— Create a measurable model for the Cloud Security-SLA that could be integrated into existing Security-SLA templates.

— Design a security-oriented Cloud service selection framework that weighs up the different trade-offs that exist among the security evaluation factors.

— Model and solve the problem of Cloud service placement from a security perspective.

— Propose an approach to the integration of the security element into the formation of Cloud federations.

— Design a Cloud resource allocation and provisioning model based on the evaluation of security risk within the data centers.

— Propose a scalable solution to the security risk-aware Cloud resource allocation problem.

— Conceive a mathematical model for the problem of resource allocation with the objective of reducing the cost of security investment.

— Evaluate the performance of the proposed solutions according to the concepts and metrics defined in the literature.

The realization of these objectives will be discussed in detail in chapter 3.

1.4 Main contributions and their originality

The main originality of this thesis lies in the design of mathematical models that incorporate the security aspect to describe the traditional problems of application deployment, resource provisioning, and workload management in Cloud Computing infrastructures from a security perspective. These models were conceived following a thorough evaluation of the Cloud security services and an innovative description of applications' security requirements that permitted the quantification and integration of the security element. The principal contributions consist of modeling these problems as well as proposing plausible solutions, and can be described as follows :

— **Proposing a quantitative security evaluation methodology that covers all the Cloud architectural layers.** This first innovation resides in defining a set of qualitative and quantitative metrics that describe the security level of a Cloud service in the context of the three service delivery models : SaaS, PaaS, and IaaS. The goal is to allow the creation of a platform where the security provided by CSPs could be rigorously and numerically assessed and compared.

— **Proposing an advanced approach to service selection based on the design of a measurable Cloud Security-SLA.** The set of security metrics that we propose is transformed into a group of Security Service Level Objectives (SSLO) to form a new and measurable model of the Security-SLA in the Cloud. The quantification of the Security-SLA paves the way for the establishment of understandable and practical representation of CSPs' security offerings and customers' security requirements, and enables the creation of a robust security-oriented Cloud service selection approach. The approach aims at effectively capturing the multi-objective aspect of the service selection problem by deliberating on the trade-offs that might exist among the different security attributes of the CIA triad security model.

— **Modeling the problem of Cloud service placement with the objective of optimizing the satisfaction of security requirements.** The proposed security metrics are then integrated into a constrained optimization model that describes the

problem of Cloud application deployment in a security-aware fashion. The model aims at maximizing the satisfaction of security requirements of Cloud applications through a process of security-based service composition in Multi-Cloud environments. The problem has a multi-objective flavor and emphasizes multiple factors related to the implementation of security solutions such as performance, cost, and risk. This is the first work in the literature that quantitatively accentuates security satisfaction in the process of service placement on the Cloud data centers.

— **Designing a security-aware Cloud federation formation framework.** The integration of the security factor into the process of federation formation in the Cloud constitutes another major innovation of this thesis. What we propose is a game theoretical approach to modeling the problem of federation formation from a security perspective. The approach is based on the evaluation of CSPs' security levels according to the Security-SLA, and aims at providing secure execution of federated workload and reducing security violations.

— **Proposing an approach to security risk evaluation and integration into the problem of Cloud resource allocation and provisioning.** The integration of the security aspect into the process of Cloud resource management constitutes a fundamental contribution of this thesis. The basis of this integration lies in the evaluation of security risk that customers introduce to the Cloud infrastructure. The goal is to support the CSPs in increasing the security level of their data centers by implementing a threat prevention model that combines resource allocation with security risk management in InterCloud settings. First, the security risk-aware resource allocation problem is modeled as a linear constrained optimization problem. Then, two different metaheuristics : Genetic Algorithm (GA) [15] and Artificial Bee Colony (ABC) [16] are applied to solve the problem in online mode.

— **Designing a Dominant-Strategy Incentive-Compatible (DSIC) online auction mechanism for the allocation of secure Cloud resources.** A significant innovative aspect of this thesis lies in the visualization of some of the problems from different angles. Our objective is to try to capture the perspectives of both CSPs and customers while trying to address the challenges to security evaluation and integration. For instance, the challenge to securing a Cloud infrastructure is not solely technical, but also financial. To support the CSPs in reducing the cost of security investments, we model the problem of allocation of secure Cloud resources in an auction-based context. To solve the problem, we design a DSIC online mechanism that aims at allocating the resources to the customers who valuate their security the most.

In general, these contributions are all related to the subject of security satisfaction in the Cloud, from both CSPs and customers' perspectives. The designed models and proposed solutions were all implemented and evaluated through extensive experimentation.

1.5 Thesis structure

Chapter 2 will review the literature related to each element of the research problem that we described. An analysis of the limitations of existing work and the gaps that must be filled will also be elaborated throughout this chapter. In chapter 3, a detailed description of our research work and published articles is given, and the relationship between our objectives is emphasized.

Chapter 4 presents the full text of the article titled "Towards Quantification and Evaluation of Security of Cloud Service Providers", which was published in the Journal of Information Security and Applications. The main contribution of this article lies in the set of quantitative and qualitative security metrics that we defined to evaluate the security of CSPs. The article proposes a security evaluation methodology in which several Cloud security services are quantitatively assessed.

Chapter 5 presents the full text of the article titled "A Broker-based Framework for Standardization and Management of Cloud Security-SLAs", which was published in Computers and Security, Elsevier. In this article, the set of security metrics previously proposed was improved and integrated into a standard form of the Security-SLA to model customers' security requirements and CSPs' security offerings. A security-oriented Cloud service selection approach was then proposed based on the evaluation of the trade-offs that might exist among the different security attributes of the CIA triad security model according to the security requirements of customers' applications.

Chapter 6 presents the full text of the article titled "Service Assignment in Federated Cloud Environments based on Multi-Objective Optimization of Security", which was published in the IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud 2017). The article mainly addresses the problem of security satisfaction in the context of service placement in a Multi-Cloud setting. It proposes an approach to application deployment on the Cloud data centers based on a security optimization model, in which the relationship between several factors related to the implementation of security solutions such as performance, cost, and risk is closely highlighted.

Chapter 7 presents the full text of the article titled "Towards Security-based Formation of Cloud Federations : A Game Theoretical Approach", which was published in the journal of

IEEE Transactions on Cloud Computing. The article emphasizes the importance of securing customers' workload in the context of Cloud federation. It proposes a federation formation model based on the theory of coalitional games. In the model, the security of CSPs is evaluated with respect to a security baseline derived from the Security-SLA and integrated into a utility function that forms the essence of the proposed solution.

Chapter 8 presents the full text of the article titled "Security Risk-Aware Resource Allocation and Provisioning in Cloud Computing", which is currently under review by the journal of IEEE Transactions on Parallel and Distributed Systems. The article presents a novel approach to resource allocation in the Cloud by considering the amount of security risk that customers introduce to the infrastructure, which we relatively evaluate according to the security implementation that customers require. In the article, the problem of Cloud resource allocation and provisioning is modeled in a security risk-aware context with the objective of optimizing resource utilization in an InterCloud infrastructure setting while imposing additional constraints related to security risk estimation. To produce scalable solutions to the problem, two metaheuristics optimization techniques from the family of evolutionary computation were used to solve the resource allocation problem : GA and ABC.

Chapter 9 presents a general analysis and discussion regarding the strong points and limitations of our research work in this thesis. Finally, chapter 10 concludes the thesis by presenting a summary of our research and a discussion about future potential research avenues that could extend our work.

CHAPTER 2 LITERATURE REVIEW

In this chapter, we discuss recent work that has been done in the research areas connected to this thesis. First, work on Cloud security evaluation will be examined. This includes the state-of-the-art methodologies for Cloud service evaluation and selection, security evaluation, and risk assessment. Then, a description of the recent research related to the design of Security-SLAs will be elaborated. Afterwards, current approaches to Cloud service composition and federation formation will be explained. Finally, the literature related to the process of resource provisioning in the Cloud will be investigated from two perspectives : security-awareness and profit maximization. At the end of the chapter, we carry out a thorough analysis of the described work to show its limitations and identify the research gaps that need to be filled.

2.1 Cloud security evaluation

In this section, we review the major work related to Cloud service evaluation, with particular focus on security evaluation and security-based service selection.

2.1.1 Service evaluation and selection

During the last five years, the migration of online services towards the Cloud Computing paradigm has witnessed a remarkable growth. Hence, the process of Cloud service evaluation and selection has attracted a lot of attention in both the research and industrial communities. Supported by the Cloud Services Measurement Initiative Consortium (CSMIC), which was formed with the objective of developing valid and globally accepted measures for evaluating the benefits and risks associated with the Cloud Computing model, the Service Measurement Index (SMI) [17] framework is in continuous development to provide and standardize such measures. These measures mainly aim at evaluating CSPs' capabilities regarding several Cloud characteristics, including accountability, agility, assurance, performance, security, privacy, and usability, to assist the industry and government decision-makers in the process of Cloud Computing adoption. Although the measures are significant, the proposed evaluation procedure is still primitive, and does not fully rely on quantitative metrics.

With QoS as the primary factor in the process of Cloud service evaluation, authors have been extensively proposing methodologies and techniques to evaluate and rank Cloud services. For instance, Wang et al. [18] performed a fuzzy synthetic evaluation of services by combining a personalized evaluation based on users' preferences with respect to several criteria like com-

puting power and storage, with uncertainty evaluation derived from QoS-related monitored data. Tang et al. [19] presented a recommender system for mobile Cloud services based on the assessment of QoS factors like response time, price, accessibility, safety, stability, and reputation. Their system can adapt to the user's usage context and recommend the appropriate service. Tang et al. [20] integrated QoS monitoring into the assessment of services' trustworthiness and combined it with users' feedback to evaluate and rank Cloud services. Ristov and Gusev [21] also evaluated the trustworthiness of the services as a function of their availability and reliability and used it to rank CSPs.

Garg et al. [22] proposed a ranking framework for Cloud services using the Analytical Hierarchy Process (AHP) technique from the family of Multiple Criteria Decision Analysis (MCDA) [23]. They used the following key performance indicators to measure service quality : sustainability, energy efficiency, suitability, accuracy, transparency, interoperability, availability, reliability, stability, cost, elasticity, usability, throughput, and scalability. Casola et al. [24] proposed a quality evaluation framework based on the formalization of CSPs' offerings and customers' requirements through a meta-model that abstracts SLA policies. They also applied the AHP technique to compute the relative weights of the quality characteristics defined in the model. MCDA techniques have been widely used in this context. MCDA is a field of operations research that has proven its effectiveness in solving complex decision-making problems. In [25], Whaiduzzaman et al. review the most used MCDA techniques in the context of Cloud service selection. For instance, to capture the multi-dimensional nature of the Cloud service selection problem and overcome the challenges of uncertainty and subjectivity in the decision making process, Wibowo et al. [26] proposed to apply a fuzzy MCDA method to evaluate the performance of Cloud services. They used the following five evaluation criteria which are very common in the literature : security, performance, accessibility, usability, and scalability. Rehman et al. [27] also proposed to solve the service selection problem in the context of multiple criteria. They provided a general and abstract model to the problem and built an evaluation procedure that compares the Cloud offerings against users' requirements.

2.1.2 Security evaluation

The evaluation of security of the hosting environment forms an intrinsic part of the activities that accompany the migration of a business to the Cloud. This evaluation consists of assessing the ability of CSPs to respond to the business' security requirements, as well as validating their compliance to information security standards and regulations. In general, to evaluate the security of their Cloud Computing systems, CSPs resort to state-of-the-art third-party self assessment tools, like the certifications provided by the International Organization

for Standardization (ISO) and the International Electrotechnical Commission (IEC) in the ISO/IEC 27000-series on information security standards and best practice recommendations. The Security, Trust and Assurance Registry (STAR) program [28] developed by the CSA is the most established industrial program for security assurance in Cloud Computing. It highlights the concepts of transparency, auditing, and standardization. STAR program provides security certification to CSPs, as well as detailed best practices to validate the security status of their service offerings. The program is driven by two main research components : 1) the Cloud Controls Matrix (CCM) meta-framework [29], which provides a structured information security roadmap that groups all Cloud-related security controls ; and 2) the Consensus Assessments Initiative Questionnaire (CAIQ), which goal is to enable Cloud consumers and auditors to verify the compliance of CSPs to the security controls of the CCM and the CSA security best practices.

In the community of research on information security, several authors have tackled the problem of Cloud security evaluation. Rizvi et al. [30] proposed to validate CSPs' answers in the CAIQ through third-party auditors. Ristov et al. [31] presented an evaluation methodology to compare on premise and Cloud environments according to the ISO 27001 :2005 [32] security control objectives. They concluded that the existing general purpose security standards do not cover all Cloud security challenges, and proposed a new ISO 27001 :2005 control objective, called Virtualization Management, with two controls covering virtualization and VM security. In [33], Ristov and Gusev used the same controls to evaluate the following open source Cloud solutions : CloudStack, Eucalyptus, OpenStack, and OpenNebula. However, both evaluation plans are based on the allocation of vaguely defined security points, which limits their rigorousness and practicality. Akinbi et al. [34] analyzed the security status of Windows Azure platform. They divided the PaaS service into three main components : developer environment, virtualization, and storage, and evaluated the security of each component in terms of standard industrial security controls related to identity and access management, encryption key management, and VM security. They concluded that more efforts need to be put in securing the Cloud-hosted web-based applications against the security threats associated with VM vulnerabilities. Probst et al. [35] described an approach to automated evaluation and analysis of the effectiveness of deployed security mechanisms in Cloud Computing environments. However, the approach only focused on access control and intrusion detection/prevention systems, without emphasizing the high importance of data and virtualization security. Abuhussein et al. [36] identified a set of security and privacy attributes including data backup, data isolation, encryption, and hypervisor security to help the Cloud customers in assessing and selecting service offerings. Lin et al. [37] defined an index system for privacy-aware Cloud service assessment that covers the whole data lifecycle

within the Cloud, which involves data generation, transmission, storage, access, reuse, archiving, and disposal. They applied the AHP technique to compare and rank service providers according to the privacy-preserving capabilities of deployed security mechanisms. However, the authors did not specify how to numerically evaluate these capabilities. Bengong et al. [38] also applied a fuzzy AHP technique to evaluate the security of Cloud services according to several factors such as the security of platform facilities and operational safety.

The approaches described above lack the quantification aspect in the evaluation. They mainly focus on computing the relative weights of the evaluation attributes and performing an abstract ranking of the services without accentuating the ability to effectively and quantitatively evaluate these attributes. Other researchers emphasized the concept of security metrology in the Cloud. Luna et al. [39] spotted the research challenges to developing a security metrics framework for the Cloud. They stated that a well-known taxonomy that is able to model the Cloud unique features should be adapted in the development of metrics. Moreover, the authors highlighted the need for investigating formal models for metrics definition and respecting the requirements of scalability, interoperability, and comprehensibility in the design of a reference architecture. Mirković [40] proposed a measurable model for the Cloud by defining a system of metrics based on the ISO 27001 standard security controls. He argued that metrics must be achievable, relevant, and timely, and enlisted a set of tools that help in measuring Cloud security. However, the defined security metrics only measure the performance of security controls in the present ISO standard and do not cover all security aspects in a Cloud environment. Finally, Da Silva et al. [41] applied the Goal Question Metric (GQM) methodology [42] to design a security metrics hierarchy that describes the security level in a Cloud Computing environment. However, the proposed metrics were not based on a specific Cloud service taxonomy and did not consider all critical security aspects in the Cloud.

2.1.3 Security risk assessment

The research on security risk assessment in Cloud Computing is also evolving. Jouini and Rabai [43] performed a comparative study of information security risk assessment models for Cloud Computing systems. According to their study, these models present several shortcomings regarding threat classification and modeling, and risk estimation. Chopra et al. [44] studied the risk of migrating data and applications to the Cloud and performed a qualitative analysis of the associated risks prior to, during, and after the migration process.

Several works have approached the subject from a quantification perspective. Tang et al. [45] modeled the risk identification phase in Cloud Computing based on the Hierarchical Holographic Modeling (HHM) framework [46] and used the fuzzy set theory to compute the

probability of security risk. Ben Aissa et al. [47] proposed a model that estimates system security based on quantifying the costs to stakeholders using the mean failure costs metric. The challenge to their model lies in the difficulty of accurately quantifying the stakes, dependability, impact, and threat matrices that they defined. Shameli and Cheriet [48] proposed a quantitative and iterative approach to evaluating the security risk associated with the Cloud platform using a fuzzy multi-criteria decision making technique. However, they did not identify the risk elements. Tanimoto et al. [49] extracted and analyzed the Cloud risk factors using a risk matrix that classifies a risk into four kinds : transference, mitigation, acceptance, and avoidance. Then, they used this matrix to approximate the asset, threat, and vulnerability values according to the generation frequency and degree of incidence. Finally, Cayirci et al. [50] performed a ranking of Cloud services based on a risk assessment model that they proposed. In their model, they considered the lists of risk scenarios, vulnerabilities, and assets provided by the European Network and Information Security Agency (ENISA), and used CSPs' answers to the CIAQ from STAR to estimate the risk level.

2.2 Security-SLA in the Cloud

The notion of Security-SLA is assuming a fundamental role in the deployment of Cloud services. Several research and industry-driven projects have been investigating this concept for the last couple of years. For instance, the Multi-Cloud Secure Applications (MUSA) project [51] was initiated with the goal of supporting security management in the context of distributed Cloud applications. One of the project's major outcomes is a Multi-Cloud SLA generator that incorporates security requirements along with QoS parameters. Another example of such projects is the SPECS framework [52] that offers Security-as-a-Service in the Cloud based on the security parameters specified in the SLA, and provides the necessary techniques to manage its life cycle. Based on the SPECS framework, Casola et al. [53] presented an automatic monitoring architecture in which different security monitoring tools are used to collect the necessary information to validate the Service Level Objectives (SLO) specified in the Security-SLA. Moreover, in [54], Casola et al. presented a framework that enables the automatic implementation of a per-service SLA model, where customers' security requirements can be tailored according to service needs. However, the missing piece in these projects is the explicit and formal definition of the security mechanisms and metrics that should form the Security-SLA.

De Carvalho et al. [55] analyzed most of the work that has been done in the literature regarding security integration in the Cloud SLA. Based on the reviewed studies, they found that the top challenge to developing Security-SLAs is the definition of standard security metrics

that could effectively be measured and monitored. Rojas et al. [56] and Rahulamathavan et al. [57] elaborated on the security requirements that need to be considered in a SLA for clouds. De Chaves et al. [58] analyzed the encountered difficulties in the process of security metrics definition. They spotted the vagueness of current metrics and realized, as many other authors also have, that security metrics need to be quantifiable and measurable. They concluded that defining a security that can be quantifiable into a service level is a challenging task. On the other hand, Luna et al. [59] emphasized the importance of standardization of Security-SLAs in the Cloud. They identified the key issues that, if properly addressed, will finally promote the usability of Security-SLAs. These issues include the definition of clear and measurable SLOs and standardized vocabularies.

Meland et al. [60] analyzed the suitability of several deontic contract languages like the eXtensible Access Control Markup Language (XACML) and WS-Agreement protocol [61] for expressing security requirements in the Security-SLA according to a set of properties including feasibility and complexity. Hale and Gamble [62] tried to associate SLA terms with security controls by building a compliance vocabulary to enable the comparison of Cloud security offerings. Guesmi et al. [63] also presented a general language that enables customers to specify their requirements in terms of access control and security properties. However, these proposals did not present a thorough analysis of the security requirements of a Cloud Computing system. Moreover, the integration of quantifiable security metrics into the proposed contract languages was not highlighted.

Several approaches have been proposed in the context of Security-SLA management in the Cloud. For example, T. Rojas et al. [64] designed a Security-SLA framework that could be integrated into the SLA lifecycle. However, they stressed on the efforts required from the research community to adequately define the security requirements and metrics to be embedded in the agreement. Da Silva et al. [65] were the first to elaborate on the definition of security metrics and their relationship to Security-SLA management. Still, the metrics hierarchy that they proposed lacks the quantification aspect and the full examination of the critical security aspects and requirements in a Cloud Computing environment. Bernsmed et al. [66] presented an approach to manage Security-SLAs in a Multi-Cloud setting. They suggested to identify security requirements in terms of the following five categories of security mechanisms : secure resource pooling, secure elasticity, access control, compliance, and incident management and response. Nevertheless, the defined requirements are not quantifiable and the process of establishing their assigned priorities lacks clarity, which could eventually lead to a sub-optimal service selection and assignment.

Few researchers have tackled the problem of security evaluation from the perspective of

Security-SLAs. For instance, Taha et al. [67] proposed an approach to assess and compare the Cloud security offerings according to the SLOs defined in the Security-SLA. They identified two types of security SLOs : Boolean, which are derived from CSPs' answers in the CAIQ, and numerical, which can be defined in terms of several levels such as encryption key size. The authors applied the AHP technique to perform weights' assignment and rank the Cloud service offerings. In [68], the same authors performed a quantitative assessment of an elaborated Security-SLA to evaluate the security levels of CSPs according to customers' requirements. However, their approaches were only based on the CAIQ reports and did not provide any details about the potential sources of numerical security metrics. Na and Huh [69] also developed a Cloud service selection model based on the evaluation of Security-SLA. In their evaluation, they considered five of the nine most notorious security threats to Cloud Computing [70] which they judged critical from the user's perspective : data breaches, data loss, account hijacking, insecure APIs, and malicious insiders. The model mostly focused on computing the subjective weights of the security controls using the Analytic Network Process (ANP) method, and did not elaborate on the security metrics used in the evaluation. Finally, Zhengwei et al. [71] proposed a quantifiable system of Cloud-oriented Security-SLA indicators based on existing standards and research. They applied the GQM metrics modeling method and divided their metrics into benefit and cost types. Then, they used a nearness calculation strategy to relatively evaluate service offerings with respect to customers' requirements.

2.3 Service composition in the Cloud

Service composition in Cloud Computing offers customers the opportunity to form composite services, where their functional and nonfunctional requirements can be optimally satisfied by different CSPs. Cloud vendors usually supply their services through a service pool, from which a Cloud broker can optimally select the appropriate combination of services after receiving customers' service requests. With the increasing number of CSPs and the complexity of defining the adequate and measurable QoS parameters involved in the composition process, the latter presents some challenges to the brokers. Supplying composite services in the Cloud is considered to be a NP-hard problem [72]. It is usually solved using classic or graph-based algorithms, or also combinatorial algorithms, due to its large search space and the need to provide service composition in real-time for some of the cases. Cloud Manufacturing (CMfg) has recently emerged as a new service-oriented manufacturing paradigm with the objective of providing a Cloud collaboration platform where distributed resources and capabilities can be combined into composite services to improve the performance and QoS [73].

The optimization of QoS parameters is usually the driver of the problem of Service Compo-

sition and Optimal Selection (SCOS). Rare is the work that highlighted security satisfaction in the operations of service composition in the Cloud. Wenge et al. [74] modeled the service composition problem in a collaboration context, where CSPs collaborate to respond to customers' requirements in terms of QoS and security, with the objective of maximizing their profit. However, the authors did not elaborate on these requirements and their way of measurements. The problem was modeled as a mixed integer program and solved using off-the-shelf optimization algorithms. She et al. [75] addressed the service composition problem from an access control perspective. To avoid undesirable information leakage between service composers, the authors proposed a composition protocol that integrates information flow control between the different services. Albanese et al. [76] proposed a general framework that enables users to express their soft and hard requirements for QoS and security during software module composition. However, the framework is not adapted to Cloud Computing services and their level of security requirements. Guesmi et al. [77] proposed a model for the placement of Cloud resources in a Multi-Cloud environment based on the formal definition of functional and nonfunctional requirements of the services. The model integrates a matching algorithm executed by the broker to select Cloud services that are compatible with customers' demands. However, the description of security requirements in their model is very abstract, and does not fully reflect the service security needs in a Cloud Computing system. Other works, such as [78] and [79] have addressed the SCOS problem from a multi-objective perspective, where multiple factors such as QoS and energy consumption are combined to form the optimization and selection model. In this thesis, the service composition problem is approached from a security satisfaction point of view, and the adherence to security requirements of the composite service becomes the main objective of such process.

2.4 Cloud federation formation

Cloud federation is a concept of service aggregation or outsourcing between CSPs, that aims at addressing the challenges of satisfying performance requirements and disaster-recovery planning through the approaches of workload co-location and geographic distribution [80]. One of the main motivations behind the process of Cloud federation usually consists in overcoming the obstacles to resources elasticity and coping with workload variations within the Cloud infrastructure. Also, Cloud federation can help CSPs in improving QoS and performance through regional workload redirection, and overcoming economic barriers and legal issues. Several Cloud federation creation and formalization architectures have been recently proposed [81]. In general, the Cloud federation formation problem is usually addressed from profit generation or QoS satisfaction perspectives. The security element is rarely assuming

the critical role it should have in this process. Bernsmed et al. [82] discussed the security challenges to Cloud federations and elaborated on the chain of transitive trust between the federated CSPs as the main security challenge to this process of service composition. Indeed, assuring the customer that the required security mechanisms are correctly implemented and performing as desired throughout the whole chain of CSPs becomes quite challenging. The authors highlighted the importance of defining a security level based on which the trustworthiness of CSPs in protecting the federated service could be evaluated, and pointed out the role that Security-SLAs could play in this area.

Several game theory-based approaches have been proposed in the literature to solve the problem of Cloud federation formation. For instance, Li et al. [83] proposed an algorithm for VMs trading in a Cloud federation using an auction-based scheduling mechanism that maximizes the profit of the federation members. Samaan [84] designed a resource sharing strategy in a Cloud federation that increases the revenue based on game theory. Mashayekhy et al. [85] introduced a Cloud federation formation game that allows CSPs to maximize their profit. Finally, Guazzone et al. [86] used the cooperative game theory to develop an algorithm that allows the formation of federations while maximizing the profit of CSPs and reducing the energy cost.

Other research have addressed the problem of formation of Cloud federations from a trustworthiness perspective. In [87], Hassan et al. proposed a federation formation mechanism using a trust-based cooperative game theory that allows CSPs to maximize their profit and minimize the SLA penalty cost on QoS by joining federations of trustworthy and reliable CSPs. In [88], Abdel Wahab et al. also proposed a trust-based hedonic coalitional game that permits the formation of Multi-Cloud communities of trustworthy services. However, none of these approaches have emphasized security satisfaction when addressing the subject of federation formation, which is critical to ensure workload protection within the formed federations.

2.5 Resource provisioning in the Cloud

Resource provisioning and allocation in Cloud Computing has always been an active research topic, due to what entails of optimization challenges. In general, the subject is tackled from QoS and energy efficiency perspectives. In this thesis, we are interested in dealing with the problem of resource allocation in a security-oriented fashion and at the IaaS service level, where resources are allocated to customers in the form of VMs. The Virtual Machine Placement Problem (VMPP) has attracted a lot of attention in the research community. It is considered to be a NP-hard problem [89], and aims at optimally assigning VMs to physical servers or data centers. In this section, we review some of the work that is of direct interest

to our contributions in this thesis.

2.5.1 Security-aware resource allocation

The allocation of security resources in Cloud Computing has been studied in previous research, but not extensively. Lu et al. [90] proposed a credit-based mechanism for resource allocation that will avoid the malicious usage of resources and, simultaneously, guarantee allocation fairness. Nandina et al. [91] proposed a framework that offers secure usage control of sensitive data within secure VMs. In the framework, resources are allocated to the VMs according to an optimization model that uses randomized algorithms. Liu and Lee [92] proposed a resource allocation algorithm for mobile Cloud Computing systems while providing a security guarantee. They used a semi-Markov decision process to model the allocation problem and calculated the optimal allocation policy with the use of linear programming. In their security-aware allocation model, security implementation is supplied by an extra number of VMs according to customers' security requirements. Liang et al. [93] proposed a Security Service Admission Model also based on a semi-Markov decision process to model the system reward for the CSP, which is the difference between the service incomes and the running expenses, while allocating the security requests of the customers in a mobile Cloud Computing setting. They divided the provided security services into two categories : normal security services which provide basic security mechanisms, and critical security services that provide more advanced security features.

2.5.2 Profit-driven resource allocation

Along with QoS optimization and power efficiency, profit maximization has been an essential factor when it comes to resource allocation in the Cloud. For instance, Goudarzi and Pedram [94] proposed a distributed solution to a SLA-based resource allocation problem, which maximizes the total profit in the system while considering the following three dimensions in the optimization : processing, data storage, and communication bandwidth. Nezarat and Dastghaibyfard [95] proposed a game theoretical model to maximize profit in a Cloud environment. In their model, a combinatorial auction mechanism is used to select the winners among the competent users. The design of truthful resource allocation mechanisms in the Cloud has been previously studied by several researchers. Nejad et al. [96] proposed an auction-based model for the problem of dynamic provisioning and allocation of VMs in the Cloud, and designed a truthful greedy mechanism that allocates the requests of the winning users and calculates their payments. Their mechanism gives incentives to users to be honest about their requests and valuations. Finally, Zhang et al. [97] proposed an incentive-compatible online auction-

based mechanism that allocates Cloud resources to users with heterogeneous demands. Their mechanism discourages the Cloud users from following a dishonest behavior when requesting resources, and its performance is comparable to that of the Vickrey–Clarke–Groves (VCG) mechanism [98].

2.6 Literature review analysis

In this section, we analyze the limitations of the presented work and discuss the research gaps that need to be filled. First, the examination of state-of-the-art security evaluation methodologies in Cloud Computing confirmed the absence of standard security metrics that can successfully reflect the security level of a Cloud infrastructure and enable a practical and fair assessment of the security of CSPs. In order to accomplish this mission, the security metrics should cover the evaluation of all service delivery models and allow for the assessment of the security of every layer of the Cloud architecture (network, virtualization, data, etc). On the other hand, as we noticed while elaborating on Cloud Security-SLAs, these agreements need to embed an adequate modeling of customers' security requirements that permits the effective quantification of their terms in order to increase their plausibility and promote their use for driving the process of Cloud application deployment. Few researchers have stressed on the role of Security-SLAs standardization and the definition of quantifiable metrics that will allow to effectively evaluate them and monitor their compliance. However, a complete framework that addresses these challenges is still missing.

Second, and beyond the quantification aspect, current security-based Cloud service selection approaches generally aim at performing a global evaluation of the security of the services without emphasizing the multi-objective aspect of the selection problem. When evaluating security, multiple factors usually intervene in the equation. For instance, when assessing a security mechanism, the impact that the mechanism will have on the performance of the application should be equally considered along with the security capabilities of the mechanism. Moreover, the evaluation of Cloud security offerings will be considered incomplete if it does not highlight the relative significance of the different security aspects to the evaluation process. For instance, it is expected that CSPs will offer different capabilities regarding the confidentiality, integrity, and availability of the delivered service, and customers will have variable appreciation of these aspects according to the security requirements of their applications. A thorough evaluation and service selection operation will unquestionably require to underline this reality.

Third, as we went through the different aspects of the Cloud service and resource management operations, its is clear that security satisfaction plays a minor role. QoS parameters

and profit-related factors are mostly the drivers of such operations. For instance, service composition and workload federation are usually performed with the objective of performance improvement, regardless of whether the service composers or the federated hosting environments are able to cope with the security requirements of customers' applications. To enable secure Cloud application deployment, these security requirements need to be effectively modeled and integrated into such operations. Moreover, the process of resource provisioning and allocation in the Cloud is normally oriented towards performance optimization and energy-efficiency guarantee, and rarely takes into account the security risk of the infrastructure. All of these operations will be tackled throughout this thesis from a security perspective, and rigorous optimization models that integrate the security element will be designed, to pave the way for secure Cloud-based service hosting.

CHAPTER 3 RESEARCH METHODOLOGY

The aim of this thesis is to propose a security evaluation model for Cloud Computing that can be effectively integrated into the problems of service selection, application deployment, and resource management with the objective of ensuring security satisfaction. To this end, several objectives were envisioned as announced in section 1.3. To attain these objectives, the work was carried out in three main phases. This chapter aims at explaining the different steps that led to the realization of these objectives and connecting them to the work presented in the subsequent chapters.

3.1 Phase 1 : Security evaluation in Cloud Computing

In the first phase of our work in this thesis, our efforts were directed towards the achievement of the first three objectives. The work consisted of : 1) first, defining a set of quantitative metrics that will form the basis of a security evaluation methodology ; 2) second, designing a standard and quantifiable form of the Cloud Security-SLA ; and 3) third, proposing a security-based service selection approach. These objectives were attained in the two articles titled "Towards quantification and evaluation of security of Cloud Service Providers" and "Broker-based framework for standardization and management of Cloud Security-SLAs" presented in chapters 4 and 5 respectively.

3.1.1 Security analysis

The process of security evaluation of Cloud Computing systems starts by identifying the different layers that form the Cloud architecture. Since the goal is to provide a global security evaluation framework that could be adapted to every service deployment scenario, it was fundamental to the process to consider the three Cloud service delivery models SaaS, PaaS, and IaaS. The first part of our systematic security evaluation methodology consisted of the following steps :

1. First, analyzing the security vulnerabilities within the Cloud Computing system that could lead to the materialization of potential security threats to the three attributes of the CIA triad security model : confidentiality, integrity, and availability. The accountability aspect of the Cloud security was also considered in the evaluation due to its special importance in a Cloud Computing system.

2. Second, identifying and categorizing these threats using the STRIDE approach [99], which is a threat classification method developed by Microsoft to evaluate the security of a computer system based on six threat categories : spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privileges.

3. Third, distinguishing the different security services that need to be provided in a Cloud Computing environment to ensure service and data protection at the different layers of the architecture. In our evaluation model, twelve security services were identified : authentication, access control management, web security, network security, storage security, virtualization security, physical security, data and computational integrity, data availability, service availability, security auditing, and security compliance.

4. Finally, associating the identified security services to the implementation of special security mechanisms and techniques.

This strategy enabled an in-depth analysis of the security of a Cloud Computing system and paved the way for the elaboration of effective security evaluation metrics.

3.1.2 Security metrology

A fundamental aspect of our first objective is to define a set of quantitative metrics that can describe the security level of a Cloud Computing system and integrate it into a security evaluation framework. The concept of security metrology is starting to get the attention it deserves from the Cloud research community. The development of measurable security metrics will facilitate the assessment of deployed security services and create an opportunity to perform a rigorous evaluation and ranking of the Cloud service offerings. Based on the Cloud security analysis that we performed as described in the previous section, a set of security metrics was defined to quantify the level of capabilities of the Cloud security services. These metrics will be presented in chapter 4. To derive such metrics, the GQM method [42] was applied. It allows the design of a measurement model according to three different levels : on the conceptual level, a goal describing the purpose of the measurement is specified in a particular context ; on the operational level, a set of questions is generated to characterize the achievement of this goal ; and finally on the quantitative level, a set of measurable metrics is developed in order to answer the generated questions. In our case, we were able to distinguish three types of security evaluation metrics based on their roles as follows : 1) implementation metrics, which intend to measure the progress in the implementation of information security programs, controls, policies and procedures ; 2) effectiveness metrics, which aim at validating if the security processes and controls are functioning correctly and operating as intended ; and

3) impact metrics, which goal is to describe the effects of security service failures on users' information security and organizations' missions. The development of these metrics formed the first step towards designing an effective security-based Cloud service selection model and building a standard Security-SLA that can be successfully enforced and monitored. These metrics can be eventually elaborated and described more formally according to the metric description model developed in [100] by the National Institute of Standards and Technology (NIST).

3.1.3 Security-SLA standardization

The growing market of Cloud-based services provides the customers with a handful of options when deploying their applications to the Cloud. However, evaluating and comparing service offerings is not a straightforward task, especially when security is involved. To be successful, the evaluation operation should have a clear understanding of customers' security requirements and CSPs' security offerings. This can be achieved through the creation of measurable Security-SLA that can validate the delivered level of deployed security services, which forms the second objective of this thesis. Following the analysis of major security vulnerabilities and threats in the Cloud Computing model and the establishment of a security metrics system, we were able to design a standard form of the Cloud Security-SLA that is based on the definition of a set of qualitative and quantitative SSLO parameters that permit to effectively represent customers' security requirements and CSPs' security offerings. This objective was achieved in both articles presented in chapters 5 and 7 respectively.

3.1.4 Security evaluation approaches

The security evaluation problem was approached in three different ways, each had its own merits and goals. Tackling the problem from a single point of view was insufficient to fill the research gaps that we encountered during our study of the related literature and allow for a successful integration of the security element into the Cloud-specific service management problems that we bear in mind. In general, effectively addressing an evaluation problem is quite challenging due to the different intervening parties and factors, which led us to establish multiple resolution perspectives, as follows :

— First, as we will see in chapter 4, a relative evaluation of the security levels of CSPs was conducted with the objective of providing a security self-evaluation platform that enables CSPs to assess their security services according to standard evaluation metrics and situate themselves in the Cloud market. This evaluation is completely quantitative and is performed by a third party in a fully objective fashion, i.e., the weights

assigned to the metrics during the evaluation are statistically computed to reflect their significance to the evaluation process.

— Second, we performed an evaluation of the Cloud security offerings in terms of the SSLO parameters which we defined in the standard Security-SLA. This evaluation consists in assessing the SSLOs offered by the CSPs with respect to customers' security requirements, to eventually generate a security satisfaction parameter in light of each security attribute of the CIA triad security model. This approach is described in chapter 5.

— Third, the security level of CSPs was evaluated with respect to a security baseline that we set in terms of the proposed Security-SLA to provide a reference security evaluation level. This approach was followed to enable an evaluation that is independent of users' security requirements, and which could be integrated into the process of secure federation formation in the Cloud, as we will see in chapter 7.

Each of these evaluation approaches helped achieving a different objective and in a different context. This strategy helped us overcome the limitations of designing a one-facet evaluation model that could eventually stand sterile in front of the adaptability and practicality requirements of our evaluation problem.

3.1.5 Security-based service selection

Our third objective involves the design of a security-oriented Cloud service selection framework that is capable of thoroughly evaluating the relationship among the factors and elements which are intrinsic to the decision making process. In its nature, the problem of Cloud service selection entails a high level of complexity and a considerable degree of uncertainty that can hardly be disguised. When security becomes involved, the problem becomes even more challenging, due to the strong connection between security and the other evaluation factors that are usually fundamental to the Cloud service selection process such as QoS and performance. To address the introduced challenges, two service selection approaches were proposed. These approaches are mainly based on the previous work in which we intended to address the security quantification problem.

3.1.5.1 A multi-criteria analysis problem

The first attempt to accomplish the third objective resulted in proposing a security-based service selection approach that is based on multiple evaluation criteria which we defined to evaluate the security level that accompanies the Cloud service offerings. The service selection

problem was modeled in a Multi-Criteria Analysis (MCA) context in which the criteria were hierarchically structured. The main criteria consisted of the following : performance, which describes the valid functioning and effectiveness of security mechanisms in achieving the desired security level ; cost, which characterizes the impact of security implementation on the performance of the Cloud application ; and implementation, which expresses the progress in implementing information security programs, controls, and procedures.

In our approach, security evaluation and service recommendation is performed by a Cloud broker who stores the CSPs' security information. The broker retrieves customers' security requirements and preferences that accompany a Cloud service request and returns the processed decisions. The AHP method, which proved to be powerful in modeling and simplifying complex and ill-structured MCA problems, was applied to the security-based service selection problem to compute the subjective weights of the criteria according to their relative significance to customers' security requirements. A procedure for objective computation of weights based on the collected security information was also included in the model to reflect the relative importance of security metrics to the decision making process. This work is described in detail in the article titled "Evaluation and Selection of Cloud Security Services based on Multi-Criteria Analysis" presented in appendix A.

3.1.5.2 A multi-objective optimization problem

To push the security-based service selection problem towards a more sincere reflection of reality and raise its level of applicability, we modeled the problem as a multi-objective optimization problem. Multi-objective optimization is an area in multi-criteria decision analysis where two or more conflicting objectives need to be optimized simultaneously. In our model, three objective functions were defined, each reflecting the dissatisfaction of customers' requirements at the level of confidentiality, integrity, and availability security attributes respectively. The idea that drives this approach is based on the assumption that the SSLOs offered by the CSPs are not usually monolithic in terms of these attributes, and no CSP will fully satisfy customers' security requirements without the need to establish some sort of trade-off among these requirements when performing service selection. The goal was then set to minimize the dissatisfaction of these competing attributes in the decision making process. To this end, an exhaustive search on the solution space was performed, due to its relatively small size, and a set of non-dominated solutions was generated to form the Pareto-front on the solution space. To find the most suitable solution to the customer's request, we proposed to use the pseudo-weight method [101], where a weight vector for each solution on the Pareto-front is derived and compared to the one declared in the request. This approach is based on

the Security-SLA that was previously proposed, and is described in detail in chapter 5.

3.1.5.3 Performance analysis

Since CSPs are not yet fully transparent regarding their security implementations and the security levels of their Cloud infrastructures, the collection of data related to the security evaluation metrics that we defined posed several challenges. To overcome this obstacle and demonstrate the effectiveness of the proposed security-based service selection approaches, we applied them in different scenarios where metrics data were randomly generated in a way that reflects the real capabilities of security mechanisms. In all cases, the goal was to show the applicability of the models to current Cloud Computing services and the way they present the possible solutions, regardless of the underling evaluation data. The computational time of the proposed approaches is also an important factor in the analysis of their performance, since eventually, these approaches will be integrated into large-scale Cloud application deployment and service placement models that need to be computationally efficient.

3.2 Phase 2 : Security-aware Cloud federations

In the second phase of our work, we aimed at integrating the concept of security satisfaction into the problems of service placement in Multi-Cloud environments and Cloud federation formation. The completion of the objectives 4 and 5 was carried out in this phase. These objectives were attained in the two articles titled “Service Assignment in Federated Cloud Environments based on Multi-Objective Optimization of Security” and “Towards Security-based Formation of Cloud Federations : A Game Theoretical Approach” presented in chapters 6 and 7 respectively.

3.2.1 Security-aware service placement

First, we emphasized the role of the Multi-Cloud service deployment model in providing secure application deployment to customers. A Cloud application is normally deployed using multiple Cloud services such as storage, computing, etc. Each service has specific security requirements, and deploying the application to a single Cloud infrastructure might not necessarily satisfy all these requirements since the CSP can sometimes perform better in some of the security domains and worse in others. The Multi-Cloud deployment model enables customers to distribute their services among several Cloud data centers that respond to the security requirements of their applications and avoid compromising their security needs. In this context, we proposed a service placement model in Multi-Cloud environments based on

security satisfaction. Chapter 6 presents the article in which objective 4 was achieved. The broker-based service placement architecture that we proposed in this article is federation-oriented but could be eventually adapted to a Multi-Cloud setting.

3.2.1.1 Modeling the problem

The objective of the model is to optimize the security level of the application in the deployment process. We believe that optimizing the security of an application should cover three aspects : (1) providing the application with the suitable level of security capabilities ; 2) minimizing the impact of security implementation on application performance ; and 3) deploying the application with minimal risk probability. These aspects have led us to model the secure service placement problem as a multi-objective optimization problem that aims at satisfying services' security requirements in terms of three factors : performance, cost, and risk, which we quantitatively evaluate with the help of the metrics that we proposed during the first phase of our work. The problem was modeled as a Constrained Integer Linear Programming (CILP) problem, in which three objective functions were formed and several constraints were derived from the SLA and Security-SLA. To reflect the reality of a Cloud Computing environment, the model was designed at the physical server granularity level under the assumption that physical servers will provide different security implementations and configurations.

3.2.1.2 Proposed solution

The multi-objective optimization problem was solved using a mixed-integer liner programming solver by applying the preemptive optimization method [102, 103], which assigns different priorities to the competing objective functions according to the application's requirements. These priorities depend on multiple factors such as the application type (e.g., web server, storage service, etc.), data sensitivity, or QoS parameters. The method seemed suitable to solve the problem since it allows a user friendly priority estimation and takes into consideration the fact that the customer might not be an expert in security by enabling her to effectively negotiate her requirements with the broker during the service placement process. The preemptive method performs the optimization by considering one objective at a time according to priorities. After optimizing every objective, an optimal objective value is obtained and used as a bound in a new constraint when optimizing the next objective. The final solution constitutes an efficient point of the initial set of feasible solutions. This approach will allow the broker to effectively deliberate on the trade-off among the different security factors during service assignment on the Cloud data centers.

3.2.1.3 Performance evaluation

To study the performance of the model, a simulation was performed in MATLAB. Simulation parameters were randomly generated due to the absence of real data regarding the proposed metrics. The power of the model was underlined through the evaluation of Security-SLA and SLA violations, and the effectiveness of the security-aware service placement approach in ensuring security satisfaction in a federation environment was demonstrated. The computational efficiency of the proposed solution was also analyzed in terms of the size of the optimization problem (e.g., number of services and number of physical servers).

3.2.2 Security-based federation formation

Objective 5 consists in proposing an approach to the integration of the security element into the formation of Cloud federations. This objective was attained in the article presented in chapter 7. In this article, we designed a security-based Cloud federation formation framework that involves two phases : 1) a security evaluation phase, in which the security levels of CSPs are assessed in terms of the quantifiable Security-SLA that we developed in chapter 5 ; and 2) a federation formation phase that incorporates this evaluation into the process of federation formation to ensure security satisfaction. The first attempt to address the problem of security integration into the operations of Cloud federation formation was based on the assessment of CSPs' security risk levels according to a set of criteria that we identified based on the security analysis that we previously conducted. This work resulted in the design of an initial federation formation model which incorporates security risk assessment into a cooperative game approach that drives the federation formation process. This model is described in the article titled "A Cooperative Game for Online Cloud Federation Formation based on Security Risk Assessment" presented in Appendix B. However, a complete framework was later designed, as presented in chapter 7, and a thorough analysis of the impact of this framework on security satisfaction and Security-SLA was conducted.

3.2.2.1 Problem definition

In order to form a Cloud federation and allow CSPs to find their potential federation candidates, the state-of-the-art has proposed several models, principally based on factors such as profit maximization, trustworthiness, and QoS parameters. Security was ignored in the Cloud federation formation process, mainly due to the difficulty of its evaluation. The federation of a service to a Cloud hosting environment that does not fully satisfy the security requirements of the service will lead to increased security risk and violations of the Security-SLA, which

will eventually hurt the reputation of the CSPs.

3.2.2.2 Proposed approach

The evaluation phase of the designed framework consisted in improving the Security-SLA model that we previously proposed by defining a set of measurable SSLO parameters that describe the security level of a CSP with respect to three aspects : security implementation, performance, and cost. The security level of CSPs was then evaluated according to a reference security baseline that we defined to provide a global evaluation of CSPs' security capabilities regardless of customers' security requirements. To evaluate the security of formed federations, the security level of CSPs was combined with their reputation, computed according to their customers' satisfaction, and the amount of resources they share with the federations.

In the federation formation phase, a game theoretical model was designed based on the previous security evaluation approach. The formation of Cloud federations represents a decision making situation that can be mathematically modeled as a combinatorial optimization problem. Cooperative game theory plays a major role in this area, especially when multiple decision makers are involved. It provides effective high-level approaches to describe the strategies and payoffs of the players. Game theory has been applied to solve different kinds of optimization and allocation problems in the areas of wireless networking, web servicing, and computer system security. In the case of Cloud federation formation, the number of possible coalition structures is too large to permit an exhaustive search for the optimal solution and finding it is a NP-complete problem [104]. Thus, a hedonic coalitional game model [105] was adopted to solve the problem. Coalition formation is a major subject in multi-agent systems, and hedonic games are a popular category of the coalitional cooperative games, in which profit allocation among the coalition members is not the main problem. In a hedonic game, the players are usually self-interested, and the stability property is guaranteed, that is, when the final partition of coalitions is formed, none of the players will have incentives to leave her current coalition to join another. In our game model, CSPs specify their preference relationships in terms of the security level of formed federations. The game reduces the loss in the security level of CSPs caused by moving from a non-cooperative state and joining a federation, and tends to maintain a stable rate of secure federations and reduce Security-SLA violations.

3.2.2.3 Performance evaluation

The game model was analyzed with respect to the stability properties defined in the literature, and its ability to provide secure workload federation was demonstrated. The proposed model

shows computational efficiency, and proves its power in reducing the rate and severity of Security-SLA violations when federating service workload between Cloud Computing data centers.

3.3 Phase 3 : Security-aware Cloud resource allocation

In the last phase of our work, our main goal was to integrate the security aspect into the process of resource allocation in Cloud Computing. The first attempt for enabling this integration resulted in the design of a resource allocation architecture that is based on security satisfaction. In this context, the resource provisioning problem was modeled as a linear optimization problem that aims at maximizing the security satisfaction of customers' services when placing them onto the Cloud data centers. Security satisfaction in this case was expressed in terms of the adherence of CSPs to customers' security requirements presented in the form of security features that need to be implemented within the data center. The proposed architecture is broker-based and can be easily adapted to a Multi-Cloud scenario, where services are composed among CSPs according to their security capabilities. This work is described in the article titled "Online Allocation of Cloud Resources based on Security Satisfaction" presented in Appendix C. Afterwards, the Cloud resource allocation problem was approached from two different perspectives : security risk assessment, and security investments.

3.3.1 Resource provisioning based on security risk

First, to support CSPs in reducing the security risk on their infrastructures, we designed a resource allocation model based on the assessment of customers' security risk which we evaluated prior to the allocation in terms of the required security implementation specified in their resource provisioning requests. In this work, objectives 6 and 7 were attained. The resultant contributions are presented in the article titled "Security Risk-Aware Resource Allocation and Provisioning in Cloud Computing", which is the subject of chapter 8.

3.3.1.1 Modeling the problem

First, we identified the different security risk factors in a Cloud infrastructure using the HMM framework [46], which is a comprehensive theoretical framework for modeling complex systems, and proposed a set of security parameters that could help the CSP in securing her Cloud environment. Then, we performed a relative evaluation of the security risk presented by customers' requests according to different security configurations that could be offered by the CSP in terms of the proposed security parameters. Based on this evaluation, we modeled

the problem of resource provisioning in an InterCloud environment in a security risk-aware context. The problem can be considered as a version of the VM Placement Problem (VMPP), which is usually described as a variant of the multi-dimensional Multiple-Knapsack Problem (MKP) [106] from the class of Bin Packing Problems (BPP). The objective of the proposed linear optimization model is to maximize resource utilization while keeping the security risk level of the Cloud infrastructure below the critical value.

3.3.1.2 Proposed solution

The multi-dimensional MKP has proven to be NP-complete [106], and no solution in polynomial time exists for this problem. Formal methods that are used to solve Integer Linear Programming (ILP) problems can only scale up to small problem instances. In our case, the method should be able to respond to a large number of requests in an online fashion. Therefore, we proposed to use two different metaheuristics approaches to find an approximated solution to the problem : the evolutionary GA [15], and the swarm-based ABC optimization [16]. The goal is to allocate the resources in a way that guarantees acceptable security risk levels within the Cloud while optimizing resource utilization. The solution should also allocate customers' VMs on the Cloud's data centers in a way that ensures workload balancing. The two proposed algorithms will be executed anytime a set of customer requests is placed on the Cloud controller server.

The GA is a population-based metaheuristics search algorithm that mimics the natural behavior of evolution, and is commonly used in automatic programming and machine and robot learning. It is suitable for the approximation of optimal solutions to complex optimization problems and could be considered computationally efficient. The GA usually starts by randomly generating an initial population of candidate solutions called chromosomes, and continuously applies the three genetic operators : selection, crossover, and mutation in order to improve the quality of the generated chromosomes with respect to the optimization problem's objective, until finally selecting the best solution to the problem. The ABC algorithm was also designed to solve complex computational problems, and is inspired by the natural intelligent behavior of honey bees. In this algorithm, the positions of the food sources constitute the candidate solutions to the optimization problem and their qualities represent the fitness values of these solutions.

3.3.2 Auction-based allocation of secure resources

Finally, to accomplish objective 8, we approached the problem of security integration in the Cloud from a financial point of view. The goal was to enable CSPs to increase their security

investments while reducing costs, and to allocate their secure resources to the customers' who require them the most. This contribution is presented in the article titled "Cloud Security up for Auction : a DSIC Online Mechanism for Secure Resource Allocation", which is shown in Appendix D.

3.3.2.1 Modeling the problem

We mathematically modeled the problem of secure resource allocation in an auction-based context. The problem was modeled as a multi-dimensional knapsack problem that aims at maximizing customers' social welfare, which is a standard criterion used to evaluate the outcome of an auction mechanism. In the model, customers are asked to show their valuation of the security of the Cloud resources in their resource provisioning requests in the form of bids. This valuation reflects the benefit that the customer receives when her request is allocated in the presence of security integration. This benefit is usually related to the degree of damage that a security breach could cause to the customer's service or data. To solve the problem while ensuring the truthfulness of the bidders, the theory of mechanism design was adopted.

3.3.2.2 The designed mechanism

Mechanism design [107] is an interesting approach to solving online allocation problems involving dynamic multi-agent environments where players should make truthful announcements for the sake of better system performance. Designing online mechanisms is becoming handy in multiple areas such as wireless networking, web servicing, and Cloud Computing. A DSIC auction mechanism is usually defined by the combination of an allocation rule that determines which bidders receive their requested items, and a truth-inducing payment rule that determines based on the allocation results, the amount that each bidder must pay for her received item [98]. The VCG auction mechanism ensures incentive-compatibility, but requires the implementation of an optimal allocation strategy. However, in the case of the defined problem, which is strongly NP-hard, implementing the VCG mechanism becomes computationally inefficient for large size problem instances.

To be able to allocate the secure Cloud resources on the fly while guarantying the truthfulness of customers, we designed a DSIC online mechanism that gives incentives to customers to reveal their actual requests and valuations as a dominant strategy. The mechanism maximizes the social welfare and reduces the cost of security investments by allocating the secure resources to the customers who valuate their security the most. The mechanism implements a greedy-based allocation procedure in which customers are prioritized according to their secu-

rity valuations, and a truth-inducing payment rule based on which the payments of customers are computed.

3.3.3 Performance evaluation

The proposed allocation algorithms were implemented and evaluated in terms of the quality of their produced solutions and their computational efficiency. Experiments were conducted using randomly generated resource provisioning requests based on the VM types usually offered by Amazon EC2. GA and ABC demonstrated their ability to achieve an acceptable approximation of the optimal solution and to be implementable in online mode. On the other hand, the designed DSIC allocation mechanism showed to be computationally efficient and achieved an acceptable approximation of the optimal solution usually produced by the VCG offline truthful mechanism. After evaluating the performance of all proposed solutions throughout the three phases of our work, the last research objective was fully attained.

3.4 Conclusion

This chapter presented a complete description of the research phases carried out in this thesis. The different methodologies that we adopted to solve the defined research problems were elaborated and the connection between the declared objectives and the contributions presented in the following chapters and appendices was established.

.

CHAPTER 4 ARTICLE 1 : TOWARDS QUANTIFICATION AND EVALUATION OF SECURITY OF CLOUD SERVICE PROVIDERS

Talal Halabi and Martine Bellaiche

Journal of Information Security and Applications, vol. 33, pp. 55-65, 2017.

Abstract

Security is still the main obstacle preventing companies and businesses which deal with private information and confidential data from migrating towards the Cloud. Cloud Service Providers should continuously perform security self-evaluation and assess the level of their security services in order to identify their limitations and improve their performance. We propose in this paper a methodology for quantification and evaluation of Cloud security services, based on a set of quantitative evaluation metrics which we developed using the Goal-Question-Metric (GQM) paradigm. We also make use of a case study scenario to demonstrate the effectiveness and practicability of the proposed methodology.

4.1 Introduction

The scientific and industrial communities are currently receiving Cloud Computing technology with growing attention and research efforts. Cloud Computing has many technical and financial advantages, such as scalability, resilience, performance, and portability. It forms a new business model and computing paradigm by providing the possibility of on-demand network access to an environment of shared and configurable resources (e.g., networks, servers, storage, applications, and services), the advantage of rapid provision and release with minimal management effort or service provider interaction, and the possibility of cost reduction through optimized computing [8].

According to NIST [108], the five essential characteristics of Cloud Computing are : on-demand self-service, broad network access, resource pooling, rapid elasticity or expansion, and measured service. The Cloud Computing architecture consists of three layers : (i) Software-as-a-Service (SaaS) which is run by Cloud Service Providers (CSPs) and mostly used by organizations ; (ii) Platform-as-a-Service (PaaS) which is a tool provided to develop applications without installing any software on the developer's side ; and (iii) Infrastructure-as-a-Service (IaaS) which includes storage, hardware, servers, and networking services operated, maintained, and controlled by the CSPs. Four different deployment models exist in Cloud

Computing : (a) public clouds in which the physical infrastructure is owned and managed by the CSP ; (b) community clouds in which the physical infrastructure is owned and managed by a group of organizations ; (c) private clouds in which the infrastructure is owned and managed by a specific organization or company ; and (d) hybrid clouds which include combinations of the previous three models [109].

Recent statistics [110] showed that most organizations that adopted Cloud Computing are running IaaS applications and over a public Cloud infrastructure. These organizations have witnessed a remarkable growth in Cloud benefits during the previous years in a variety of categories, including higher availability, geographic reach, cost savings, and business continuity. These statistics also showed that security concerns are one of the main factors slowing down the Cloud adoption, especially for healthcare companies and businesses that deal with sensitive and confidential information.

Although organizations can take advantage of many security benefits provided by the Cloud, compared to traditional on-premises technologies, the Cloud presents many specific features that introduce new vulnerabilities and security challenges. Besides its very large scale, resources in the Cloud are completely distributed, heterogeneous and virtualized. Thus, traditional security mechanisms in their current form, involving identity management, authentication, authorization, and infrastructure security, are no longer sufficient to ensure the protection for clouds. The Cloud may present different risks to an organization compared to traditional IT solutions, which makes moving critical applications and sensitive data to public Cloud environments a great concern for corporations [8].

To reduce these concerns and entice companies to take advantage of the many benefits of Cloud Computing technologies, CSPs should be able to assess their security levels through continuous self-evaluation. Evaluation of Cloud security services should be based on standard vocabularies and effective metrics, which are missing in today's clouds. Evaluating Cloud security services will also allow the providers to situate themselves within the Cloud market and help them determine the price of their services based on their security level. We propose through this research, a methodology to quantify and evaluate the Cloud security services. The contribution of this work is twofold :

— First, we identify the Cloud security services provided on each layer of the Cloud architecture and propose a set of measurable security metrics using the Goal-Question-Metric (GQM) method in order to evaluate these services and quantify their capabilities.

— Second, we design a methodology for relative evaluation of CSPs' security services based on objective weighting of metrics which facilitates automation.

The remainder of this paper is organized as follows. Section 4.2 discusses the literature

review related to Cloud security challenges and existing Cloud security evaluation approaches. Sections 4.3 and 4.4 define the Cloud security aspects and services respectively. In section 4.5, the set of Cloud security evaluation metrics is proposed. Section 4.6 describes the designed Cloud security evaluation methodology. In section 4.7, the methodology is applied through a case study scenario. Finally, section 4.8 concludes the paper.

4.2 Literature Review

In this section, the recent research that tackled Cloud Computing security challenges, issues, and existing and emerging solutions is presented. Then, the recent work related to the evaluation of Cloud security is surveyed.

4.2.1 Cloud Computing Security Challenges

Many recent papers address the security challenges in the Cloud. The Cloud Security Alliance (CSA) [9] specified the top threats to Cloud Computing security as : abuse and nefarious use, insecure interfaces and APIs, malicious insiders, shared technology, data loss or leakage, account or service hijacking, and unknown risk profiles [10]. It also developed security guidance for critical areas of focus in Cloud Computing including Cloud architecture, governance, and operation [111]. The guide describes best practices and recommended security solutions in the domains of data security, virtualization, encryption and key management, identity and access management, and incident response.

Subashini and Kavitha [112] identified the security issues that emerged due to different service delivery models in a Cloud system. They categorized Cloud security problems into four fundamental categories : data storage security, data transmission security, application security, and security related to third party resources. In their survey, they provided a detailed description of security issues in SaaS Cloud models, and considered data security, network security, service availability, and identity management as essential elements for secure SaaS applications.

Gonzalez et al. [113] presented a three-dimensions Cloud security taxonomy by arranging Cloud risks and vulnerabilities into hierarchical categories. The architecture dimension comprises the issues related to network security, interfaces and virtualization ; the compliance dimension deals with required responsibilities toward CSPs ; and the privacy dimension is based on data security and legal issues.

Khalil et al. [8] conducted a survey on the current Cloud security issues and state-of-the-art security solutions. They divided Cloud security into five categories : security standards, network, access control, Cloud infrastructure, and data, and discussed the security issues for

each category, in addition to the relationships between them. They also described known attacks against clouds in terms of causing vulnerabilities, provoked incidents, and related consequences, and provided a comparative analysis of the solutions and countermeasures.
Bhadauria and Sanyal [114] discussed the threats to security in Cloud Computing on the basic level, network level, and application level. They mentioned three different threats to basic security, namely : SQL injection attacks, Cross Site Scripting (XSS) attacks, and Man in the Middle attacks (MITM). They also described the threats to network level security, such as DNS attacks, sniffer attacks, issues of reused IP addresses, and BGP prefix hijacking. Finally, they described threats to application level security, such as security of hypervisor, Denial of Service (DoS) attacks, and other threats related to security of web applications.
Sen [115] identified the threats to different Cloud security aspects. He discussed internal and external attacks and data leakage as threats to confidentiality, user access and data segregation and quality as threats to integrity, and DoS, change management, physical disruption, and inefficient backup procedures as threats to availability.
Xiao et al. [6] described in detail the vulnerabilities and threats related to Cloud security, in addition to existing defense strategies, in the context of five different Cloud security attributes, namely : confidentiality, integrity, availability, accountability, and privacy. They identified Cloud vulnerabilities as : Virtual Machines (VM) co-residence, loss of physical control, bandwidth under-provisioning, and Cloud pricing model, and related them to three main Cloud characteristics : outsourcing of data, multi-tenancy, and massive data and intense computation.

4.2.2 Cloud Security Evaluation

Cloud security evaluation is a challenging area in the research on Cloud Computing. Although present security standards can help CSPs in implementing their security systems, more efforts are needed for Cloud security standardization. NIST [100, 116] and CSA [9] are investing much efforts in this area. CSA GRC [117] and CCM [29] projects are good examples of such work. The main motivation of our work in this paper is the lack of standard Cloud security evaluation metrics that can be used to perform a rigorous assessment of CSPs' security levels. The metrics proposed here can be described more formally using the model proposed by NIST in [100] for Cloud Computing service metrics description. Many authors have been trying to develop standard methodologies to evaluate the performance of security services in Cloud Computing. We review here some of the work conducted in this area.
The Service Measurement Index (SMI) [17] is in continued development to provide valid measures that help in evaluating and comparing Cloud services but no quantitative security evaluation metrics have been developed yet. Da Silva et al. [41, 65] proposed a Cloud manage-

ment methodology as well as an approach to Security Service Level Agreement (Security-SLA) based on a security metrics hierarchy that describes the security level in a Cloud Computing environment. However, they did not consider all critical security aspects and services in the Cloud. In [118], the same authors used the proposed system of metrics to evaluate the Return On Security Investment (ROSI) in the Cloud.

Ristov et al. [31] defined a methodology to quantify the ISO 27001 :2005 requirements [32] grouped into control objectives, based on which they compared on premises and Cloud environments. They concluded that the existing general purpose security standards do not address all Cloud security challenges, and proposed a new ISO 27001 :2005 control objective, called Virtualization Management, with two controls covering virtualization and VM control. However, they did not specify any metrics that measure the performance of the new security control.

Mirković [40] proposed a measurable model for the Cloud by defining a system of metrics based on the ISO 27001 standard security controls. He argued that metrics must be achievable, relevant, and timely, and enlisted a set of tools that help in measuring Cloud security. The security metrics defined in his work only measure the performance of security controls in the present ISO standard without covering all security aspects in a Cloud environment.

Probst et al. [35] proposed an approach for security evaluation and analysis in Cloud Computing environments. Their objective was to provide an automated way to evaluate the effectiveness of security mechanisms deployed to protect Cloud environments, but they focused only on access control and intrusion detection/prevention systems, and did not consider data and virtualization security. The authors did not provide any details about the evaluation metrics needed for their evaluation process, and their approach is currently being investigated and implemented.

Zhengwei et al. [71] constructed a quantifiable Cloud-oriented Security-SLA from multiple standards and existing research following a metrics modeling method, and proposed the use of a subjective and objective weighting synthesis method and nearness calculation approach to evaluate the distance to the best service level, the distance to the worst service level, and the distance to customers' demands. Na and Huh [69] also proposed a Cloud selection method based on weights evaluation using the Analytical Hierarchy Process (AHP) method, but did not discuss the evaluation metrics that can evaluate the security of CSPs. Luna et al. [119] and Taha et al. [67] also used the AHP technique to benchmark Cloud security based on the Security-SLA provided by the CAIQ [9], but they did not propose any other quantitative metrics to measure the security of CSPs.

The current Cloud lacks standard security evaluation metrics that cover all parts of its architecture (virtualization, network, storage, etc.). Current research is focusing on Cloud

Security-SLA selection techniques in order to help Cloud customers in choosing what best suits their requirements. Helping CSPs in processing a security self-evaluation and relative security comparison is also an essential step towards developing secure Cloud infrastructures and providing customers with better services, and that is the aim of what we propose in this paper.

4.3 Cloud Computing Security Aspects

Based on the security challenges and issues discussed in the previous section, we conclude four essential security aspects (or attributes) that need to exist altogether in a Cloud Computing environment, namely : confidentiality, integrity, availability, and accountability. The privacy and confidentiality aspects are evaluated together since they usually offer and apply similar security services and mechanisms.

4.3.1 Cloud Confidentiality

Cloud confidentiality concerns with protecting sensitive information from unauthorized disclosure. CSPs usually protect the confidentiality of customers' data by implementing a combination of different access control techniques. Unlike in house enterprise applications which could be protected with traditional edge security solutions like firewalls and proxies, Cloud-based applications require the implementation of more effective security measures to ensure confidentiality protection since they are running on untrusted networks and hardware [111]. The confidentiality of Cloud users' data and applications is usually protected using an Identity and Access Management solution (IAM) and appropriate virtualization and network security, in addition to the physical security implemented on the CSP site.

4.3.2 Cloud Integrity

Cloud integrity pertains to accuracy, completeness and validity of information and computation in regards with business requirements and expectations. Data and Metadata in the Cloud must be protected at rest, in transit, and while processed, both cryptographically and physically. Data loss and manipulation are possible threats in today's clouds where servers are untrusted in terms of security and reliability. Vulnerabilities introduced by the loss of physical control can lead to the compromise and loss of sensitive information.

4.3.3 Cloud Availability

Cloud availability concerns with information being operational and accessible whenever required. It is maintained by appropriate incident management plans, and protection against network attacks (Intrusion Detection and Prevention Systems (IDS/IPS)). Implementing incident response plans and computer forensics in a Cloud environment requires different tools, techniques, and training in order to accurately assess a security incident and capture appropriate evidence. Inefficient data backup and recovery plans may also affect data availability.

4.3.4 Cloud Accountability and Compliance

Cloud accountability and compliance can be defined as the awareness and adherence to obligations (e.g., corporate social responsibility, applicable laws, ethical guidelines), including the assessment and prioritization of corrective actions deemed necessary and appropriate. This element concerns with keeping track of actions that are related to security responsibilities and violation of regulations, laws and Security-SLAs through security audit and assessment plans, and penetration testing. Cloud accountability also stands for the ability of Cloud customers to detect dishonest computing and faults within the Cloud and to ensure that their workload is being correctly handled.

4.4 Cloud Computing Security Services

Based on a detailed analysis of the Cloud security issues and challenges related to all service delivery models (SaaS, PaaS, and IaaS), we identify the security services that are usually offered in the Cloud to protect the described security aspects. Figure 4.1 illustrates the Cloud architecture reference model with corresponding security services.

Authentication

System users and administrators have to be securely authenticated in the shared environment of Cloud Computing. In a regular enterprise application, subscribers are usually authenticated against the enterprise user store (Active Directory or Lightweight Directory Access Protocol (LDAP)) using a typical UserID and Password credential. Since Cloud applications are widely accessible through various devices, authentication and identity management mechanisms that manage the access of users or processes to Cloud systems must now operate across all components of the Cloud infrastructure. Hence, authenticating with simple UserID/password credentials would not be considered the best solution.

Cloud enterprises use open standards for identity management such as Security Assertion Markup Language (SAML) to help ensuring portability. Most enterprises usually plan for using stronger authentication techniques for their Cloud applications like risk-based authentication or two-factor authentication techniques. Some providers also offer the Single Sign-On (SSO) functionality [111] which allows users to access all Cloud services upon authentication through the provider's identity management solution. SaaS providers can delegate the authentication process to the internal LDAP/AD server of customers to provide them with management over users' identities. Providers also need to apply encryption to credentials and credentials exchange.

Authorization and Access Control

Unauthorized access to sensitive information in public, private, and hybrid clouds represents a major security concern. In the Cloud there is a need to specify IAM in terms of identity proofing, strength of credentials, and access control mechanisms for effective authentication and authorization. The entitlement procedure, which consists in mapping Identities and related Attributes to specific privileges (e.g., access to applications and data), is essential in this context. It transforms the customer's business and security requirements into a set of authorization rules that will manage the access to the Cloud system [111].
Regardless of the access control model (discretionary access control, mandatory access control, role-based access control, or attribute-based access control) used, customers and providers also need to consider emerging standards such as XACML to manage user account provisioning, authentication, and authorization and to flexibly enforce confidentiality and integrity requirements within the Cloud environment [111]. The creation, deletion, and maintenance of access policies can be provided by a standard policy management interface. In addition, logging and auditing of access to Cloud resources by customers and providers is a must.

Web Application Security

Cloud services are usually accessed via the web. Security holes in web applications on the Application/Service and OS/Platform layers create several vulnerabilities to Cloud services, especially SaaS. The top threats faced by web applications as defined by The Open Web Application Security Project (OWASP) [120] are : injection flaws like SQL, OS and LDAP injections, cross-site scripting, broken authentication and session management, insecure direct object references, cross-site request forgery, security misconfiguration, insecure cryptographic storage, failure to restrict URL access, insufficient transport layer protection, and invalidated redirects and forwards. Traditional network security solutions are not sufficient to protect Cloud services against these threats, thus the need for effective defense strategies at this level

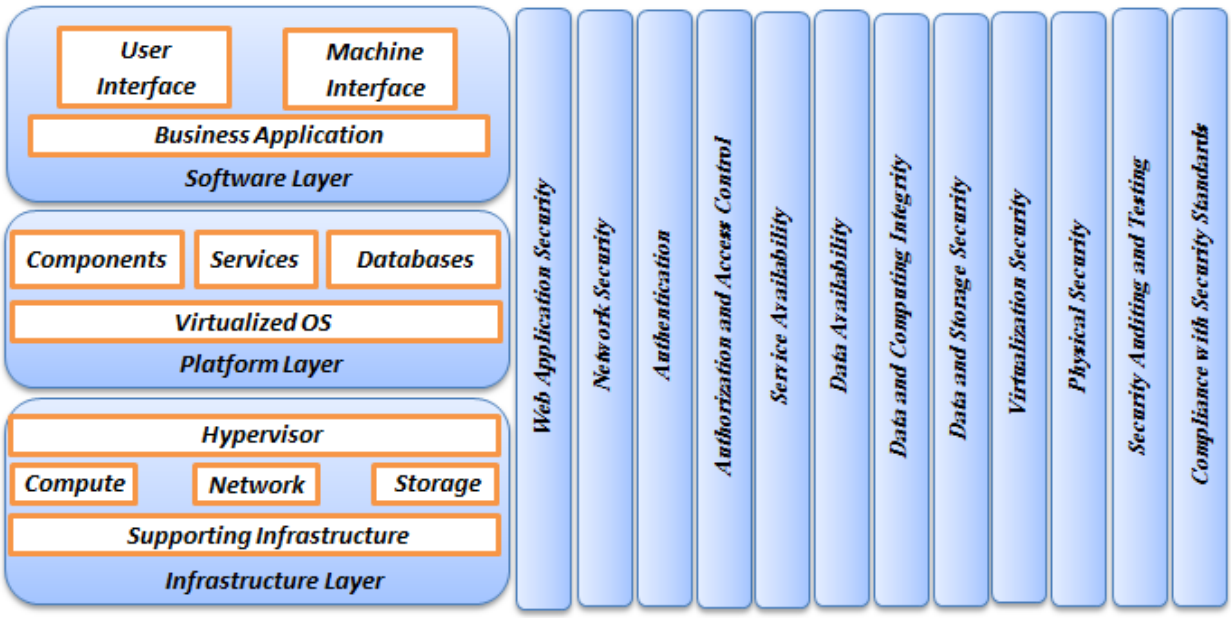


Figure 4.1 The Cloud architecture reference model and corresponding security services.

[112].

Network Security

Since all interactions with the Cloud service offerings happen over the Internet, CSPs should ensure that sensitive data as well as identities and access information are not leaked, and Cloud services are effectively protected against network layer threats such as DoS, Fraudulent Resource Consumption (FRC), MITM, Network Sniffing, and Port Scanning attacks. Most of CSPs usually employ SSL and TLS for traffic encryption and follow strong network security protocols. Enterprises may also connect to Cloud services through a Virtual Private Network (VPN) to ensure further network security [111].

Data and Storage Security

In both public and private Cloud deployments, and throughout the different service models, the lack of physical control over data and resource sharing bring a whole new set of security issues (privileged user abuse, snapshots and backups, data deletion, data leakage, etc.), which makes it important to protect data at rest and in transit, including data moving from traditional infrastructures to Cloud infrastructures, between Cloud infrastructures, and between instances (or other components) within a Cloud infrastructure [111].

Cloud providers usually implement encryption techniques and special key management plans at the application or network layers to protect customer's data from application level threats and multi-tenancy vulnerabilities. Also, database security is an essential component to protect SaaS applications data.

Virtualization Security

With IaaS Cloud offerings and private clouds, and at the backend of PaaS and SaaS Cloud providers as well, virtualization is the key technology. It provides many benefits such as better server utilization and data center consolidation. With the use of virtualization, CSPs can reduce hardware cost and increase operational efficiency.

However, for efficient use of virtualization, specific security techniques exist to alleviate its security concerns which can be present at the hypervisor level, like inter-VMs attacks and blind spots, or at the VM level such as instant-on gaps, data co-mingling, the difficulty of encrypting VM images, and residual data destruction [111]. Security of VM images and VM migration is also essential to protect customers' sensitive information from leakage and manipulation.

Physical Security

Physical security includes all measures which purpose is to prevent unauthorized physical access to a building, resource, or stored information. The CSP should make all facilities involved in providing the Cloud service available for inspection by the Cloud customer or auditor, which permits the evaluation of the physical security against the security requirements.

Data and Computing Integrity

Data integrity involves protecting the data from being lost and manipulated while it resides or is being processed in the Cloud, and maintaining the accuracy, completeness and validity of information and computations. CSPs should use appropriate data integrity protection techniques such as Provable Data Possession (PDP) techniques [121]. Integrity can be also threatened by dishonest computation on the Cloud servers' side where computation details are not sufficiently transparent to Cloud customers, so the Cloud servers may behave unfaithfully and return incorrect computing results [6]. Re-computation, replication, and trusted computing [122] are conventional strategies to check external computation integrity.

Data Availability

CSPs should be able to provide consistent backup and fast restoration of Cloud-based data

and resources in the event of service downtime or failure. Data replication policies are essential for disaster recovery to reduce the damages on customers' critical information. Some providers also deploy data encryption schemes to prevent loss or leakage of sensitive information in the backup data.

Service Availability

The infrastructure and services of CSPs should be resilient to hardware/software failures, as well as to DoS attacks. CSPs are responsible for maintaining service uptime and business continuity by reducing single points-of-failure (fault tolerance) and implementing effective incident response and remediation plans. Application Change Management also needs to be optimized in order to minimize service downtime and outage. In order to protect the Cloud from DoS and Distributed Denial of Service (DDoS) attacks, many recent works have proposed to implement defense solutions at the virtual machine level [123].

Security Auditing and Testing

Logging and auditing of access on the different levels within the Cloud are important to ensure system security and policies' fulfillment. Since Cloud customers are ultimately responsible for proving compliance with security standards regardless of the location of their data and systems, it is important to ensure that CSPs are obliged to undergo external audits [7], such as : (1) Security audit plans, which should be carefully designed to reflect proper organizational governance, and independently conducted to cover all resources, protocols, and standards. (2) Security assessment plans, which include industry standards-based auditing (e.g., NIST, ISO, etc) of Cloud services (e.g., infrastructures, applications) or on-premises systems by the customers or Cloud provided solutions. (3) Penetration tests, which are security testing methodologies that involve simulating an attack from a malicious source in order to measure the strength of the target's network and system security [111].

Compliance with Regulatory and Industrial Standards

Cloud services need to be periodically assessed to meet regulatory and industry standards [124] such as SAS Type I and II, ISO 27001, SOX, GLBA, and HIPAA, which cover a wide range of IT procedures, including logging, auditing, authentication, authorization, data access, data archiving, backups, recovery, and physical security of Cloud servers. CSPs should develop their systems in compliance with these regulations and implement effective mechanisms to protect data privacy and business continuity.

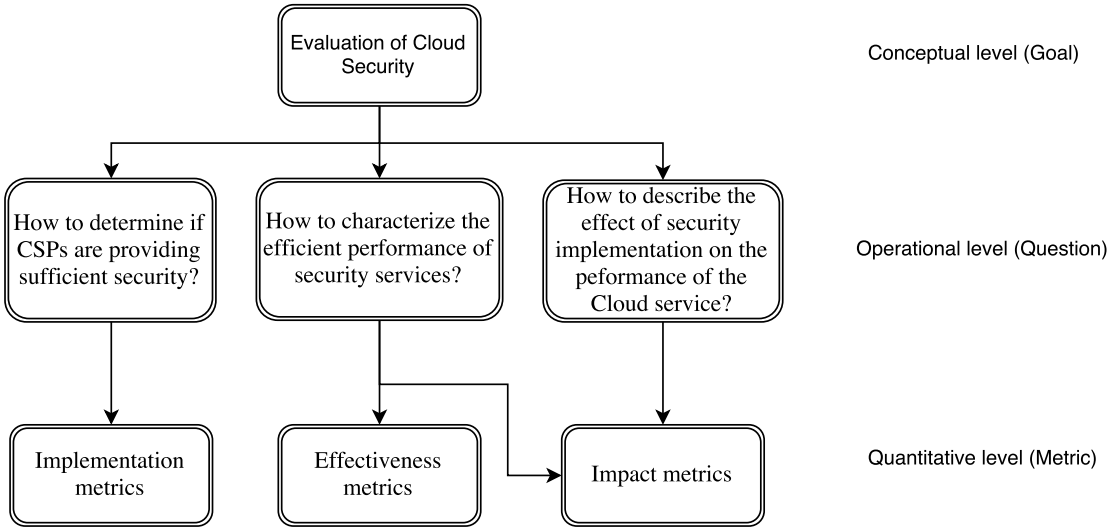


Figure 4.2 The Goal-Question-Metric structure for Cloud security evaluation.

4.5 Development of Cloud Security Evaluation Metrics

In this section, we make use of the Goal-Question-Metric (GQM) method [42] in order to develop a set of measurable metrics that can quantify the performance of Cloud security services. In the GQM definition phase, a measurement model is defined on three levels as shown in Figure 4.2 : on the conceptual level, a goal describing the purpose of the measurement is specified in a particular context ; on the operational level, a set of questions is generated to characterize the achievement of this goal ; and finally on the quantitative level, a set of measurable metrics is developed in order to answer the generated questions [42]. In our case, security metrics can be statistical (e.g., % security deployment) or technical (e.g., attacks detection rate), and with the help of the GQM paradigm and the guide in [125], we were able to distinguish three types of security evaluation metrics based on their roles as follows :

1. **Implementation metrics.** These metrics intend to measure the progress in the implementation of information security programs, controls, policies and procedures. Most of these metrics are expressed in the form of percentages to measure the progress in the deployment of appropriate security mechanisms.
2. **Effectiveness metrics.** These metrics are intended to monitor if the security processes and controls are implemented correctly and operating as intended. They help in measuring the performance and security level of deployed security techniques and procedures.
3. **Impact metrics.** These metrics are intended to articulate the effect of security service failures on users' information security and organizations' missions. They also

express the effect of implementing security mechanisms on the performance of the Cloud application (e.g., computational efficiency).

Information about implementation metrics is provided by the CSP through technical details concerning the deployment of services and applications. Information about effectiveness and impact metrics can be provided by multiple sources such as security auditing and testing tools, vulnerability scanning and risk assessment systems, and the Security Incident and Event Management (SIEM) system [126]. Table 4.1 shows the evaluation metrics that we have developed for each Cloud security service.

Table 4.1 Cloud security services evaluation metrics.

Authentication
Implementation
% Cloud nodes deployed with authentication mechanisms ;
% Applications deployed with :
- capability of integration with existing customer-based SSO solutions ;
- enforcement of Multi-Factor authentication for local and remote access of privileged users ;
- enforcement of policies on password strength and expiration, and blocking of invalid login attempts ;
- separation of identities' database from other databases.
Effectiveness
False Acceptance Rate and False Rejection Rate of authentication mechanism.
Impact
% Users or applications that suffer from credentials or data compromise per year ;
% Data loss or application offline-time due to authentication failure.
Authorization and Access Control
Implementation
% Resources configured with implementation of access policies and rules ;
% Applications deployed with :
- techniques for protection of sensitive configurations from unauthorized access ;
- the ability to use temporary access credentials ;
- implementation of risk-based entitlement decisions ;
- controlled access to all infrastructure layers (network, system, application, process, storage, and data) ;
- restriction, logging and monitoring of access to information security management systems ;
% Services offered with implementation of dedicated secure networks for management access ;
% Deployment of controlled access to the application program and object source code.
Effectiveness
Frequency of review of access control logs and accounts activity ;
Frequency of system users and administrators' entitlements periodic review ;
% Users entitled with trusted attributes.

Impact

Average response time of transactions ;

% Increase in response time with increasing users scalability.

Web Application Security

Implementation

% Applications deployed with testing of newly emerged APIs/interfaces.

Effectiveness

Intrusion detection success rate ; Intrusion detection false positives ; Injection attacks detection success rate ; Malicious activities detection rate by web application scanning ; False positives of web application scanners.

Network Security

Implementation

% Systems deployed with :

- configuration of security groups and Access Control Lists (ACLs) on virtual interface ports ;
- configuration of host-based and guest-based firewalls ;
- capability of IPSec VPNs between Cloud provider and customer's data centers ;
- configuration of SSL protected network endpoints ;
- configuration of network monitoring tools and IDSs against network attacks ;
- separation of the corporate and production networks ;
- implementation of VLANs for logical separation.

Effectiveness

Mean-time to discover attack ; Mean-time to mitigate attack ; Detection rate ; False positive rate ; Latency or response time ; Packet drop rate ; Throughput or network traffic rate.

Impact

Processing time ; CPU load or processing usage ; Memory consumption rates or memory overhead ; Network load or bandwidth overhead ; Financial cost of implementation.

Data and Storage Security

Implementation

% Applications deployed with :

- implementation of key management procedures ;
- internal storage of encryption keys ;
- capability of creation of a unique encryption key per tenant ;
- capability of open encryption methodologies ;
- enabling of HTTP Strict Transport Security (HSTS) ;
- implementation of data dispersion techniques ;
- implementation of data loss/leakage prevention techniques ;
- support of data secure deletion ;
- continuous monitoring of infrastructure against information security baseline ;

% Databases deployed with implementation of controlled access and authentication ;

% Databases deployed with SSL protected client connections (encrypted database transactions) ;

% Systems deployed with installation of anti-malware and anti-virus programs ;

% Cloud nodes separated by private network or that use data transfer encryption ;

Effectiveness

Frequency of update of anti-malware and anti-virus programs ; % Undetected virus and malicious activities ;

% Data encrypted at rest ; Strength of encryption keys and block ciphers ;

% Volumes protected from snapshot cloning/exposure and from being explored by the Cloud admins ;

% Volumes protected from being exposed by physical loss of drives ;

Number of separate PKI/number of different security domains ;

% Information security and privacy policies aligned with industry standards.

Impact

Percent of computational overhead ; Percentage of communication overhead ; Percentage of storage overhead ; Computational complexity ; Online latency ; Time overhead due to encryption/decryption by Data Loss Prevention (DLP) technique ; Storage nodes' probability of being compromised ; Storage node latency in responding to read-write requests.

Virtualization Security

Implementation

% Systems deployed with :

- VMs interference prevention and instances isolation mechanisms (against cross-VM attack via Side Channels) ;
- implementation of role-based access control on hypervisor level ;
- events monitoring and audit by hypervisor ;
- configuration of SSH secure communications ;
- capability of storage encryption ;
- validation of virtual images ;
- implementation of isolated migration network ;
- implementation of encrypted live migration ;
- monitoring and protection of instances (firewall, anti-virus) ;
- capability of VM encryption ;
- data destruction procedures after migration ;
- auditing and monitoring of in-motion VMs ;
- regular assessment of virtualization vulnerabilities ;
- VM backup, restoration and clean-up capabilities ;
- implementation of secure APIs for VMs/applications ;
- implementation of strong physical security on the hypervisor server.

Impact

% Performance degradation due to VMs isolation ;

% Slowdown in migration time due to migration security ;

Time spent scanning VM images vulnerabilities ;

Performance overhead due to running filters and maintenance services on VM images.

Physical Security

Implementation

% Systems deployed with implementation of :

- physical security perimeters ;

- monitoring, controlling and isolating data storage and process service points from unauthorized access ;
- role-based access control systems ;
- monitoring of environmental conditions that could affect computer systems.

Data and Computing Integrity

Implementation

% Applications that use check-sums or PDP techniques for data validation ;
% Applications deployed with computing replication and integrity checking techniques ;
% Data and VM images encrypted during transport across networks and hypervisor instances ;
% Applications deployed with implementation of data loss/leakage prevention techniques.

Impact

Computation time by PDP techniques ; Bandwidth overhead for tokens transmission by PDP techniques ; Processing overhead of accountable computation ; False positives of computational faults ; Task computation and communication time of remote computation verification.

Data Availability

Implementation

% Applications deployed with :
- replication of data between Cloud nodes ;
- implementation of Hardware independent restore and recovery capabilities ;
- implementation of software/provider independent restore and recovery capabilities ;
- multi-failure disaster recovery capability ;
- capability of infrastructure service fail-over to other providers.

Service Availability

Implementation

% Systems deployed with :
- implementation of security incident response plan ;
- configuration of SIEM incident reporting, analysis and alerting ;
- implementation of regular testing procedures for business continuity plans ;
- implementation of security mechanisms and redundancies to protect equipments from utility service outages (e.g., power failures, network disruptions, etc.).

Effectiveness

Mean-time of incident discovery ; Mean-time of incident recovery ; Recovery time of failed tasks ; Cloud failure rate ; FRC and DDoS attack detection time ; Time to identify attack source ; FRC and DDoS attack detection rate ; Detection False Positives : percent of legitimate traffic identified as attacks ; Detection False Negatives : percent of attacks not detected ; Percent of attacks incorrectly classified.

Impact

Mean-cost of incident recovery ; Mean-cost of loss caused by incidents ; Application offline-time due to incidents ; Financial cost of fault tolerance ; Data replication financial cost ; Processing overhead of detection mechanisms ; Application response time under attack.

Security Auditing and Testing

Implementation

% Systems deployed with :

- regular network penetration tests of Cloud service infrastructure ;
- regular internal and external auditing and risk assessments ;
- regular (network/application/OS) layers vulnerability scans ;
- implementation of patch management plans.
Effectiveness
Frequency of network penetration tests ; Percentage of endpoints covered ; Reliability level of audit plan ; Frequency of auditing ; Coverage of auditing plan ; Frequency of risk assessments ; Risk assessment coverage ; Frequency of vulnerability scans ; Mean-time to mitigate vulnerability ; Mean-time to patch vulnerability ; Mean-time to complete configuration changes.
Impact
Mean-cost to mitigate or patch vulnerability.
Compliance with Regulatory and Industrial Standards
Implementation
% Applications compliant with ISO, SOX and HIPAA standards ;
% Applications deployed with Security-SLA auditing systems.
Effectiveness
Number of violations/year ; Level of violations of Security-SLA.

4.6 Evaluation of Cloud Security Services

The evaluation process of Cloud security services starts by forming an evaluation matrix for each security service using the metrics developed in the previous section. Then, scaling of metrics is performed in order to compute a relative evaluation matrix for each security service. Weighting of metrics is necessary during the evaluation process since each metric can have a special effect on the evaluation results. At the end, a relative evaluation vector for each security service is computed. The following steps describe the evaluation process in details.

Step 1. Computing Relative Evaluation Matrices

At the beginning of the evaluation process, we construct an Evaluation Matrix (EM) for each Cloud Security Service (CSS), which contains the values of metrics for each CSP under evaluation. The matrix has the following form :

$$EM_{CSS} = \begin{pmatrix} m_{11} & \cdots & m_{1M} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NM} \end{pmatrix} \tag{4.1}$$

where M is the number of metrics that evaluate the service, and N is the number of CSPs under evaluation. Every element $m_{ij},\ (i = 1, 2, \ldots, N;\ j = 1, 2, \ldots, M)$ of the matrix repre-

sents the value of metric j for CSP i.
Qualitative metrics, if there are any, are quantified using special techniques or mapping functions. Then, scaling of metrics is performed to facilitate the evaluation process since metrics (collected raw data) have different units of measurement. Therefore, values of metrics are transformed to be relative to the ideal metric value. A metric is considered positive when a higher value of it indicates higher security (e.g., attack detection rate), and negative when a higher value of it reflects lower security (e.g., mean-time to discover attack). The Relative Evaluation Matrix (REM) for each security service is then computed by applying the following formulas on the elements of the matrix EM :

$$rm_{ij} = \begin{cases} \frac{m_{ij} - min_j}{max_j - min_j} & \text{if metric j is positive} \\ \frac{max_j - m_{ij}}{max_j - min_j} & \text{if metric j is negative} \\ 1 & \text{if } max_j = min_j \end{cases} \tag{4.2}$$

where max_j and min_j are respectively the maximum and minimum values of metric j in the matrix EM_{CSS}, and rm_{ij} represents the relative value of metric j for CSP i.
This matrix has the following form :

$$REM_{CSS} = \begin{pmatrix} rm_{11} & \cdots & rm_{1M} \\ \vdots & \ddots & \vdots \\ rm_{N1} & \cdots & rm_{NM} \end{pmatrix} \tag{4.3}$$

Step 2. Computing Metrics' Weights
In order to reflect their relative importance to the evaluation process, weights are assigned to each of the metrics. After scaling the metrics' values, their objective weights are generated using the standard deviations of normalized values. Most of the state-of-art evaluation techniques involved the use of subjective weights, which makes the evaluation more difficult to automate. In contrast, our method adds transparency and objectivity to the evaluation process since it requires no human intervention and depends rather on collected data, which helps in implementing the automation of the evaluation process. An alternative weight computing technique consists in using the Coefficient of Variation (CV) [71] of non-normalized metrics values. However, in our case, since all the values are relative and belong to the interval [0,1], a comparison based on the standard deviation should be sufficient. Therefore,

objective weights are calculated using the following formula :

$$W_j^{obj} = \frac{\sigma_j}{\sum_{j=1}^{m} \sigma_j} \tag{4.4}$$

where W_j^{obj} and σ_j are respectively the objective weight and standard deviation of metric j.

Step 3. Computing Relative Evaluation Vectors

After computing the evaluation matrix for each Cloud security service, we compute the Relative Evaluation Vectors (REV) by combining evaluation matrices and weight vectors.

Table 4.2 Service availability evaluation metrics for three different CSPs.

Evaluation Metrics	CSP1	CSP2	CSP3
% applications deployed with implementation of security incident response plan	100	80	70
% applications deployed with configuration of SIEM incident reporting	89	72	60
% systems deployed with implementation of regular testing procedures	90	80	66
% systems deployed with implementation of redundancies	79	74	80
Mean-time of incident discovery (hours)	1	1.2	1.3
Mean-time of incident recovery (hours)	0.9	1	2
Recovery time of failed tasks (seconds)	120	80	300
Cloud failure rate (%)	2	1	3
FRC and DDoS attack detection time (seconds)	82	62	44
Time to identify attack source (seconds)	100	120	160
FRC and DDoS attack detection rate (%)	99	97	95
Detection False Positives	1	3	5
Detection False Negatives	5	6	10
Percent of attacks incorrectly classified	2	4	5
Mean-cost of incident recovery (x1000$)	9	12	20
Mean-cost of loss caused by incidents (x1000$)	3	5	7
Application offline-time due to incidents (hours)	2	3	5
Financial cost of fault tolerance (x10,000$)	40	30	40
Data replication financial cost (x10,000$)	3	4	5
Processing overhead of attack detection mechanisms (seconds)	20	30	25
Application response time under attack (seconds)	5	4	10

The REV of each security service is obtained using the following formula :

$$REV_{CSS} = REM_{CSS} \otimes W^{obj} \tag{4.5}$$

where W^{obj} is the vector of weights of the metrics that evaluate the Cloud security service (CSS).
The vector REV_{CSS} represents the relative evaluation of CSPs with respect to the Cloud security service (CSS). CSPs can visualize the evaluation results through performance diagrams in order to process a security self-evaluation with respects to each other as explained in the following section.

4.7 Case Study : Cloud IaaS Providers

Due to the limited access to CSPs' security information, the proposed evaluation methodology is tested on a given Cloud scenario. The objective of this case study is to show that : (1) the proposed set of evaluation metrics is collectible, (2) the examination of evaluation results can effectively lead to the objective of the evaluation methodology, and (3) the evaluation methodology can be applied without introducing a considerable overhead, in case of a future integration into an application deployment framework.

Cloud scenario

This case study considers a scenario of three Cloud IaaS providers, CSP1, CSP2 and CSP3, that apply different levels of security services within their Cloud infrastructures. Table 4.2 shows the values of the evaluation metrics related to service availability for each of the three CSPs (this example shows the evaluation metrics related to only one security service due to space limitation).

Simulation

We implemented our methodology in MATLAB. The program takes the values of metrics related to all security services as input, forms the evaluation and relative evaluation matrices using equation 4.3, computes the weight of each metric using equation 4.4, and finally outputs a relative performance diagram as the one shown in Figure 4.3 by applying equation 4.5.

Results analysis

By examining the output diagram of evaluation results in Figure 4.3 , several points can be concluded :

— CSP3 suffers from low performance at the level of virtualization security relatively to the other CSPs. CSP3 should implement suitable security solutions to protect VM images and the VM migration process.

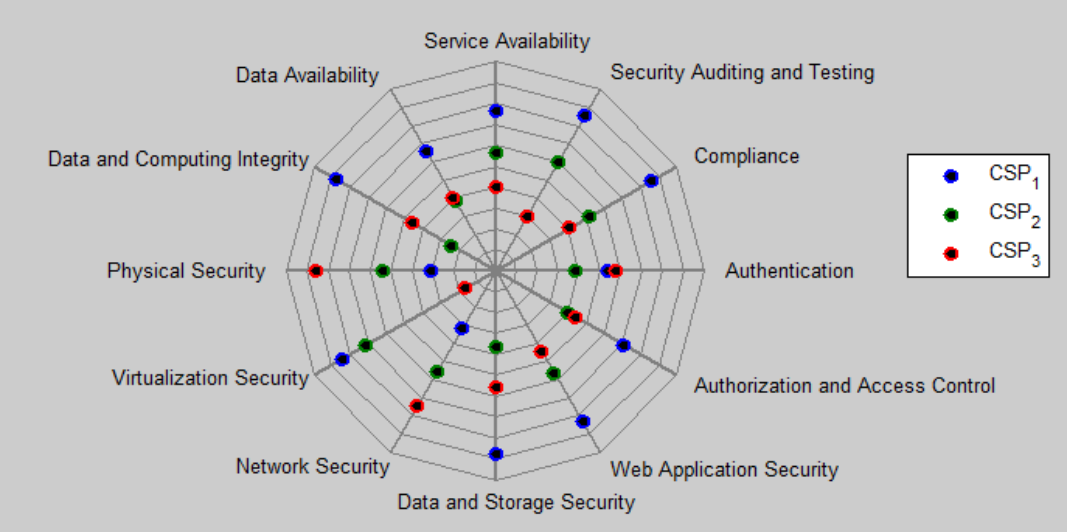


Figure 4.3 Evaluation results of the case study.

— CSP1 and CSP2 outperform CSP3 in terms of security auditing and testing, and compliance to security standards. In order to attract Cloud customers, CSP3 should invest additional efforts in implementing security audit and assessment plans, which will help her raise the level of her security compliance.

— At the integrity level, the performance of CSP2 is too low compared to the other CSPs. CSP2 needs to implement appropriate and efficient data integrity protection techniques such as PDP.

Figure 4.4 shows the computational time of the proposed evaluation methodology for different numbers of CSPs. The computational time is relatively small even for a high number of CSPs. We can conclude that implementing this methodology for security evaluation purposes in a Cloud environment and integrating it into service selection and resource allocation frameworks will introduce negligible overhead to the involved optimization problems. CSPs can easily take advantage of this methodology for self-evaluation and performance improvement purposes. This evaluation is fully objective and depends only on the information and measurements collected from CSPs' security services. This methodology can be deployed on a third-party's side (e.g., Cloud broker) that can handle the security data collection process and provide the CSPs and also Cloud customers with the evaluation results.

4.8 Conclusion

Cloud Computing technology is becoming increasingly popular in the IT and business domains due to its numerous technical and financial advantages. It allows companies to take

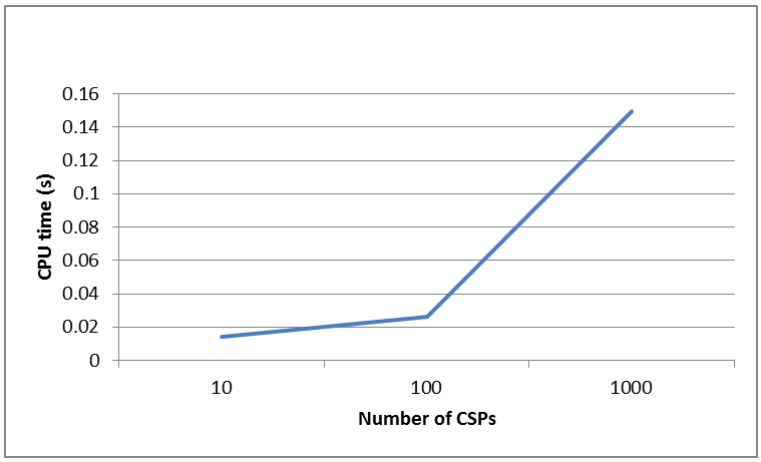


Figure 4.4 Computational time of the evaluation methodology.

advantage of the various types of offered services such as computing and storage in order to deploy very reliable and efficient applications with lower costs and higher revenues. However, the Cloud adoption process is still relatively slow due to security reasons. CSPs should continuously assess the level of their security services and perform a relative self-evaluation in order to identify their limitations and improve their performance. In this paper, we proposed an evaluation methodology of Cloud security services that can help CSPs to process a security self-evaluation and situate the level of their security services within the Cloud market. This methodology is based on objective evaluations and can be easily automated on a Cloud broker's side. We also provided an example to show the applicability of the proposed methodology and the effectiveness of the evaluation results.

The developed set of evaluation metrics can also be considered as a primary step towards the standardization of Cloud security evaluation due to their diversity and significance. Standard and common testbeds and practices are also required to implement a successful Cloud security measurement and monitoring platform [127]. In our future work, we will try to apply the proposed evaluation methodology on a set of real cloud security data which we will collect from existing Cloud infrastructures by collaborating with major Cloud vendors. We also plan to develop a methodology that could help Cloud customers in optimizing the selection of Cloud security services according to their security requirements.

CHAPTER 5 ARTICLE 2 : A BROKER-BASED FRAMEWORK FOR STANDARDIZATION AND MANAGEMENT OF CLOUD SECURITY-SLAS

Talal Halabi and Martine Bellaiche
Computers and Security, vol. 75, pp. 59-71, 2017.

Abstract

Security is still one of the main barriers discouraging companies and businesses which deal with sensitive information and confidential data from migrating toward the Cloud. Recent efforts have tried to specify the security level of the Cloud service with the help of Security Service Level Agreements (Security-SLA). However, Security-SLAs in their current form and with their present terms are not fully measurable and are hard to monitor. Quantification and standardization of Security-SLAs will surely speed up the Cloud adoption process and attract more customers to benefit from the advantages of Cloud computing in a more confident and secure fashion. In this paper, we propose a broker-based framework that manages the Cloud Security-SLA. We first develop a standard, quantitative, and measurable form to represent the agreement. Then we propose an evaluation and selection model that is fundamentally based on computing the adequate trade-off between the security CIA triad attributes (Confidentiality, Integrity, and Availability) in the context of a multi-objective optimization problem. Simulation results show the set of Pareto-optimal solutions and how the customer can select the most suitable service provider using higher level information that is related to the nature of the service and financial cost.

5.1 Introduction

Cloud computing technology is becoming increasingly popular due to its many economic and technological benefits. The Cloud market has witnessed a growth rate of approximately 20 percent in 2017 and is expecting to keep a stable growing pace in the next few years [128]. Customers enjoy many interesting features such as scalability, resilience, performance, portability, on-demand and measured service when provisioning their services in the Cloud. The Cloud computing architecture consists of three layers : (i) Software-as-a-Service (SaaS) which is run by the Cloud Service Providers (CSPs) and mostly used by organizations ; (ii) Platform-as-a-Service (PaaS) which is a tool provided to develop applications without installing any software on the developer's side ; and (iii) Infrastructure-as-a-Service (IaaS) which includes

storage, hardware, servers, and networking services operated, maintained, and controlled by the CSPs.

Although the use of Cloud services is rapidly growing, these services still represent a small portion of the global IT market. One of the main obstacles to the full migration to the Cloud is the lack of security. The Cloud introduces many security challenges and threats, especially to data privacy and service availability, which increases customers' concerns about security assurance and transparency of the CSPs. In the context of regular Service Level Agreements (SLAs) that usually address the Quality of Service (QoS) parameters, quantitative and measurable indicators are normally considered (e.g., response time). But when it comes to security, such indicators are hard to explore since quantifying the Cloud security is a lot more challenging. For instance, if a Virtual Machine (VM) is being attacked by its neighboring VM via a Side Channel, a specific metric that helps announcing the potential security threat and detecting the degradation in the quality of security on this VM does not exist.

The research on specification of security parameters in SLAs is currently very active. With security as an essential driver of the Cloud market, standardization and quantification of the Security-SLA template will surely speed up the Cloud adoption process. They will not only help guarantee security and transparency to users, but also provide the CSPs with the necessary information and tools to continuously monitor the quality of their provided security solutions in order to predict future violations and reduce the possible damages. On the other hand, a quantitative Security-SLA will facilitate the comparison of CSPs' security offerings and allow customers to choose the one that best suits their security requirements.

In this paper, we propose a broker-based framework to efficiently manage the offered Cloud Security-SLAs. Our contributions are described as follows :

- First, we propose a standard, quantitative, and measurable form to represent the Security-SLA contract, based on the evaluation of major Cloud-specific vulnerabilities and threats.
- Second, we model the Security-SLA evaluation and selection problem in the Cloud as a multi-objective optimization problem that is fundamentally based on computing the adequate trade-off between the three security CIA triad attributes (Confidentiality, Integrity, and Availability).
- Third, we propose to use the Pseudo-Weight method [101] to perform service selection based on security satisfaction.
- Finally, we propose a Security-SLA monitoring model and a violation prediction and remediation process that will help the CSPs in minimizing the damage caused by security breaches and incidents.

This paper is organized as follows. Section 5.2 discusses the literature review related to Security-SLAs in the Cloud. Section 5.3 describes the proposed evaluation and selection framework. In Section 5.4, the standard Security-SLA is developed. In Section 5.5, the evaluation and selection model is defined and an adequate solution is proposed. Section 5.6 presents the Security-SLA monitoring model. In Section 5.7, simulation is performed and results are analyzed. Finally, Section 5.8 concludes the paper.

5.2 Literature Review

Security quantification and evaluation is currently an active research topic. In our previous work [129, 130], we started to address the problem of security evaluation and service selection in the Cloud, but without considering customers' security requirements. The Cloud Security Alliance (CSA) [9] and NIST [116, 100] are continuously working on the standardization of Cloud security metrics and contracts through different projects (e.g., CCM [29]). The SPECS project [52] is also an example of such effort. Its main objective is to develop a framework that offers Security-as-a-Service in the Cloud based on the security parameters specified in the SLA, and provide the necessary techniques to manage its life cycle. Rak et al. [131] presented an approach for Security-SLA and Casola et al. [132] developed a Cloud application for Security-SLA evaluation based on the SPECS architecture. Eventually, our work in this paper on Cloud security quantification could be incorporated into the SPECS framework for more effective results.

Few researchers have addressed the management of the Security-SLA life cycle in the Cloud. However, their work either only focused on the security controls proposed by the CSA which are hard to quantify, or did not cover the whole set of security services that are normally offered or should be offered in a Cloud environment. For instance, Da Silva et al. [41, 65] developed an approach to build a Security-SLA based on a security metrics hierarchy that describes the security level in a Cloud computing environment. Mirkovic [40] proposed a model to measure Cloud security by defining a system of metrics based on the ISO 27001 standard security controls [32]. Nonetheless, these models did not cover all critical security aspects and services in the Cloud.

Bernsmed et al. [66] presented an approach to manage Security-SLAs in federated Cloud environments. Rojas et al. [64] presented a framework to orchestrate the security-SLA life cycle in Cloud computing. Na et al. [69] provided a broker-based Security-SLA evaluation model based on threat analysis and service type. Zhengwei et al. [71] constructed a quantifiable Cloud-oriented Security-SLA based on a metric modeling method. Taha et al. [67] used the Analytical Hierarchy Process (AHP) method to evaluate the Cloud security based

on the parameters provided by the CAIQ [9]. Trapero et al. [133] presented a Security-SLA monitoring framework to validate its fulfillment and detect violations. However, these proposals did not address in detail the evaluation metrics that can quantify and measure the security of CSPs. Moreover, one of the main contributions of our work over the previously proposed Security-SLA evaluation and selection frameworks is that our model considers the possible trade-off between the different security attributes, and which customers would need to evaluate during service selection.

5.3 The Cloud Security-SLA Life Cycle

Several proposals describing the SLA life cycle were found in the literature. Like the traditional SLA life cycle, the life cycle of Security-SLAs includes multiple phases each describing a specific role or responsibility held by either the service provider or customer. In the context of our framework, the life cycle of the Cloud Security-SLA involves four fundamental phases as shown in Figure 5.1.

— **Phase 1 : Definition and Specification**. In this phase, the security parameters and metrics to be included in the Security-SLA are defined. This involves the standard template design and publication by the CSPs. The Security Service Level Objectives (SSLOs) that describe the offered security level are also specified for later discovery by the Cloud customers. The standard version of the agreement that will be used in this phase is proposed in section 5.4.

— **Phase 2 : Evaluation and Negotiation**. In this phase, the security requirements are carefully engineered by the customers and provided to the Cloud broker to initiate the evaluation process. Selection and negotiation of the most suitable offered Security-SLA are then performed based on the adequate estimation of the necessary trade-off between the different security attributes. All these steps are explained in detail in section 5.5.

— **Phase 3 : Deployment and Execution**. In this phase, the required security services are deployed through the implementation of security mechanisms and procedures.

— **Phase 4 : Monitoring and Reporting**. During this phase, the Security-SLA is continuously monitored in real-time. This involves the reporting of security and performance levels, the prediction of contract violations, the management of corrective actions, and the implementation of incident response and remediation plans. The CSP also keeps track of any changes in the customer's security requirements and the deployment environment for possible renegotiation or termination. The monitoring and reporting phase is discussed in section 5.6.

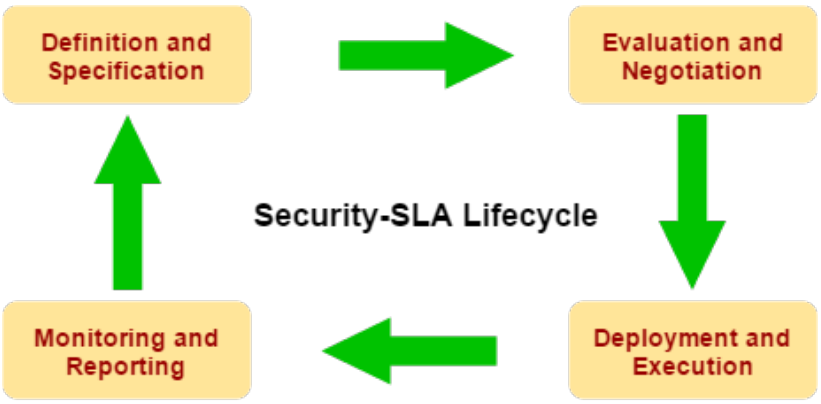


Figure 5.1 The Cloud Security-SLA life cycle in the context of the proposed framework.

In this paper, the focus is essentially oriented toward the phases 1, 2 and 4 of the life cycle. In our proposed framework, the first phase is emphasized by building a standard version of the Security-SLA for Cloud computing, that is based on a deep analysis of the security services which are usually deployed to protect the data and infrastructure from the threats and incidents associated to the Cloud paradigm. The term standard refers to the fact that the agreement will always consist of the same metrics which will facilitate the evaluation and comparison of different security offerings. For the evaluation and negotiation phase, we propose a selection model that is based on computing the adequate trade-off between the three fundamental security attributes : confidentiality, integrity, and availability, since no security offering will perfectly align with the customer's security requirements. The evaluation and negotiation are performed with the help of a third party (e.g., Cloud broker). Finally, the developed metrics and measurements will be used to monitor the fulfillment of the contract, which is accomplished during phase 4 of the life cycle.

5.4 The Standard Cloud Security-SLA

Standardization of the Cloud Security-SLA will allow an easy and automated evaluation of the service providers' security offerings and help the Cloud customers in conducting a more guided decision making and selection process. We propose in this paper a novel approach to construct the standard Security-SLA format based on the security services that are normally provided or should be provided in a Cloud environment. We consider the protection of the fundamental security attributes defined by the CIA triad security model (Confidentiality, Integrity, and Availability) as the main objective behind this security agreement. These

attributes are defined as follows :

- Confidentiality, which concerns with protecting the customer's sensitive information and confidential data from unauthorized disclosure by implementing adequate Identity and Access Management (IAM) solutions using a combination of different authentication and access control techniques (e.g., XACML [111]).
- Integrity, which consists on protecting the accuracy and validity of data and computations using appropriate cryptography techniques and efficient data loss and manipulation prevention methods (e.g., Provable Data Possession [121]).
- Availability, which describes the ability of the service and data to be accessible whenever required by the customers, and is usually maintained by implementing effective incident management and data backup and recovery plans.

Vulnerability and risk assessment plans and efficient security auditing procedures should also be implemented to protect each of the three security attributes from the Cloud's potential threats and attacks. Some of these threats are presented in Table 5.1 and more information about them can be found in [6] and [8]. We use the STRIDE methodology [99] to categorize the Cloud threats, which is a threat classification model developed by Microsoft in order to reason about the possible threats to a computer system. The six threat categories found in the STRIDE model are described in the following :

- Spoofing, which involves stealing other users' identities and authentication information, and illegally using them to access confidential data.
- Tampering, which consists on maliciously manipulating and modifying other users' data.
- Repudiation, which is the ability of users to perform certain actions without other parties (e.g., the Cloud administrator) being able to prove or trace these actions.
- Information disclosure, which involves the exposure of the user's confidential information to other users or even to a malicious Cloud administrator who are not allowed to have access to it.
- Denial of service, which is a potential threat to Cloud computing since all users usually share the same infrastructure and resources. This threat, when translated into an attack, will essentially target the service availability and try to saturate the Cloud resources to make them unresponsive to the requests of legitimate users.
- Elevation of privileges, which is when unprivileged users or Cloud internals gain unauthorized access privileges.

In order to protect the Cloud security attributes from the above described threat categories, several security services are normally supplied and deployed by the service provider. We chose to make these security services the main foundation for the construction of the standard Cloud

Table 5.1 Some of the threats to the Cloud security attributes.

Security attribute	Threats				
	Threat description	Vulnerability	STRIDE classes	Source	Damaged assets
Confidentiality	Cross-VM attack via Side Channels	VM co-residence	S-I-E	Neighboring VM	Data
	Malicious SysAdmin	Loss of physical control over data	S-T-I-E	Malicious insider	Data
Integrity	Data loss or manipulation	Loss of physical control over data	T	Cloud server ; server administrator	Data
	Dishonest computation in remote servers	Outsourced computation	T	Cloud server	Computations
Availability	Direct and indirect DoS attacks	VM co-residence ; bandwidth under-provisioning	D	Malicious user ; neighboring VM	Service or Data
	Economic Denial of Sustainability (EDoS) attack	Cloud pricing model	D	Malicious user	Service

Table 5.2 Cloud security services, related threat classes, and protected security attributes.

Security service	STRIDE threat class						Protected security attribute		
	S	T	R	I	D	E	Confidentiality	Integrity	Availability
Authentication	●	●	●	●			✓	✓	
Authorization and access control		●	●	●		●	✓	✓	
Web application security	●		●	●	●		✓		✓
Network security	●			●	●		✓		✓
Data and storage security		●		●			✓	✓	✓
Virtualization security	●	●		●	●	●	✓	✓	✓
Physical security	●	●		●		●	✓	✓	
Data and computational integrity		●	●					✓	
Data and service availability					●				✓

Security-SLA, since this will grant us the possibility of quantifying their performances and measure the level and quality of the provided security, and compare it between different CSPs. This quantification aspect was missing in the state-of-the-art research, especially because the authors were always focusing on the set of security controls identified by the CSA and were not looking into the Cloud security aspect from a deeper perspective. These security services are presented in Table 5.2, along with their relationship to the security threats and attributes. Detailed explanation of the Cloud security services is provided in our previous work [129].
A set of well-defined and measurable security metrics was then developed in the context of our standardization and evaluation framework, to form the essence of the quantifiable Cloud Security-SLA. To allow the latter to be controllable, the CSP must be able to exercise full control over the values of these metrics, which determine the security level of the provided service. Three types of these metrics can be distinguished according to their role as follows :

— Positive metric (denoted by P), a real number or percentage, usually expressing a

Table 5.3 A set of evaluation metrics related to each Cloud security service.

Security service	Security mechanisms and techniques	Evaluation metrics	Metric type
Authentication	- Multi-factor authentication - Federated IDs (e.g., using LDAP or OAuth) - Single Sign On (SSO)	- False Acceptance Rate - False Rejection Rate - Average response time of transactions - Enforcement of policies on password strength and expiration, and blocking of invalid login attempts	N N N I
Authorization and access control	- Identity management - Authorization policies - XACML access control - Auditing and logging	- Implementation of risk-based entitlement decisions - Ability to use temporary access credentials - Frequency of system users and administrators' entitlements periodic review - Frequency of review of access control logs and accounts activity	I I P P
Web application security	- Web application scanners - IDS/IPS - Malware detection	- Intrusion detection success rate - Intrusion detection false positives - Injection attacks detection success rate - Testing of newly emerged APIs/interfaces for application deployment	P N P I
Network security	- IDS/IPS - DDoS mitigation - IPSec VPNs deployment - SSL protection - Firewalls	- Configuration of security groups and Access Control Lists (ACLs) on virtual interface ports - Mean-time to discover and mitigate attack - Latency or response time during attack - Packet drop rate during attack - CPU load or processing usage of IDS - Network load or bandwidth overhead of IDS - Frequency of network penetration tests	I N N N N N N
Data and storage security	- Secure storage schemes - Encryption techniques - Key management procedures - Data dispersion techniques - Data loss/leakage prevention techniques - Secure data migration - Breach reporting and recovery	- Internal storage of encryption keys - Capability of creation of a unique encryption key per tenant - Capability of open encryption methodologies - Enabling of HTTP Strict Transport Security (HSTS) - Support of secure data deletion - Database deployment with SSL protected transactions - Percentage of communication overhead or online latency - Percentage of key storage overhead - Storage node latency in responding to read-write requests	I I I I I I N N N
Virtualization security	- VMs' interference prevention - Hypervisor-level role-based access control - VM encryption - SSH secure communications	- Events monitoring and auditing by hypervisor - Capability of encryption of virtual storage - Validation of VMs - Implementation of encrypted live migration - Data destruction procedures after migration - Regular assessment of virtualization vulnerabilities - VM backup, restoration and clean-up capabilities - Slowdown in migration time due to secure migration - Time spent scanning VM images' vulnerabilities - Performance overhead due to running filters and maintenance services on VM images	I I I I I I I N N N
Physical security	- Physical security perimeters - Role-based access control systems	- Monitoring, controlling and isolating data storage and process service points - Monitoring of environmental conditions that could affect computer systems	I I
Data and computational integrity	- PDP techniques for data validation - Computing integrity checking techniques	- Deployment of applications with computing replication - Encryption of data and VM images during transport across hypervisor instances - Implementation of data loss/leakage prevention techniques - Computation time by PDP techniques - Bandwidth overhead of PDP tokens' transmission - Processing overhead of accountable computation - False positives of computational faults	I I I N N N N
Data and service availability	- Incident response plans - Data replication between Cloud nodes - DDoS attack detection system - Risk assessment plans - Vulnerability scan	- Multi-failure disaster recovery capability - Capability of infrastructure service fail-over to other providers - Configuration of SIEM incident reporting, analysis and alerting - Regular testing procedures for business continuity plans - Mean-time of incident discovery and recovery - Recovery time of failed tasks - Cloud failure rate - FRC and DDoS attack detection time - Time to identify attack source - Processing overhead of detection mechanisms - Application response time under attack - Mean-time to mitigate or patch vulnerability	I I I I N N N N N N N N

Table 5.4 An example of a metric collection template.

Author	Transmission date	Metric title	SSLO	Metric description
Unit of measure (e.g., time, %, frequency, etc.)		Objective : why is this metric useful ?		Metric source of information
Evaluation usage	Frequency or measurement interval		Metric calculation (e.g., data, formula, etc.)	

benefit aspect of security. A higher value of this metric normally indicates higher level of security (e.g., attack detection rate).

— Negative metric (denoted by N), a real number or percentage, usually expressing a cost aspect of security. A higher value of this metric normally indicates lower level of security (e.g., mean-time to discover attack).

— Implementation metric (denoted by I), a Boolean 0 or 1 that indicates if a specific security policy or mechanism is implemented or absent.

Some of the developed metrics are presented in Table 5.3, along with the security mechanisms and solutions they usually intend to evaluate. Detailed analysis of the Cloud network security evaluation metrics was also provided in our previous work [134]. The offered value of each metric by the CSP, as well as the required one by the Cloud customer, is called the Security Service Level Objective (SSLO). These SSLOs reflect the level of provided security. For positive metrics, SSLOs could express the maximum performance that could be supplied by the security solution, whereas for negative metrics, they could be used to set the error margins to which CSPs should adhere. The developed metrics could be collected from the CSPs by the Cloud broker through a special template like the one presented in Table 5.4. The next section discusses in detail the Security-SLA evaluation process based on the SSLO values.

5.5 The Proposed Evaluation and Selection Model

There is no security offering in the Cloud that will guarantee a definite satisfaction of all the security requirements of the customer, without the need to make a preponderance of some of her needs. For instance, if the customer's data are extremely sensitive that they will need protection from unauthorized disclosure at all costs, but their availability is relatively less important, the customer will try to find the CSP that could provide the most secure storage scheme even if she finds that the CSP's plan concerning network security and protection against Denial of Service (DoS) attacks is not completely satisfying for her requirements. This is an example of a case where the customer has to make a trade-off or a compromise between confidentiality and availability during the Cloud service evaluation and selection process.

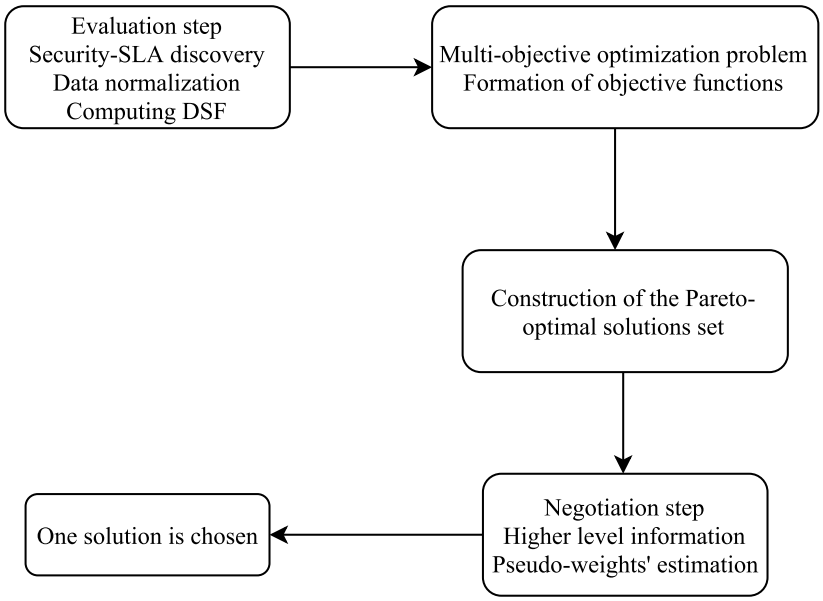


Figure 5.2 The Security-SLA evaluation and selection process.

For this reason, the selection model in our proposed framework is based on a multi-objective optimization problem, where multiple aspects of security (sometimes conflicting) need to be optimized. These aspects are represented by the three CIA attributes : confidentiality, integrity, and availability. The optimization process is preceded by an evaluation phase where the Security-SLAs of multiple CSPs are discovered by the Cloud broker following a customer service request. The evaluation results which are computed based on the customer's requirements are the input to the optimization process. The model then outputs the set of possible service providers that can qualify to satisfy the customer's security requirements. The final step consists on determining the most suitable CSP by computing the adequate trade-off between the different security aspects. The whole process is illustrated in Figure 5.2 and is described in detail in the following subsections.

5.5.1 The Evaluation Step

The quantitative and standard Security-SLA that we proposed in this paper permits an efficient and automated evaluation of CSPs' security offerings. This evaluation is fundamentally based on the computation of the Euclidean distance between security offerings and requests. In Cartesian coordinates, if $P = (p_1, ..., p_m)$ and $P' = (p'_1, ..., p'_m)$ are two points in the Euclidean m-dimensional space, then the distance between points P and P' is given by the

following formula :

$$d(P, P') = \sqrt{\sum_{i=1}^{m}(p_i - p'_i)} \tag{5.1}$$

In this phase, data related to the security metrics in the Security-SLA are collected from the CSPs and stored on the Cloud broker side. We introduce the Dissatisfaction Factor parameter (DSF) which will be computed for each of the security attributes. Basically, this parameter represents the distance between the requested and offered values of the metrics that evaluate each security attribute. The following notations are used in the evaluation and selection model :

- $CSP = \{CSP_n \mid 1 \leq n \leq N\}$ is a set of N Cloud service providers under evaluation.
- $Co = \{Co_c \mid 1 \leq c \leq C\}$ is a set of C metrics that evaluate the confidentiality attribute. P^{Co}, N^{Co}, *and* I^{Co} are subsets of Co that respectively contain the positive, negative, and implementation metrics that evaluate the confidentiality attribute, with $Co = P^{Co} \cup N^{Co} \cup I^{Co}$.
- $In = \{In_i \mid 1 \leq i \leq I\}$ is a set of I metrics that evaluate the integrity attribute. P^{In}, N^{In}, *and* I^{In} are subsets of In that respectively contain the positive, negative, and implementation metrics that evaluate the integrity attribute, with $In = P^{In} \cup N^{In} \cup I^{In}$.
- $Av = \{Av_a \mid 1 \leq a \leq A\}$ is a set of A metrics that evaluate the availability attribute. P^{Av}, N^{Av}, *and* I^{Av} are subsets of Av that respectively contain the positive, negative, and implementation metrics that evaluate the availability attribute, with $Av = P^{Av} \cup N^{Av} \cup I^{Av}$.
- $R^{Co} = (R_1^{Co}, \ldots, R_C^{Co})$ is the vector containing the requested values (by the customer) of the metrics in the set Co and $O_n^{Co} = (O_{n,1}^{Co}, \ldots, O_{n,C}^{Co})$ is the vector containing the offered values (by CSP_n) of the same metrics. The same applies with the metrics that evaluate the integrity and availability attributes.
- DSF_n^{Co}, DSF_n^{In}, *and* DSF_n^{Av} denote the dissatisfaction factors of CSP_n related to confidentiality, integrity, and availability respectively.

During evaluation, normalization of the metrics' values (the quantitative ones) is performed since they have different units of measurement. To this end, the following scaling formula is

applied to the offered values according to the metric category :

$$\hat{O}_{n,c}^{Co} = \begin{cases} \frac{O_{n,c}^{Co} - min_c^{Co}}{max_c^{Co} - min_c^{Co}} & \forall Co_c \in P^{Co} \\ \frac{max_c^{Co} - O_{n,c}^{Co}}{max_c^{Co} - min_c^{Co}} & \forall Co_c \in N^{Co} \\ 1 & \text{if } max_c^{Co} = min_c^{Co} \end{cases} \tag{5.2}$$

where $min_c^{Co} = \min_{1 \leq n \leq N} \{O_{n,c}^{Co}\}$, $max_c^{Co} = \max_{1 \leq n \leq N} \{O_{n,c}^{Co}\}$, and $\hat{O}_{n,c}^{Co}$ is the metric value normalized to the interval [0,1]. The same applies to the requested metric value R_c^{Co} to obtain the normalized value $\hat{R}_c^{Co}$.

The process of computing the dissatisfaction factors of all CSPs is presented in Algorithm 1. We only describe the computing steps of the DSF related to confidentiality since the same process is applied to the other security attributes. Since some of the security services are deployed to protect more than only one attribute as seen in Table 5.2, some of the metrics could be used to evaluate multiple attributes simultaneously. Therefore, we assume that $Co \cap In \neq \emptyset$, $Co \cap Av \neq \emptyset$, and $In \cap Av \neq \emptyset$. In our algorithm, we assume that if the provided security level is higher than the required one, the metric is not considered in the DSF computing process, and the customer will be satisfied with that level unless other types of cost may apply. This applies to all three metric types (positive, negative, and implementation). The dissatisfaction of positive metrics is computed in lines 3 to 8 in the algorithm, of the negative metrics in lines 10 to 15, and of the implementation metrics in lines 17 to 22. The total dissatisfaction value is computed in line 23 of the algorithm.

5.5.2 The Multi-Objective Optimization Problem

Since the three security attributes need to be optimized altogether during the selection process, we decide to model our selection problem as a multi-objective optimization problem. Multi-objective optimization is an area in multi-criteria decision analysis where two or more conflicting objectives need to be optimized simultaneously. We first define the following three objective functions that express for all CSPs involved in the selection process, the distance between required and offered security values with respect to confidentiality, integrity, and availability respectively :

$$f^{Co} = \sum_{n=1}^{N} DSF_n^{Co} x_n \tag{5.3}$$

$$f^{In} = \sum_{n=1}^{N} DSF_n^{In} x_n \tag{5.4}$$

$$f^{Av} = \sum_{n=1}^{N} DSF_n^{Av} x_n \tag{5.5}$$

where $x_n = \{0, 1\},\ n = 1, \ldots, N$, is a binary variable that indicates the assignment of the service to the provider $CSP_n \in CSP$.

Two approaches to multi-objective optimization exist : the preference-based procedure and the ideal procedure [135]. In the former, objective functions are combined to form a single

---

Algorithm 1 DSF^{Co} Computing Algorithm

Input:
- CSPs' offered values and customer's required values of the metrics in the set Co

Output:
- Confidentiality Dissatisfaction Factor DSF^{Co} for each CSP

for all provider $CSP_n \in CSP$ **do**
  $d_P = 0; d_N = 0; d_I = 0, d = 0.$
  for all metric $Co_c \in P^{Co}$ **do**
    if $\hat{R}_c^{Co} > \hat{O}_{n,c}^{Co}$ **then**
      $d = d + (\hat{R}_c^{Co} - \hat{O}_{n,c}^{Co})^2$
    end if
  end for
  $d_P = \sqrt{d}$
  $d = 0$
  for all metric $Co_c \in N^{Co}$ **do**
    if $\hat{R}_c^{Co} < \hat{O}_{n,c}^{Co}$ **then**
      $d = d + (\hat{O}_{n,c}^{Co} - \hat{R}_c^{Co})^2$
    end if
  end for
  $d_N = \sqrt{d}$
  $d = 0$
  for all metric $Co_c \in I^{Co}$ **do**
    if $R_c^{Co} = 1$ **then**
      $d = d + |R_c^{Co} - O_{n,c}^{Co}|$
    end if
  end for
  $d_I = \frac{d}{|I^{Co}|}$
  $DSF_n^{Co} = d_P + d_N + d_I$
end for

function by computing a weighted sum. This necessitates that the weights of the objectives should be known in advance. With the ideal procedure, a set of feasible trade-off solutions is generated and one is eventually selected based on higher level information. In our model, we decide to follow the second approach which will allow the customer to form a global vision of the solution set and possess full control over her decision during the selection process. Therefore, the Security-SLA selection problem is formulated as a linear multi-objective optimization problem that aims at minimizing the dissatisfaction factors related to all security attributes as follows :

$$\min \{f^{Co}, f^{In}, f^{Av}\} \tag{5.6}$$

Subject to :

$$\sum_{n=1}^{N} x_n = 1 \tag{5.7}$$

where Constraint 5.7 ensures that the service will be eventually assigned to only one CSP.

5.5.3 The Proposed Solution

In our model, the set of candidate solutions (number of CSPs involved in the evaluation) is relatively very finite. Therefore, we decided to perform an exhaustive search on the solution space to solve the multi-objective optimization problem and generate the set of all non-dominated solution points. A more sophisticated heuristic optimization approach could be adopted in the case where a large number of service requests is received by the broker. The set of non-dominated solutions in our case is constructed based on the following definition. A feasible solution $x = (x_1, x_2, \ldots, x_N)$ is considered efficient if there is not another solution $y = (y_1, y_2, \ldots, y_N)$ such that $f^q(y) \leq f^q(x)$ for all $q \in \{Co, In, Av\}$ and $f^q(y) < f^q(x)$ for at least one $q \in \{Co, In, Av\}$. In other words, if solution y does exist, both solutions x and y are considered to be equally good, since in both cases, none of the objective functions can be bettered without worsening some of the others. The set of all efficient points is called the efficient frontier or Pareto-front, and the best solution to the problem will be selected from this set.

To find the most suitable solution to the customer's request, the latter needs to evaluate a specific trade-off among the three objective functions based on their relative importance. To this end, we propose to use the Pseudo-Weight method [101], where a weight vector for each

solution on the Pareto-front can be derived using the following formula :

$$w_x^i = \frac{(f_{max}^i - f^i(x))/(f_{max}^i - f_{min}^i)}{\sum_{q \in \{Co,In,Av\}} (f_{max}^q - f^q(x))/(f_{max}^q - f_{min}^q)} \tag{5.8}$$

where w_x^i is the weight of the $i-$th objective function for the non-dominated solution x, and f_{max}^i and f_{min}^i $(i \in \{Co, In, Av\})$ are respectively the maximum and minimum values of the same function. The sum of all weights is equal to 1. For instance, if the customer has a pre-estimated weight vector for the three security attributes $w = (w^{Co}, w^{In}, w^{Av})$ during the selection process, the customer should choose the solution that has the closest pseudo-weight vector to the required one. The quantitative and qualitative factors that could help the customer in determining the adequate weight vector for the requested service are presented in Figure 5.3. This higher level information could guide the customers during the evaluation and selection process. For example, a customer might prioritize availability over integrity if the requested service is a web server or if the Cloud service delivery model is of type SaaS. Whereas, in the case where the customer is indifferent to all security attributes but mostly interested in the cost of security implementation, the customer would prioritize the security attributes in increasing order of their financial cost.

5.6 Cloud Security-SLA Monitoring

Once the appropriate Security-SLA has been selected during the negotiation phase, it will be enforced and implemented onto the Cloud, where special resources will be allocated for the execution of specific security mechanisms and the deployment of required security features. This is phase 3 of the Security-SLA life cycle, and which could be automated using special

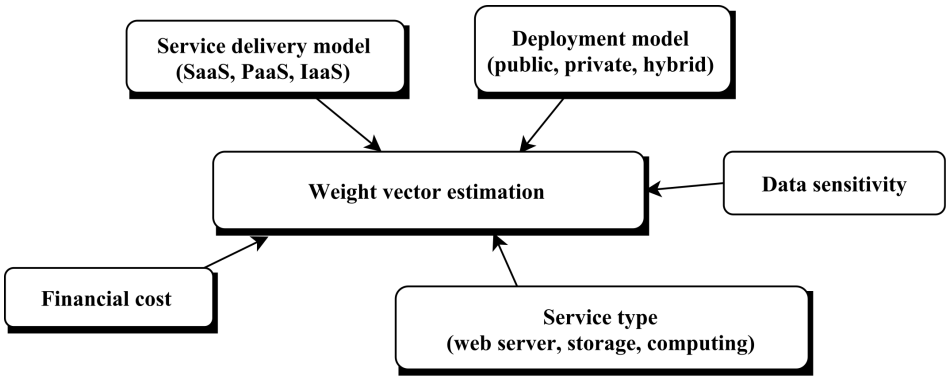


Figure 5.3 The set of factors that influence the estimation of security attributes' weights.

Cloud automation tools. The quantifiable format of the Cloud Security-SLA proposed in this paper will facilitate the monitoring and assessment of its state. Phase 4 of the Security-SLA life cycle involves the continuous monitoring of the negotiated security level in real-time and the management of the different states of the contract. These states and the transitions between them are illustrated in Figure 5.4. The Security-SLA is usually in the satisfaction state if the monitoring and reporting process indicates that the security level agreed upon with the customer is still conserved with no considerable fluctuations detected. Monitoring data normally include access log files, browser based monitoring data, infrastructure based monitoring data, and client-side based monitoring data, and are usually gathered by implementing specific mechanisms on all Cloud service deliver models.

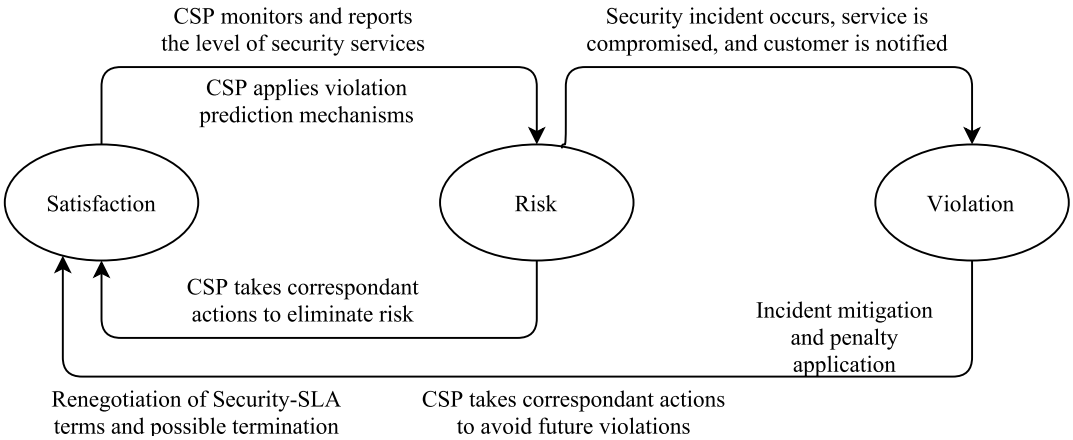


Figure 5.4 The states of the Cloud Security-SLA during the monitoring phase.

The risk state indicates the presence of a potential threat to the Security-SLA fulfillment. The CSP can predict this risk by measuring and capturing different kind of parameters as shown in Table 5.5. These parameters are basically related to the Cloud application's usage pattern, and could influence the performance of the deployed security services. For instance, if the application's packet arrival rate abnormally increases, it could affect the detection performance of the deployed IDS and lead to a violation of the SSLO agreed upon with the customer. It could also indicate an abnormal or malicious activity that might be a part of a DoS attack. Therefore, monitoring such parameters could help the CSP to automatically manage to eliminate the risk of violation by implementing special actions and responses, like changing the applied security policy or mechanism, or migrating the VM to another physical server where the existence of the threat is less probable.

If a violation occurs, the customer is notified and a penalty may apply. The penalty could take the form of compensation in terms of the Cloud resources and could be computed according to the actual and promised quality of security of the service. In this case, the CSP will start

Table 5.5 A set of parameters that could help CSPs in detecting security and performance variation.

Security service	Security and performance metrics	Influencing parameters
Authorization and access control	- Access verification time - Transaction processing overhead	- Traffic load - Request arrival rate
Data and storage security	- Encryption/decryption time - Read/write latency - Average throughput (Ops/s or MB/s) - Computing overhead	- Key size - File size - Database workload
Data and computational integrity	- Integrity check execution time - Computational faults detection accuracy - Processing overhead of PDP techniques	- Database scalability (number of nodes) - Data size - Database updates - Number of constraints to check
Network security	- Attack detection speed (time) - Attack detection accuracy - Packet drop rate - System overhead (e.g., CPU usage) - IDS scalability	- Traffic load - Packet arrival rate - Number of rules to check

the remediation process, and may choose to federate the service to another more secure CSP to avoid future violations. The customer in this case is free to request a renegotiation of the Security-SLA terms or terminate the contract. The customer's feedback on the service could be used by the broker to help with future evaluation and selection experiences.

5.7 Experimentation and Results

In this section, we perform an evaluation of the proposed Security-SLA evaluation and selection model and analyze the results. The goal is to show how our model could guide Cloud customers during service selection.

5.7.1 Experimental Setup

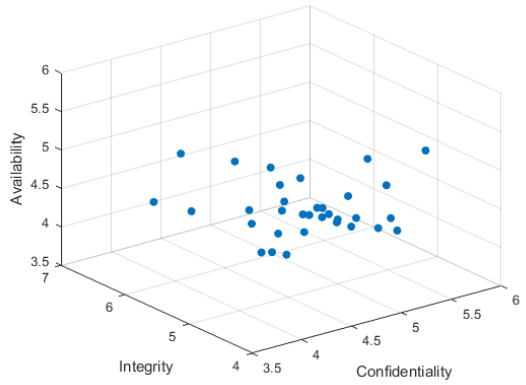
Due to the limited availability of data related to security implementation on the CSPs' side and their lack of transparency about the performance of their deployed security services, it was relatively difficult to efficiently construct an evaluation data set that could represent the actual performance of CSPs with respect to the proposed security metrics. Thus, we decided to generate our own data for the sake of this experiment. Inspired by their nature and role, Table 5.6 shows how the values of some of these metrics could be expressed in a real-life scenario. We also take into consideration the nature of the security solutions (e.g., [121, 136]) that the metrics intend to evaluate, and represent the ones that aim at evaluating

Table 5.6 An example of SSLO values in a real-life scenario.

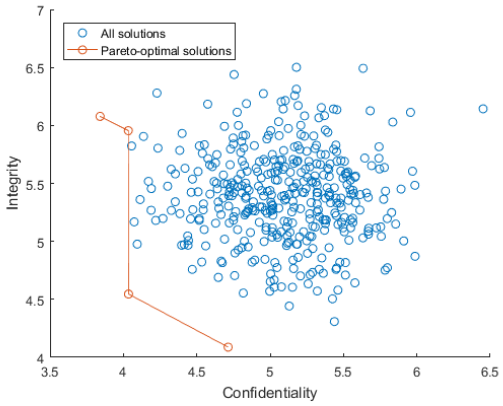
Security attribute	Security metrics	SSLO value			
		Customer	CSP1	CSP2	CSP3
Confidentiality	False Rejection Rate of authentication mechanism	$\leq 10\%$	$\leq 8\%$	$\leq 5\%$	$\leq 10\%$
	Enforcement of Multi-Factor authentication	No	yes	yes	No
	Frequency of review of access control logs	2/day	3/day	1/day	4/day
	SSL protected database connections	Yes	Yes	Yes	Yes
Integrity	False positives of computational faults	$\leq 15\%$	$\leq 12\%$	$\leq 8\%$	$\leq 20\%$
	Computing time of PDP techniques	$\leq 15\%$	$\leq 10\%$	$\leq 5\%$	$\leq 11\%$
	Processing overhead of accountable computation	$\leq 5\%$	$\leq 15\%$	$\leq 5\%$	$\leq 10\%$
Availability	Replication of data between Cloud nodes	Yes	Yes	Yes	No
	Configuration of host-based and guest-based firewalls	Yes	Yes	No	Yes
	Mean-time to mitigate attack	$\leq 2hrs$	$\leq 3hrs$	$\leq 1hrs$	$\leq 4hrs$

the performance and cost criteria (e.g., computational time) of security mechanisms in the form of percentages of the basic values (when security services are not provided) to stress on the generality of the model. Therefore, the evaluation data is randomly generated as follows : the values of positive metrics in the interval [60, 97], the values of negative metrics in the interval [1, 3], and the values of implementation metrics in the binary form (0 or 1). The proposed model was implemented in MATLAB and the tests were run on a Core i7, 2.1 GHz CPU machine with 12 GB of RAM and running Windows 7, and the results illustrated are the average of fifty runs.

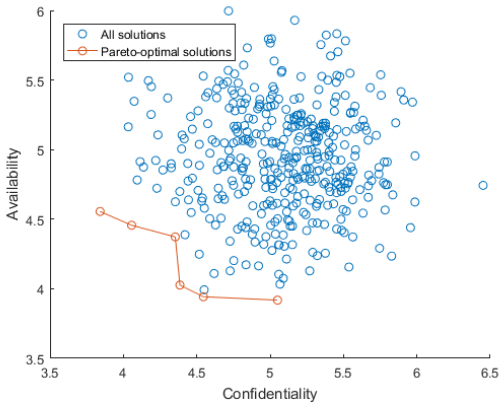
According to Algorithm 1, when computing the DSF of a specific security attribute, not all related metrics will necessarily be considered. For instance, considering the positive metric “frequency of review of access control logs" under the confidentiality attribute, if the SSLO value required by the customer (2/day in the example) is lower than the SSLO value provided by the CSPs (e.g., CSP1 and CSP3), this metric then will not be considered in the computation of the confidentiality dissatisfaction factor for these CSPs. The same analysis applies in the case of negative metrics. Considering the negative metric “mean-time to mitigate attack" under the availability attribute, when the SSLO value required by the customer (maximum of 2 hours in the example) is higher than the SSLO value provided by the CSPs (e.g., CSP2), this metric then will not be considered in the computation of the availability dissatisfaction factor for these CSPs. Finally, for implementation metrics, if the customer's required SSLO value for a certain security policy or mechanism is No (e.g., enforcement of Multi-Factor authentication), then this metric will not be considered in the computation of DSF.



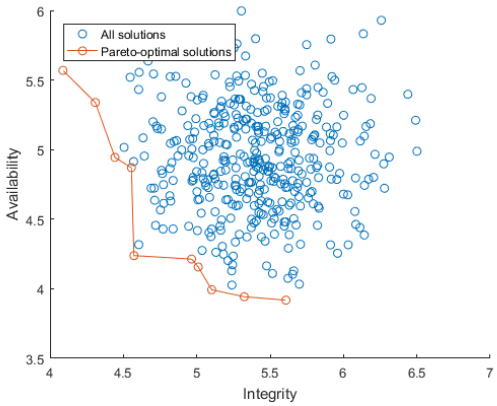
(a) The set of Pareto-optimal solutions involving trade-off between the three security attributes.



(b) Set of solutions involving trade-off between confidentiality and integrity.



(c) Set of solutions involving trade-off between confidentiality and availability.



(d) Set of solutions involving trade-off between integrity and availability.

Figure 5.5 Pareto-optimal solutions to our problem when $N = 200$.

5.7.2 Results Analysis

Figure 5.5 shows the sets of non-dominated solutions to the multi-objective optimization problem after performing an exhaustive search on the solution set when the number of CSPs involved in the evaluation process is equal to 200. Figure 5.5a illustrates the whole Pareto-optimal set of solutions, where each pair of these solutions could be considered equally good, since the satisfaction of one of the three security attributes will not improve without degrading at least one of the others. Thus, to be able to select one solution from this set, the customer will need to process higher level information and perform a guided decision based on external factors. Figures 5.5b, 5.5c, and 5.5d show the Pareto-optimal solutions involving a trade-off between two security attributes at a time. For instance, in Figure 5.5b, all solutions presented in the Pareto-optimal set satisfy the availability requirements of the customer,

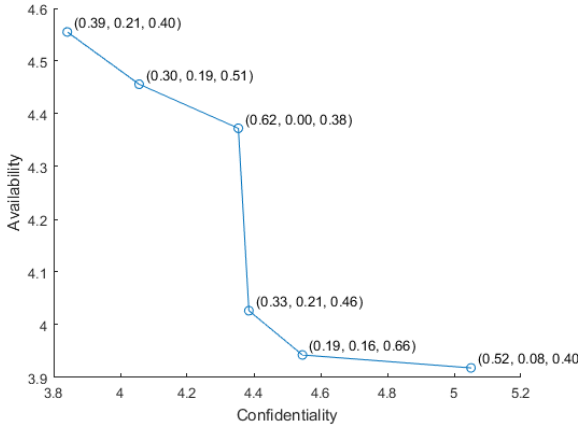


Figure 5.6 Pseudo-weights of the objective functions for the same solutions of Figure 5.5c.

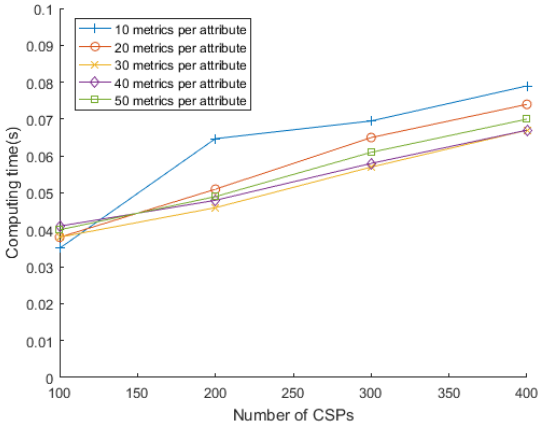


Figure 5.7 Computational time of the evaluation and optimization model.

while the satisfaction of the two other attributes is conflicting. Thus, to be able to choose a single solution out of this set, the customer should decide which of the two attributes, confidentiality and integrity, is more critical to the Cloud service she is willing to deploy.

One of the main contributions of this work with respect to other evaluation and selection frameworks is the fact that, in our model, the customer is presented with a set of solutions, not a single one, which gives the customer more control over her decision and permits to perform a well-guided selection. The reason behind this logic is the fact that security offerings in the Cloud could be very diverse and the full and simultaneous satisfaction of the three security attributes is not always guaranteed. In Figure 5.6, we show the weights of the three objective functions which we compute using Equation 5.8. These weights will determine the degree of the compromise between the corresponding security attributes and which the customer will decide to suffer when selecting a particular CSP to deploy her service. For instance, if the customer prioritizes the attributes in the following order : confidentiality, availability, and integrity, according to the requirements of the deployed Cloud service as explained in section 5.3, and computes an estimated weight vector of (0.6, 0.1, 0.3) using a special weighting procedure (e.g., the one we used in [130]), the customer is more likely to choose the solution with the pseudo-weight vector of (0.52, 0.08, 0.40).

Figure 5.7 shows the computational time of the model when the number of CSPs under evaluation varies between 100 and 400, and while varying the number of metrics used in the evaluation from 10 to 50. Algorithm 1 has a linear time complexity of $O(NM)$ where N is the number of evaluated CSPs and M is the number of metrics used in the evaluation. This will

allow the algorithm to be integrated in the Cloud resources allocation frameworks without introducing a considerable overhead. Performing an exhaustive search on the solution space has a time complexity of $O(N^2)$, where N is the number of CSPs involved in the evaluation. In our model, we focused more on the aspect of evaluation and selection of Security-SLAs and how trade-off solutions could be presented to customers. In the cases where Cloud resources need to be provisioned and allocated to satisfy a large number of service requests, which is a problem that is known in the literature to be NP-complete, performing an exhaustive search for the solution is not ideal, since it will be computationally inefficient. Thus, heuristic approaches such as multi-objective genetic algorithm could be adopted as they could be more suited to the scale of the problem. In all cases, Algorithm 1 that performs the evaluation of Security-SLAs offered by CSPs, could be implemented onto the VM central management unit on the broker side, which will control the placement of VMs on the CSPs' data centers according to security satisfaction.

5.8 Conclusion

Security is one of the main reasons that are holding back the full migration to Cloud computing. Security-SLAs could be employed as a tool for service differentiation. They will allow the CSPs to ascertain their credibility and attract the Cloud customers, and eventually, enhance trustworthiness evaluation between the Cloud stakeholders. Quantification and standardization of Security-SLAs will help speed up the Cloud adoption process, by providing a fully measurable and monitored Cloud security. In this paper, we proposed a broker-based framework that manages the whole life cycle of a Cloud Security-SLA. First, a quantifiable standard version of the Security-SLA for Cloud computing is constructed following a deep analysis of the security services that are usually deployed to protect the data and infrastructure from the threats and incidents associated to the Cloud, and an extensive set of evaluation metrics is developed. Then, the Security-SLA evaluation and selection problem is modeled as a multi-objective optimization problem that aims at generating multiple solutions to the customer which will eventually perform an informed decision making based on computing the adequate trade-off between the three principle security attributes : confidentiality, integrity, and availability. Finally, a Security-SLA monitoring and violation prediction process that is based on states transition is presented.
Experimentation shows how the set of Pareto-optimal solutions could be presented to the customer and how the latter can select the most suitable service provider using higher level information. In future work, we will focus on creating a real data set that could help in evaluating the security of CSPs in a real-life scenario, as well as developing adequate metho-

dologies to monitor the fulfillment of the Security-SLA and the integrity and transparency of the Cloud service providers with respect to security. Extending our current proposal to include the management of Security-SLAs in Cloud federations is also a part of our future plan.

CHAPTER 6 ARTICLE 3 : SERVICE ASSIGNMENT IN FEDERATED CLOUD ENVIRONMENTS BASED ON MULTI-OBJECTIVE OPTIMIZATION OF SECURITY

Talal Halabi and Martine Bellaiche

IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 39-46, 2017.

Abstract

Cloud federation allows interconnected Cloud Computing environments of different Cloud Service Providers (CSPs) to share their resources and deliver more efficient service performance. However, each CSP provides a different level of security in terms of cost and performance. Instead of consuming the whole set of Cloud services that are required to deploy an application through a single CSP, consumers could benefit from the Cloud federation and flexibly assign the services to multiple CSPs in order to satisfy all their services' security requirements. In this paper, we model the service assignment problem in federated Cloud environments as a Multi-objective optimization problem based on security. The model allows consumers to consider a trade-off between three security factors : cost, performance, and risk, when assigning their services to CSPs. The cost and performance of the delivered security services are evaluated using a set of quantitative metrics which we propose. We then solve the problem using the preemptive optimization method which permits to take into consideration the customer's priorities. Simulations showed that the model helps in reducing the rate of security and performance violations.

6.1 Introduction

Cloud Computing has become an increasingly popular technology due to its accessibility, scalability, and Pay-Per-Use service model. However, Cloud insecurity is still preventing the wide adoption of Cloud services and is brought about by numerous new threats presented to the domains of data and application security, virtualization, encryption and key management, identity and access management, and incident response. Deploying an application to the Cloud might not necessarily satisfy all its security requirements since the Cloud Service Provider (CSP) can sometimes perform better in some of the security domains and worse in

others, which can cause violations of security and performance contracts and financial losses. A Cloud consumer usually ponders some sort of trade-off between the application security needs before selecting a service provider which can amount to a challenging sacrifice. The Cloud application is normally deployed using multiple Cloud services such as storage, computing, and monitoring, which can be provisioned using a group of Virtual Machines (VM). For each Cloud service, the application has specific security requirements and settings related to the workload nature and security needs. Thus, distributing these services to different Cloud infrastructures could be the key to deliver more secure applications and avoid security violations.

A Cloud federation can be defined as multiple data centers belonging to different CSPs connected altogether and sharing their resources to deliver efficient service performance. In a federated Cloud, resources are usually distributed and managed between CSPs with respect to multiple factors, such as resource usage, regional workloads, and legal issues. In this context, assigning the Cloud services that are required to deploy an application to different infrastructures within the federation based on their security requirements could achieve the optimal security level that is required for the application in order to function securely. Optimizing the security of an application is principally based on three aspects : (1) minimizing the security cost due to security overhead and loss resulting from security failures, (2) providing the application with the suitable level of security performance, since providing lower or higher levels can lead to technical or financial losses, and (3) deploying the application with minimal risk probability. For instance, failures in data protection or availability due to lower than required security levels can harm the application's trustworthiness and cause a drop in the number of users ; the application may also suffer from Quality of Service (QoS) degradation due to over-security that can be sometimes unnecessary, which can cause Service Level Agreement (SLA) violations.

Our main idea in this paper is presented in the following example. We consider a Cloud application with two components : a computing unit and a storage node (e.g., database). The application performs some computational tasks and makes use of the information stored in the database. Each component entails the deployment of various security services. The computing unit requires the presence of an integrity checking mechanism which takes care of verifying the process' computational integrity, and the storage node necessitates the use of a strong encryption technique in order to protect the data at rest and clients' transactions from unauthorized disclosure. The Cloud application also expects that a set of performance and security constraints must be respected as stated in the SLA and Security-SLA. The whole set of requirements cannot be provided by a single CSP within the federation. Thus, assigning the services to different CSPs based on the optimization of security cost, performance,

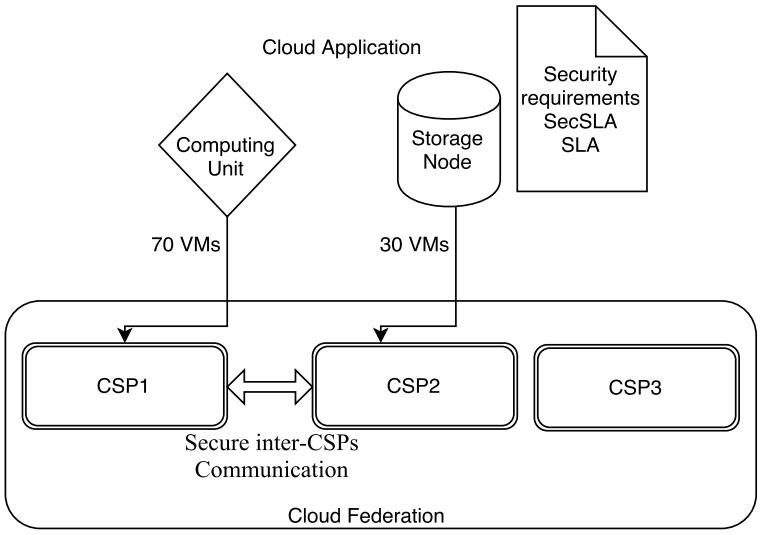


Figure 6.1 An example of assigning an application with two components to multiple CSPs in a Cloud federation.

and risk could benefit both the consumer and the provider by reducing the rate of security and performance violations. This possible solution is represented in Figure 6.1 and can be implemented onto the VM central management unit that controls VMs' placement on the federation data centers and can be fully transparent to consumers.

Security optimization encompasses the quantitative evaluation of security services associated with the different Cloud service delivery models (Software-as-a-Service, Platform-as-a-Service, and Infrastructure-as-a-Service) with the use of well-defined sets of measurable metrics. These security services include identity and access management, data and storage security, network security, and service availability. Multiple security mechanisms (e.g., authentication techniques, attack detection systems, etc.) are implemented in each data center in order to deliver these security service. In this paper, we propose a service assignment model that is based on the evaluation of these security services and mechanisms. The contribution of this paper is twofold :

— First, we model the service assignment problem in a Cloud federation as a Multi-objective optimization problem that aims at satisfying the services' security requirements. The model considers three security factors while assigning the consumer's services to CSPs : cost, performance, and risk, and which are quantitatively evaluated using a set of metrics that we propose.

— Second, we solve the optimization problem with the use of the preemptive method which helps evaluating the trade-off to deliberate between the security factors during service assignment, and run a set of simulations to evaluate the performance of our

model.

The paper is organized as follows. Section 6.2 discusses related work. Section 6.3 describes the adapted federation architecture. Section 6.4 explains our optimization model. In section 6.5 we perform the simulation and analyze the results. Finally, section 6.6 concludes the paper.

6.2 Related Work

The optimization of security is fundamentally based on its efficient quantification and evaluation. CSA [9] and NIST [116] are currently working to standardize Cloud security evaluation using explicit and definite metrics. The optimization model proposed here also involves developing a potential set of quantitative metrics that will help in the security optimization process, as we also did in [129]. Few researchers have discussed the Cloud service selection problem based on security evaluation. Na and Huh [69] proposed a Cloud selection approach based on weights evaluation using the Analytical Hierarchy Process (AHP) method. Luna et al. [119] and Taha et al. [67] also used the AHP technique to benchmark Cloud security based on the SLA provided by the Consensus Assessments Initiative Questionnaire (CAIQ) of the CSA [9]. However, these approaches are not based on measurable security metrics, and always aim to assign the client's services to a single CSP without evaluating the trade-off between the performance of the different security services.

The concept of federation in Cloud environments is still relatively young. Many researchers have addressed the subject of Cloud federation formation from a profit generation perspective. Abadi et al. [137] presented an approach for resource management in IaaS Cloud federations, focusing on energy and cost reduction while allocating the VMs to satisfy users' requests. In [87], Hassan et al. proposed a federation formation mechanism using a trust-based cooperative game theory that allows CSPs to maximize their profit and minimize the SLA penalty cost on QoS by joining federations of trustworthy and reliable CSPs. Service composition based on security trade-offs was not yet studied in previous research due to the immaturity of the security evaluation and quantification aspect in Cloud Computing. For instance, Bernsmed et al. [66] presented an approach to manage Security-SLAs in federated clouds, but did not address in detail the security quantification aspect and the factors that are considered in the assignment process.

6.3 The Security-oriented Federation Architecture

The architecture that we propose for the federation model is service-centric, since it essentially aims at guarantying the quality of security of the Cloud service, and is inspired by the work

of Buyya et al. [138]. It is based on a centralized approach where Cloud customers indirectly interact with the federation through a Cloud broker.
The macro components of the architecture are depicted in Figure 6.2. When a Cloud service request is placed, the Cloud customer negotiates the terms of security and performance contracts with the Cloud broker. The broker collects the necessary security-related data from the CSPs in the federation and applies the proposed evaluation and optimization model in order to assign the customer's services and allocate the VMs to the data centers. The broker continues to monitor the security and performance of the assigned services and performs a new assignment in case of violations. The reassignment is also triggered in the case where the customer wants to renegotiate the security requirements.

6.4 System Model

In this section, we propose the optimization model. We decided to use Multi-Objective optimization since the problem involves optimizing multiple factors (sometimes conflicting) during the assignment process. The input to the optimization model consists of : 1) the security features and parameters provided by the federation data centers (data related to security metrics), 2) the security requirements of the customer, and 3) the security and performance constraints derived from the SLA and Security-SLA. The output is a set of possible optimal Cloud service-to-server assignments, which are technically reflected by the provisioning of VMs.

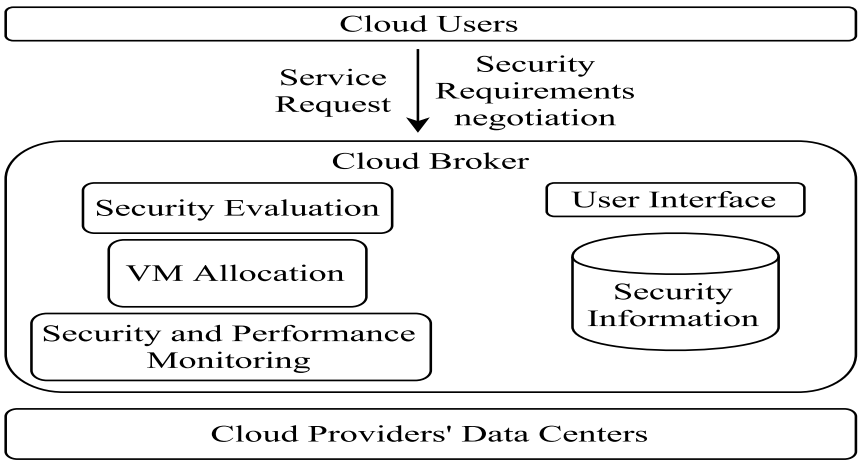


Figure 6.2 Service-centric security-oriented Cloud federation architecture.

6.4.1 Security Factors

A data center usually consolidates hundreds of servers. On each server, a certain number of security mechanisms is deployed. A single service provider can own more than one data center in the federation. When assigning the services to the federation's data centers, the following security factors should be considered.

6.4.1.1 Security Cost

Security cost is partially presented by the security overhead which can be expressed as a percentage of the application resource usage. Overhead can occur at the level of multiple resources such as processing power and memory. Performance impact, which is the impact of the security mechanism on the Cloud application's performance, may also introduce an additional cost. The impact occurs when security overhead exceeds an acceptable threshold, and can be related to certain performance factors such as the application's response time. The impact can be described as a percentage of the application's performance level to indicate degradation. When performance impacts occur, a specific cost must be paid by the customer since the VMs are no more capable of providing the same level of service.

6.4.1.2 Security Performance

The performance of security services is normally reflected by different indicators such as intrusion detection rate, false positives, and detection time. Each data center or service provider offers a security mechanism with different performance metrics, such as the ones presented in Table 6.1. A critical role of the optimization process consists in matching the offered values of these metrics with the required ones and minimizing the difference between them in order to reduce the cost caused by lower or higher security levels.

6.4.1.3 Security risk

Security risk generally exists due to certain vulnerabilities that lie within the security mechanism or on the Cloud server itself, which can create a place for unresolved security threats. The risk is different among the servers and data centers and is usually influenced by environmental, infrastructural, and workload-behavior related factors (e.g., VMs co-location and interference, lack of resource isolation, and poor patch management).
Risk estimation can involve multiple factors as shown in Figure 6.3. For instance, an infrastructure that hosts services for companies and businesses that deal with sensitive information and confidential data (e.g., banks, government, etc.) is more likely to be exposed to attacks

Table 6.1 Some of the Cloud security services and corresponding mechanisms and evaluation metrics.

Security Service	Security Mechanisms	Evaluation Metrics	
		Security Performance	Security Overhead
Identity and Access Management	- Multi-Factor authentication - Single Sign On (SSO) - Access policies and rules - Risk-based entitlement	- False acceptance rate - False rejection rate - Frequency of review of access control logs and accounts' activity - Average response time of transactions	- Processing time - CPU load or processing usage - Memory consumption rates or memory overhead - Network load or bandwidth overhead - Percent of computational overhead - Percent of communication overhead - Percent of storage overhead - Computational complexity - Mean-cost of incident recovery
Network Security	- Network monitoring tools and Intrustion Detection Systems (IDSs) - Host-based and guest-based firewalls - Access Control Lists (ACLs)	- Mean-time to discover attack - Mean-time to mitigate attack - Attack detection rate - False positive rate - Latency or response time - Packet drop rate - Throughput or network traffic rate	
Data and Storage Security	- Data encryption - Key management procedures - Data loss/leakage prevention - Data dispersion - Anti-malware and anti-virus programs - SSL protected connections	- Strength of encryption keys and block ciphers - Frequency of update of anti-malware and anti-virus programs - Percent of data encrypted at rest - Percent of information security and privacy policies aligned with industry standards	
Data and Service Availability	- Data backup - Data replication and redundancy - Multi-failure disaster recovery - Security incident response plan - Business continuity plan - SIEM incident reporting, analysis and alerting	- Mean-time of incident discovery - Mean-time of incident recovery - Recovery time of failed tasks - Cloud failure rate - FRC and DDoS attack detection rate and time	

and threats, which makes the type of consumers an important factor in the risk calculation process. In the literature, Cloud risk estimation was addressed by several works such as [45].

6.4.2 Notations

The granularity level of the proposed optimization model is the server level, since not all the servers in the same data center have the same security configuration. The notations used to express the sets, the constant parameters, and the variables involved in our model are described in this section.

Sets :

— G is a set of Cloud services that are required to deploy an application and are provisioned using groups of VMs. Each service $g \in G$ has different security requirements from the other services.

— D is a set of data centers within the Cloud federation. Each data center $d \in D$ consolidates hundreds of servers.

— S_d is a set of servers in the data center d. Security mechanisms are implemented on each server $s \in S_d$.

— N is the set of Cloud security services that are evaluated during the assignment process. Each security service $n \in N$ is delivered by implementing a set of security mechanisms.

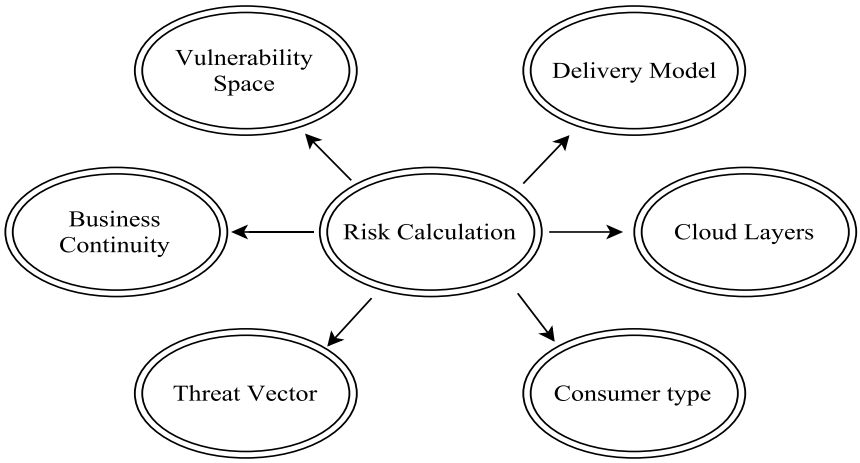


Figure 6.3 Factors involved in Cloud risk estimation.

— M_n is a set of metrics that are used to evaluate the Cloud security service $n \in N$. A metric in the set is denoted by $m \in M_n$.
— R is the set of resources that are usually available in a data center. An overhead might occur on the level of a resource $r \in R$.
— F is a set of performance factors. A performance impact can be caused on the level of each factor $f \in F$.

Constant Parameters :

— $O_n^{ds,r}$ is the overhead introduced by the security service $n \in N$ with respect to resource $r \in R$ on the server $s \in S_d$ in data center $d \in D$.
— u_g^r is the expected usage of service $g \in G$ with respect to resource $r \in R$.
— τ_d^r is the cost rate of resource $r \in R$ in data center $d \in D$.
— $I_n^{ds,f}$ is the performance impact introduced by the security service $n \in N$ with respect to performance factor $f \in F$ on the server $s \in S_d$ in data center $d \in D$.
— L_g^f is the financial loss factor of service $g \in G$ related to the performance impact with respect to performance factor $f \in F$. This parameter is application-specific and can be calculated with the help of experts.
— $reqP_n^{g,m}$ is the required value of metric $m \in M_n$ related to security service $n \in N$ by the Cloud service $g \in G$.
— $avP_n^{ds,m}$ is the available value of metric $m \in M_n$ related to security service $n \in N$ by the data center $d \in D$ on the server $s \in S_d$. Since metrics have different units of measurements, both $reqP_n^{g,m}$ and $avP_n^{ds,m}$ are normalized to the interval [0,1] in order to homogenize computations.
— R_n^{ds} is the risk factor or probability related to security service $n \in N$ on the server $s \in S_d$ in data center $d \in D$.

Variables :

x_{ds}^{g} is a binary variable $\in \{0,1\}$ that determines if service $g \in G$ is assigned to server $s \in S_d$ in data center $d \in D$.

6.4.3 Defining Objective Functions

The objective functions that should be minimized during the service assignment process are defined in the following.

Total Security Cost (TSC) :

The cost of overhead introduced by the deployment of security service $n \in N$ to all resources $r \in R$ for a Cloud service $g \in G$ on a server $s \in S_d$ in data center $d \in D$ is given by :

$$\sum_{r=1}^{R} \tau_d^r * u_g^r * O_n^{ds,r} \tag{6.1}$$

The cost of performance impact introduced by the deployment of security service $n \in N$ with respect to all performance factors $f \in F$ for a Cloud service $g \in G$ on a server $s \in S_d$ in data center $d \in D$ is given by :

$$\sum_{f=1}^{F} L_g^f * I_n^{ds,f} \tag{6.2}$$

Therefore, the objective function f_{TSC} that expresses the total security cost introduced by all security services when assigning a Cloud service $g \in G$ to a server $s \in S_d$ in data center $d \in D$ can be defined as :

$$f_{TSC} = \sum_{g=1}^{G} \sum_{d=1}^{D} \sum_{s=1}^{S_d} x_{ds}^{g} * \sum_{n=1}^{N} (\sum_{f=1}^{F} L_g^f * I_n^{ds,f} + \sum_{r=1}^{R} \tau_d^r * u_g^r * O_n^{ds,r}) \tag{6.3}$$

Mean Security Performance Dissatisfaction (MSPD) :

This function aims at assigning the Cloud service to the CSP who is offering the optimal security performance, that is, the closest to the security level required by the consumer's application. To this end, the mean security performance dissatisfaction related to a security service $n \in N$ and resulting from assigning a Cloud service $g \in G$ to a server $s \in S_d$ in data center $d \in D$ is evaluated using the Euclidean distance and is given by :

$$1/M_n * \sqrt{\sum_{m=1}^{M_n} (avP_n^{ds,m} - reqP_n^{g,m})^2} \tag{6.4}$$

The objective function f_{MSPD} that expresses the total Security Performance Dissatisfaction

of all security services $n \in N$ and resulting from assigning a Cloud service $g \in G$ to a server $s \in S_d$ in data center $d \in D$ can be then defined as :

$$f_{MSPD} = \sum_{g=1}^{G} \sum_{d=1}^{D} \sum_{s=1}^{S_d} x_{ds}^{g} * 1/N * \sum_{n=1}^{N} (1/M_n * \sqrt{\sum_{m=1}^{M_n} (avP_n^{ds,m} - reqP_n^{g,m})^2} \tag{6.5}$$

A specific weight can also be computed for each metric and considered during the evaluation process. Weights can be calculated using the standard deviation of collected data which could reflect the relative importance of the metrics to the optimization model.

Mean Security Risk (MSR) :

One of the goals of the security optimization process is to minimize the possible risk when assigning a Cloud service to a CSP's infrastructure. The objective function f_{MSR} that expresses the mean Security risk which exists on the level of all security services $n \in N$ deployed on a server $s \in S_d$ in data center $d \in D$ can be defined as :

$$f_{MSR} = \sum_{g=1}^{G} \sum_{d=1}^{D} \sum_{s=1}^{S_d} x_{ds}^{g} * 1/N * \sum_{n=1}^{N} R_n^{ds} \tag{6.6}$$

6.4.4 Optimization Constraints

The optimization problem is subject to some constraints derived from security and performance contracts, as described in the following :

— For each Cloud service $g \in G$, the total security overhead should not exceed an allowed threshold $Overhead_g^r$ with respect to every resource $r \in R$, in order to avoid QoS degradation or financial cost increase. This constraint could be specified in the SLA and is represented in our model by :

$$\sum_{d=1}^{D} \sum_{s=1}^{S_d} x_{ds}^{g} * \sum_{n=1}^{N} O_n^{ds,r} < Overhead_g^r, \ \forall r \in R \ and \ \forall g \in G \tag{6.7}$$

— For each Cloud service $g \in G$, the total performance impact should not exceed an allowed threshold $Impact_g^f$ with respect to every performance factor $f \in F$ (e.g., response time), as also specified in the SLA. This constraint is specified as follows :

$$\sum_{d=1}^{D} \sum_{s=1}^{S_d} x_{ds}^{g} * \sum_{n=1}^{N} I_n^{ds,f} < Impact_g^f, \ \forall g \in G \ and \ \forall f \in F \tag{6.8}$$

— For each Cloud service $g \in G$, the mean security risk should not exceed an acceptable

threshold $Risk_g$, as specified in the Security-SLA. This constraint is given by :

$$\sum_{d=1}^{D}\sum_{s=1}^{S_d} x_{ds}^{g} * 1/N * \sum_{n=1}^{N} R_n^{ds} < Risk_g,\ \forall g \in G \tag{6.9}$$

In addition, the assignment constraint states that each Cloud service $g \in G$ must be assigned to at least and only one server in the federation :

$$\sum_{d=1}^{D}\sum_{s=1}^{S_d} x_{ds}^{g} = 1,\ \forall g \in G \tag{6.10}$$

Other constraints related to the mean security performance dissatisfaction and the values of metrics could also be derived from the Security-SLA. These constraints are not discussed here and are left for future work.

6.4.5 Problem Formulation

The problem described in this paper can be formulated as a Multi-Objective Constrained Integer Linear Programming (MOCILP) problem as follows :

$$\min \{f_{TSC}, f_{MSPD}, f_{MSR}\} \tag{6.11}$$

Subject to :

$$\sum_{d=1}^{D}\sum_{s=1}^{S_d} x_{ds}^{g} * \sum_{n=1}^{N} O_n^{ds,r} < Overhead_g^r,\ \forall r \in R\ and\ \forall g \in G \tag{6.12}$$

$$\sum_{d=1}^{D}\sum_{s=1}^{S_d} x_{ds}^{g} * \sum_{n=1}^{N} I_n^{ds,f} < Impact_g^f,\ \forall g \in G\ and\ \forall f \in F \tag{6.13}$$

$$\sum_{d=1}^{D}\sum_{s=1}^{S_d} x_{ds}^{g} * \sum_{n=1}^{N} R_n^{ds} < Risk_g,\ \forall g \in G \tag{6.14}$$

$$\sum_{d=1}^{D}\sum_{s=1}^{S_d} x_{ds}^{g} = 1,\ \forall g \in G \tag{6.15}$$

6.4.6 Proposed Solution : Preemptive Optimization

In this section, we solve our multi-objective minimization problem. A feasible solution x is considered efficient if there is no other solution y such that $f_q(y) \leq f_q(x)$ for all $q \in \{TSC, MSPD, MSR\}$ and $f_q(y) < f_q(x)$ for at least one $q \in \{TSC, MSPD, MSR\}$. The

Table 6.2 Simulation parameters.

		MIN	MAX
Number of Cloud services G		3	10
Number of data centers D		5	40
Number of servers S_d		10	1000
Number of security services N		12	12
Number of metrics M_n		7	10
Number of resources R		3	3
Number of performance factors F		3	3
Overhead $O_n^{ds,r}$ (%)		0.5	2
Resource usage u_g^r	CPU (h)	40	60
	Memory (GBs)	0.5	2
	Disk (GBs)	5	20
Resource cost rate τ_d^r	CPU ($/h)	0.01	0.11
	Memory ($/GBs)	1	3
	Disk ($/GBs)	1	3
Performance impact $I_n^{ds,f}$ (%)		0	1
Financial loss factor L_g^f ($/1% of impact)		1	2
$reqP_n^{g,m}$ and $avP_n^{ds,m}$ normalized values of metrics		0	1
Risk factor R_n^{ds} (%)		1	7
Overhead threshold $Overhead_g^r$ (%)		15	20
Performance impact threshold $Impact_g^f$ (%)		5	15
Risk threshold $Risk_g$ (%)		4	8

set of all efficient points is called the efficient frontier, and the best solution to the problem will be selected from this set regardless of the objectives' prioritization procedure.

In our case, the solution to the assignment problem can be obtained by assigning priorities to the competing objective functions based on what is required by the consumer's application. These priorities depend on multiple factors such as the application type (e.g., web server, storage service, etc.), data sensitivity, security budget, or QoS. The Preemptive Optimization method [102, 103] seemed suitable to solve the problem, since it allows a user friendly priority estimation and takes into consideration the fact that the customer might not be an expert in security. For instance, a Cloud customer whose requested service does not involve dealing with private information and confidential data, and would like to maintain the adequate QoS level of the Cloud application while providing an acceptable level of security, would more likely choose the priority order $TSC > MSPD > MSR$ during the negotiations with the broker.

The preemptive method performs the optimization by considering one objective at a time according to priorities. After optimizing every objective, an optimal objective value is obtained and is used as a bound in a new constraint when optimizing the next objective. The final solution constitutes an efficient point of the initial set of feasible solutions.

6.5 Simulation and Results

In this section, we perform a set of simulations in order to evaluate our model. Due to the difficulty in collecting real data that reflect the performance of CSPs regarding security, we decided to generate our own data for the sake of this experiment. We solve the optimization problem using the Mixed-Integer Linear Programming function «intlinprog()» from the MATLAB Optimisation toolbox [139]. The solver follows a basic strategy that consists of, first, reducing the problem size, then solving an initial relaxed problem and finding integer-feasible solutions using heuristics, and finally applying a Branch and Bound algorithm to systematically search for the optimal solution. The simulation parameters are randomly generated in the intervals [Min,Max] as presented in Table 6.2. In general, the solver will try to determine the best service-to-server configuration based on the given priority order. Once the optimal configuration is found, the solver returns the equivalent value of each objective function using equations 6.3, 6.5 and 6.6 respectively. All following scenarios are run on a Core i7, 2.1 GHz CPU machine with 12 GB of RAM and running Windows 7.

6.5.1 Simulating different priority orders

In the first scenario, we solve the assignment problem for different optimization priority orders. Table 6.3 shows the values of the objective functions for each priority order. All the results are the average of fifty runs. For instance, if we consider the first two cases where the cost has higher priority than performance and vice versa respectively, the value of TSC increased from 567.9$ in the first case to attain more than the double in the second case, whereas MSPD decreased from 30.1% in the first case to reach a more optimal value of 23.84% in the second. This shows how flexible the service assignment can be based on what actually is required by the consumer's application and the nature of the trade-off between the different security factors.

6.5.2 Security and performance violations

The second scenario is presented to demonstrate the contribution of our model. We consider an assignment problem where three Cloud services (Computing, Storage, and Monitoring) with different security requirements and constraints need to be mapped to a set of servers. We consider two different cases during the simulation. In the first case, the three services will be assigned to only one server in the federation. A trade-off between the services' security requirements is to be considered in this case. We apply the optimization model with priority order $TSC > MSR > MSPD$ as per the customer's preferences and only according to the

Table 6.3 Evaluation of objective functions for different priority orders.

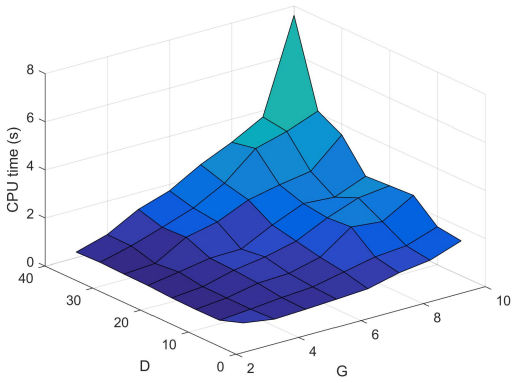
	Priority order TSC >MSPD >MSR		Priority order MSPD >TSC >MSR		Priority order MSR >TSC >MSPD	
	Avg	*Std*	*Avg*	*Std*	*Avg*	*Std*
TSC ($)	567.9	114.2	1198.5	376.77	1357.9	361.1
MSPD (%)	30.1	1.46	23.84	0.58	30.11	1.04
MSR (%)	12.08	1.12	11.83	1.17	6.97	0.4

Table 6.4 Evaluation of Security-SLA and SLA violations. Case (1) : all Cloud services are assigned to one server and Case (2) : the proposed model is applied. Optimization priority order : $TSC > MSR > MSPD$. D=10 and $S_d \in$ [10,100].

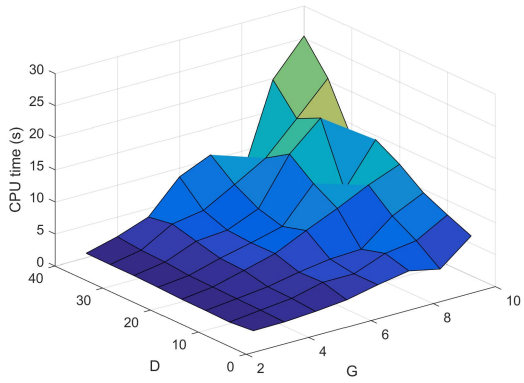
	Security Overhead (%)		$Overhead_g^f$ (%)	Security-SLA Violation		Performance Impact (%)		$Impact_g^f$ (%)	SLA Violation		Security Risk (%)		$Risk_g$ (%)	Security-SLA Violation	
	(1)	(2)		(1)	(2)	(1)	(2)		(1)	(2)	(1)	(2)		(1)	(2)
Computing	14.2	14.2	16	No	No	6.45	6.45	14	No	No					
	14.6	14.6	17	No	No	7.34	7.34	10	No	No	3.81	3.81	6	No	No
	16.7	16.7	19	No	No	5.57	5.57	9	No	No					
Storage	14.2	13.3	15	No	No	6.45	4.3	5	Yes	No					
	14.6	16	20	No	No	7.34	7.12	8	No	No	3.81	3.24	5	No	No
	16.7	13	15	Yes	No	5.57	6.42	14	No	No					
Monitoring	14.2	15	17	No	No	6.45	4.32	6	Yes	No					
	14.6	13	16	No	No	7.34	5.76	6	Yes	No	3.81	4.4	7	No	No
	16.7	15	16	Yes	No	5.57	5.11	7	No	No					

security requirements and constraints of the computing service which is considered to be of higher interest to the customer in our scenario. When a solution is found, the other services will be assigned to the same server even if the chosen server does not fulfill their security requirements. In the second case, our proposed assignment model will be applied and the services will be distributed on multiple servers in the federation according to their security requirements.

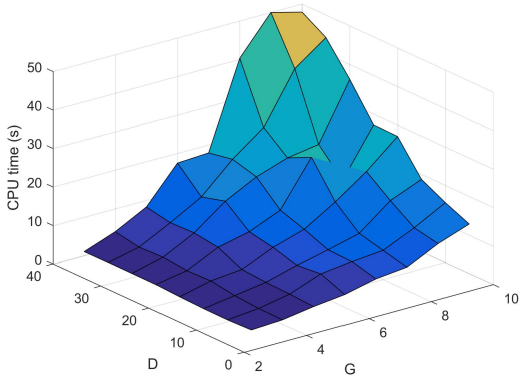
As we see in Table 6.4, violations of the Security-SLA and SLA constraints have appeared for the storage and monitoring services in Case (1). For instance, the security overhead introduced to the disk resource (16.7%) exceeded the overhead threshold allowed for the storage (15%) and monitoring (16%) services, and the performance impact related to the first performance factor (6.45%) also exceeded the impact threshold allowed for the storage (5%) and monitoring (6%) services. These violations were avoided in Case (2) where the optimization model proposed in this paper was applied and the damaging compromise between the security requirements of the different services was eliminated.



(a) CPU time when the number of servers is between 100 and 400.



(b) CPU time when the number of servers is between 400 and 700.



(c) CPU time when the number of servers is between 700 and 1000.

Figure 6.4 Computational time of the model.

6.5.3 Computational time

In the last scenario, we aim at measuring the processing time that is required to find the optimal solution to the problem, while varying the number of Cloud services to be distributed (parameter G), the Cloud federation size (parameter D), and the Cloud size (parameter S_d). The results are shown in Figure 6.4. We can see that the three factors contribute in a way or another in influencing the processing time of the model, since the number of decision variables is directly related to these three parameters. For instance, increasing the number of Cloud services during the assignment process had higher computational overhead in the last case (Figure 6.4c) where the Cloud size was the largest. In conclusion, we can essentially relate the high increase in computational time to both the federation size and the Cloud size, since the model is based on a server level granularity. The computational time could be enhanced if we

consider a cluster of servers of similar security configuration as the assignment destination instead of a single server.

6.6 Conclusion and Discussion

We proposed in this paper a security multi-objective optimization model that will help Cloud consumers to benefit from Cloud federation and flexibly assign their services to multiple CSPs in order to achieve the optimal trade-off between the different security factors such as cost, performance, and risk. We considered specific quantitative security evaluation metrics during the optimization process. We then solved the assignment problem using the Preemptive Optimization method which will allow consumers to control the assignment process based on what is mostly required by their applications. Simulations showed that this model helps in reducing the rate of security and performance violations. The factors that we considered in the security optimization problem are basic but essential. Other factors related to Cloud federation environment such as the reputation and trust of the providers could also be estimated [140] and considered, leading to an enhanced trade-off evaluation. Factors that are not related to security such as geographical location and network links' cost can also influence the assignment process, since they control the overhead that is introduced due to the communication between the different services deployed in the federation.

The security-oriented federation architecture that we proposed is service-centric and broker-based, focusing mainly on the quality of security of the Cloud service. However, it could be adapted to become a decentralized architecture where the role of the Cloud broker is fulfilled by the providers themselves through a peer-to-peer approach. This provider-centric security-oriented federation architecture will primarily consider the cost and profit that providers can obtain when delegating or locally deploying the users' services, and could be based on the work in [141]. In our model, the federation aspect was assumed to enable the required communication between the different service components which would be difficult if the infrastructures were note connected. However, if this communication is not required, it would be possible to adapt the model to a Multi-Cloud setting where services are deployed with different CSPs through a mix and match delivery model to exploit the security capabilities of each CSP.

Finally, we realize that it might be challenging for the optimization method that we proposed to solve the multi-objective problem, to cope with the large scale and scalability characteristics of a Cloud federation. Another way to solve the optimization problem would have involved the use of the Weighted Sum method [142], where the objective functions are first normalized (since not all of them have the same units of measurements), then multiplied

by their weights and summed in order to form a single objective function to be minimized. In that case, the priorities can be expressed by computing a subjective weight (based on consumer's preferences) for each of the security factors involved in the optimization process. In future work, we plan to use a metaheuristic approach to approximate the solution to the service placement problem and provide a model that is implementable in real-time mode. We are also designing a framework for quantification and management of Cloud Security-SLAs between multiple CSPs in a federation environment.

CHAPTER 7 ARTICLE 4 : TOWARDS SECURITY-BASED FORMATION OF CLOUD FEDERATIONS : A GAME THEORETICAL APPROACH

Talal Halabi and Martine Bellaiche

IEEE Transactions on Cloud Computing, 2018.

Abstract

Cloud federations allow Cloud Service Providers (CSPs) to deliver more efficient service performance by interconnecting their Cloud environments and sharing their resources. However, the security of the federated service could be compromised if the resources are shared with relatively insecure CSPs, and violations of the Security Service Level Agreement (Security-SLA) might occur. In this paper, we propose a Cloud federation formation model that considers the security level of CSPs. We start by applying the Goal-Question-Metric (GQM) method to develop a set of parameters that quantitatively describes the Security-SLA in the Cloud, and use it to evaluate the security levels of the CSPs and formed federations with respect to a defined Security-SLA baseline, while taking into account CSPs' customers' security satisfaction. Then, we model the Cloud federation formation process as a hedonic coalitional game with a preference relation that is based on the security level and reputation of CSPs. We propose a federation formation algorithm that enables CSPs to join a federation while minimizing their loss in security, and refrain from forming relatively insecure federations. Experimental results show that our model helps maintaining higher levels of security in the formed federations and reducing the rate and severity of Security-SLA violations.

7.1 Introduction

The adoption of Cloud Computing technology is presently on the rise, and the Cloud market is expecting to keep its rapid growing pace in the next few years [1]. By transferring their businesses to the Cloud, customers benefit from many interesting features such as scalability, resilience, high performance, on-demand and Pay-Per-Use service model. However, outsourcing services to a third party adds a new level of risk due to loss of control, and introduces many security threats such as data breaches, data loss, and denial of service [70], which makes security an essential driving factor of the Cloud market today. Customers expect from the Cloud Service Providers (CSPs) to maintain the security and availability of their data and services, and demonstrate compliance with current security standards. Evaluating the

security of CSPs is a tough task, due to the difficulty in fully quantifying it, but it will permit to speed up the Cloud adoption process by providing sufficient and transparent information about the security of the offered services to customers and facilitating their comparison during the decision making process. The work on developing the adequate terms and policies that will form the future Cloud Security Service Level Agreement (Security-SLA) which will govern security management between providers and customers is presently very active. This agreement will help in guarantying the rights of each party and will require full commitment from both sides to avoid security violations and what entail in terms of financial and technical penalties.

The Cloud Computing paradigm also introduces important challenges for CSPs, such as performance guarantee, resource limitation, disaster-recovery planning, regional distribution of workloads, and legal issues. To address these problems, the concept of Cloud federation was born. It allows a CSP to flexibly and transparently outsource a portion of its users' requests to other independent CSPs, especially when the limitation of available resources on the CSP's side can't cope with the dynamic nature of the Cloud workload and the variability in users' requests for data and computing-intensive applications. This federation between CSPs can occur at different service delivery models : Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS), and along two different dimensions : horizontal, which takes place at matching layers of the Cloud Stack, and vertical, which spans multiple layers in order to service the additional requests on one specific layer through delegation [5]. By providing larger amounts of resources, federated Cloud infrastructures maintain higher performance and Quality of Service (QoS) levels, and improve cost-effectiveness and energy efficiency [5]. In addition, their objectives can also involve enhancing resilience against failures or unexpected situations, and redundancy implementation by replicating computations and data among CSPs to increase security or availability [143].

7.1.1 Problem Definition

In order to form a Cloud federation and allow CSPs to find their potential federation candidates, the state-of-the-art has proposed several models, principally based on factors such as profit maximization, trustworthiness, and QoS parameters. Security was ignored in the Cloud federation formation process, mainly due to the difficulty of its evaluation. For instance, joining a federation of trustworthy CSPs could ensure service reliability and maintain or enhance the required QoS level. Similarly, entering a federation that maximizes profit and reduces cost could be very beneficial to CSPs, but both approaches do not guarantee service safety. Forming a federation with insecure CSPs could increase security risk and hurt the reputation of the federation members. For example, if a CSP delegates its workload to ano-

ther CSP with a security level that is lower than its own, it might risk to not satisfy all its clients' security requirements and end up violating some of the terms in the Security-SLA. In addition, this kind of federation could be dangerous to a CSP since its federated VMs still can communicate with the VMs that exist on its own infrastructure, which permits to an attack or any malicious activity to propagate through the infrastructures of the federation members.

Therefore, to guarantee security satisfaction, limit the rate of security violations, reduce the cost of applied penalties, and protect the Cloud infrastructures from the risk of avoidable security threats, the level of security services provided by the federation members should be efficiently evaluated before the formation of federations. Such services include identity and access management, data and application security, encryption and key management, virtualization security, and incident response plans. Since other criteria such as pricing and QoS (i.e., response time and availability) are also important for the formation of Cloud federations, which is normally performed with the objective of reducing deployment costs or increasing the level of performance of large applications, these criteria could eventually be combined along with the security criterion to provide more efficient and secure services. In this paper, we tackle the Cloud federation formation problem within a security context. Our focus is on the IaaS model and the horizontal federation between independent Cloud infrastructures, where resources are shared between the CSPs in the form of Virtual Machine (VM) instances as illustrated in Figure 7.1. This form of resource federation is the basis of the other federation models, and performing it securely will imply the protection of all kinds of federated services. The security of the IaaS model is explicitly highlighted since the other service delivery models are built on top of it and any security breach on the IaaS layer will result in security compromise on the other layers.

7.1.2 Contribution

In this paper, we discuss the Cloud federation formation problem while considering the security levels of CSPs. We design a security-based hedonic coalitional game that models the Cloud federation formation process, and which goal is to increase the number of secure federations by minimizing the number of insecure members within. The game reduces the loss in the security level of CSPs caused by moving from a non-cooperative state and joining a federation, and tends to maintain a stable rate of secure federations. Our contributions are stated as follows :

— First, we make use of the Goal-Question-Metric (GQM) [42] method to develop a set of quantitative Security-SLA parameters that evaluate the level of deployed security services on the different layers of the Cloud architecture.

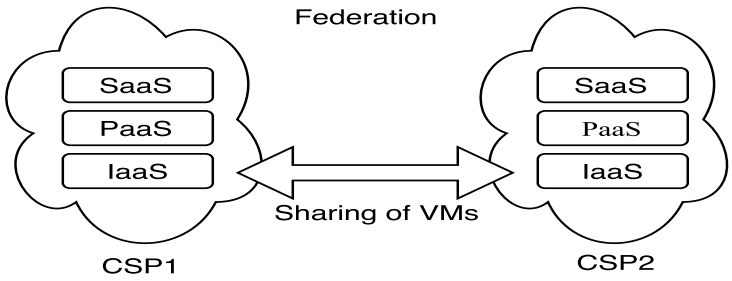


Figure 7.1 Federation model between two CSPs.

— Second, we propose a method to evaluate and compute the security levels provided by CSPs' Security-SLAs, and the security levels of formed federations relatively to a defined security baseline, while considering CSPs' reputations.
— Third, taking into account the computed security levels, we propose a Cloud federation formation algorithm based on a hedonic coalitional game with a security preference relation that satisfies the stability property [105], that is, none of the CSPs has incentive to leave its current federation to join another one that is relatively more secure.

The properties of the security-based Cloud federation formation game are then analyzed, and a set of experiments is performed to study the efficiency of the proposed algorithm. Results show that the model helps in separating secure from insecure CSPs when forming federations, and leads to reduced rates of Security-SLA violations.

7.1.3 Paper Organization

The remainder of the paper is structured as follows. Section 7.2 discusses the literature review related to Cloud security evaluation and federation formation. Section 7.3 gives an overview of the proposed security-based Cloud federation formation framework. Section 7.4 addresses the development and quantification of the Cloud Security-SLA and the evaluation of security of CSPs and federations. Section 7.5 describes the security-based federation formation game model that we propose. In Section 7.6, experimental results are presented and analyzed. Finally, section 7.7 concludes the paper.

7.2 Literature Review

Current security standards are helping CSPs to some extent in implementing and evaluating their security systems. However, more work is still required to standardize the evaluation of security in Cloud Computing, which on one hand, will facilitate the comparison between the

different available Cloud services and the customer's decision making process, and on the other hand, will create a competitive market where CSPs are encouraged to deploy more secure and transparent services, speeding up the Cloud adoption movement. The National Institute of Science and Technology (NIST) [116] and the Cloud Security Alliance (CSA) [9] are investing a lot of efforts in this area through different projects like the Cloud Computing Service Metrics Description [100] and the Cloud Controls Matrix (CCM) within the Governance, Risk Management and Compliance (GRC) Stack [117] initiated by the CSA. However, these projects still lack the security quantification aspect. The SPECS project [52] is also an example of such effort. Its main objective is to develop a framework that offers Security-as-a-Service in the Cloud based on the security parameters specified in the SLA, and provide the necessary techniques to manage its life cycle. In our previous work [129], we tried to quantify the evaluation of security of CSPs and developed a potential set of quantitative security metrics that can be used to evaluate their deployed security services. One of our contributions in this paper over our previous work is the evaluation of security according to a well-defined baseline instead of evaluating the security level of CSPs with respect to each other. We review here some of the work on Cloud security evaluation and the formation of Cloud federations.

Da Silva et al. [41] described a hierarchy of security metrics which they derived using the GQM methodology to characterize security in the Cloud. The hierarchy is based on categorizing the metrics according to their objectives while ignoring their nature and type of measurement. This influences the accuracy of the evaluation process, in addition to the use of a conversion function that maps the actual value of the metric onto a discrete scale causing possible loss of important information. Mirković [40] also defined a system of metrics based on the ISO 27001 standard security controls to measure security in the Cloud. However, his model did not cover all critical security aspects and services in the Cloud and did not lead to a possible quantitative evaluation methodology.

Na and Huh [69] developed a Cloud service selection model based on the evaluation of Security-SLA. They considered five of the nine most notorious security threats to Cloud Computing [70] which they judged critical from the user's perspective : data breaches, data loss, account hijacking, insecure APIs, and malicious insiders, and mostly focused on computing their subjective weights using the Analytic Network Process (ANP) method. Zhengwei et al. [71] proposed a quantifiable system of Cloud-oriented Security-SLA indicators also using the GQM method. They divided their metrics according to the benefit and cost types and used a nearness calculation strategy to relatively evaluate the service level by computing the normalized weighted distance from the real service level to the best and worst service levels. Taha et al. [67] used the Analytic Hierarchy Process (AHP) technique to evaluate

Cloud security based on the Security-SLA provided by the Consensus Assessments Initiative Questionnaire (CAIQ) of the CSA [9]. In [68], the same authors performed a quantitative assessment of a more elaborated Security-SLA to evaluate the security levels of CSPs according to customers' requirements. However, their approaches did not address the quantification and measurement of security of CSPs with respect to a standard security baseline, independently from customers' security requirements, which is the approach we follow in this paper.
The concept of Cloud federations is still immature. Many researchers have addressed the subject of Cloud federation formation from a profit generation perspective. For instance, Li et al. [83] proposed an algorithm for VMs trading in a Cloud federation using an auction-based scheduling mechanism that maximizes the profit of the federation members. Samaan [84] designed a resource sharing strategy in a Cloud federation that increases the revenue based on game theory. Mashayekhy et al. [85] introduced a Cloud federation formation game that allows CSPs to maximize their profit. Finally, Guazzone et al. [86] used the cooperative game theory to develop an algorithm that allows the formation of federations while maximizing the profit of CSPs and reducing the energy cost.
Other research have addressed the problem of formation of Cloud federations from a trustworthiness perspective. In [87], Hassan et al. proposed a federation formation mechanism using a trust-based cooperative game theory that allows CSPs to maximize their profit and minimize the SLA penalty cost on QoS by joining federations of trustworthy and reliable CSPs. In [88], Abdel Wahab et al. also proposed a trust-based hedonic coalitional game that permits the formation of multi-cloud communities of trustworthy services. However, none of these approaches have considered the security factor when addressing the subject of federation formation, and which could have serious consequences on the security of the formed federations. Evaluating the security level of CSPs and taking it into account while forming the federations is the research gap which we are trying to fill with our work in this paper.

7.3 The Proposed Security-Based Federation Formation Framework

The proposed framework for security-based Cloud federation formation involves two stages and is depicted in Figure 7.2. The first stage addresses the evaluation of security in the Cloud and consists on, first, developing a Security-SLA for the Cloud, that describes in quantifiable terms the security provided by CSPs and measures its performance and cost, and second, evaluating the security level of CSPs and the possible federations that could form between them. This evaluation takes into account the reputation of CSPs with respect to security and which is computed based on their customers' satisfaction, and the amount of resources they intend to share with the federation.

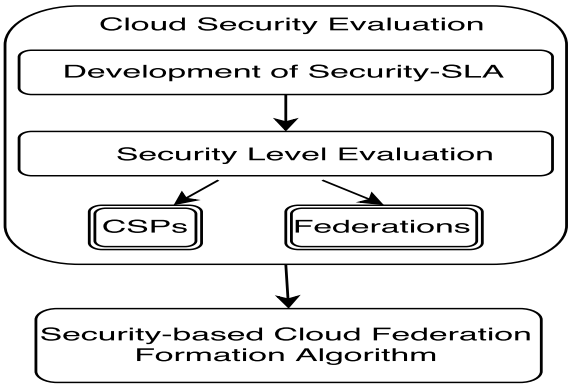


Figure 7.2 The proposed framework.

The second stage consists on applying a federation formation algorithm that we propose based on a hedonic coalitional game which takes into consideration a security preference relation that we define. The security level computed during the first stage will be fed to the formation algorithm which will generate a set of Cloud federations. In the proposed framework, the Cloud broker acts as an intermediate party between the customers and providers. The broker stores the information related to the Security-SLA offered by each CSP along with the reputation values of CSPs which are regularly updated based on the new interactions with their customers. During the federation formation game, the CSPs interact with the Cloud broker in order to obtain the necessary information about each other's security levels and reputations. The framework is discussed in detail in the following sections.

7.4 Cloud Security Evaluation

Cloud security evaluation creates many challenges, such as the necessity to consider all security aspects in the Cloud, the need to cover all the layers of the Cloud architecture and what could entail of threats and incidents, the difficulty in quantifying the security level of a Cloud infrastructure, and the requirement for standardization to establish a fair and reasonable evaluation of CSP's security offers. In this section, we propose a model of the Cloud Security-SLA and a method to evaluate the security level of CSPs based on their offered Security-SLAs.

7.4.1 Cloud Security-SLA

The Cloud Security-SLA is an agreement between the Cloud provider and customer that describes, using generic statements or numerical values, how and how much the CSP will ensure the security of user's data and services. In this agreement, we should mainly find a

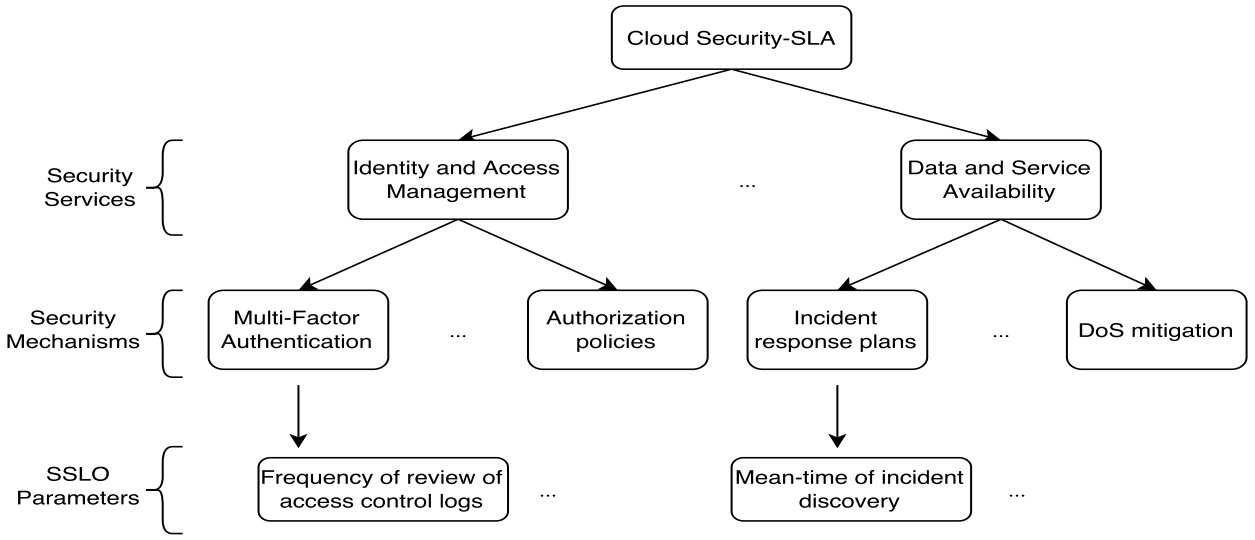


Figure 7.3 The Cloud Security-SLA model.

specification of the deployed security services on the provider's side, along with the implemented security techniques and mechanisms, and a definition of their performance levels, usually done with the help of what we called Security Service Level Objective (SSLO) parameters. The Cloud Security-SLA could be hierarchically modeled as illustrated in Figure 7.3. In current practices, CSPs do not specify the level of their deployed security services, mainly due to the lack of standard vocabularies and adequate quantitative parameters that express this level, and also to avoid transparency. We propose in this paper, that the Security-SLA should include measurable security parameters, along with the qualitative ones, such as regulations and legal restrictions for data processing and storage. These quantitative parameters should also be standardized in order to facilitate the evaluation and comparison of security of different Cloud infrastructures.

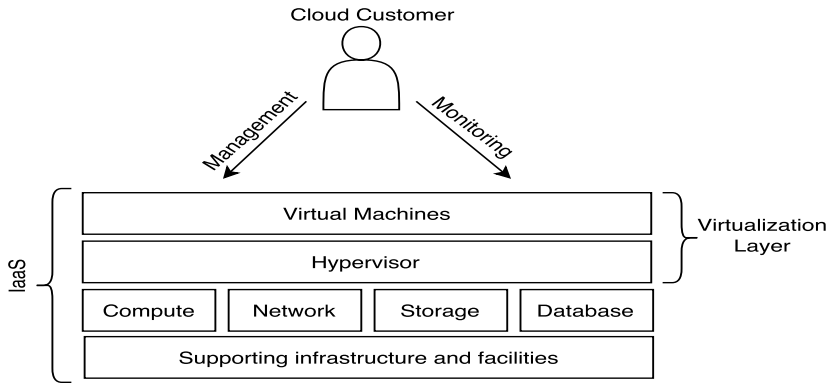


Figure 7.4 Architecture of the Cloud IaaS service delivery model.

Table 7.1 Some common threats to the Cloud IaaS model.

Architecture component	Threats
Virtualization	- Cross-VM attack via Side Channels - VM hopping or escape - Insecure VM migration
Data storage	- Data leakage or manipulation - Data loss - Data scavenging
Network	- Denial of Service - Sniffing/Spoofing of virtual networks - Malware injection

The Security-SLA should cover all components of the offered Cloud service and their associated vulnerabilities and threats. The architecture of the Cloud IaaS service delivery model is depicted in Figure 7.4, and Table 7.1 shows some of the common threats identified for each component of the architecture. For instance, the virtualization layer, which sits between the Operating System and all the other components and enables the partitioning of resources on the hardware level into shared virtual resources between multi-tenant users, is accompanied by many vulnerabilities within the virtual machines and hypervisor. Some of these vulnerabilities are : the possible formation of covert channels between co-located VMs, unrestricted management of the resources by the VMs, and uncontrolled VM migration [8]. Moreover, data-related vulnerabilities such as its co-location, incomplete deletion, and insecure transfer between the different components could also cause the occurrence of dangerous threats and incidents. To assure their customers, CSPs should describe in their Security-SLAs the different security services that they deploy in order to manage these vulnerabilities and ensure protection against the associated threats, since they are solely responsible for the security of the Cloud IaaS service components. The fundamental security attributes of the CIA triad security model (Confidentiality, Integrity, and Availability) need to be considered when implementing these security services, as we mentioned in our previous work [129]. The confidentiality attribute concerns with protecting the customer's sensitive information and confidential data from unauthorized disclosure ; the integrity attribute consists on protecting the accuracy and validity of data and computations ; and the availability attribute describes the continuous accessibility to the Cloud service and data.

The performance of these security services can be described by the means of relevant and measurable SSLO parameters. To derive these parameters, the following set of criteria can be used : accuracy, functionality, correctness, robustness, coverage, continuity, resource ove-

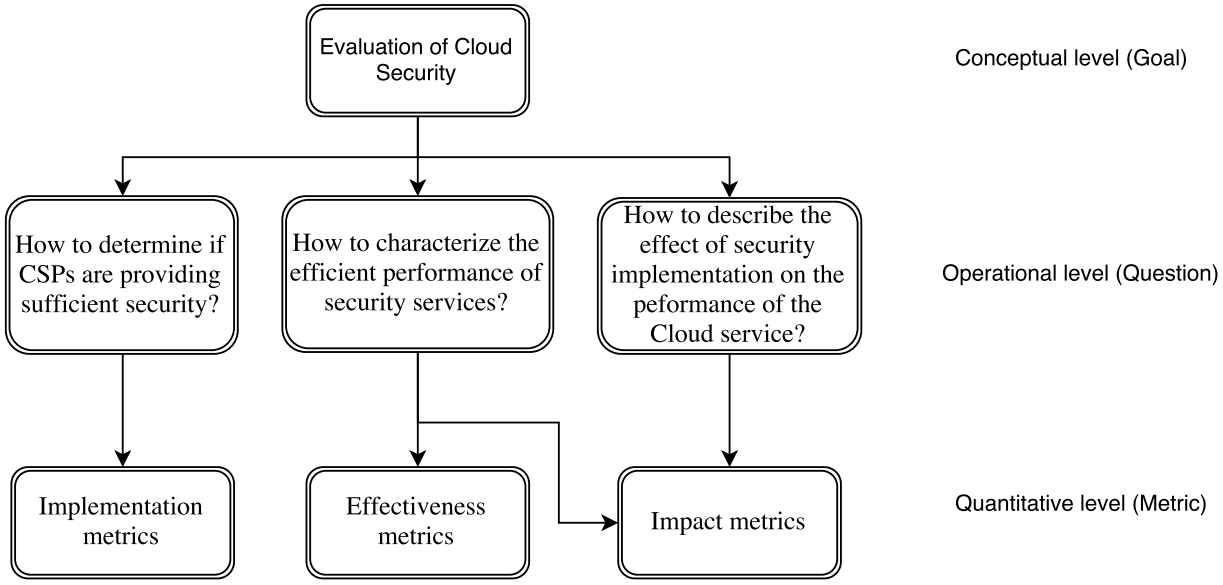


Figure 7.5 The Goal-Question-Metric structure for Security-SLA quantification.

rhead, response time, and failure impact, in order to compensate for the difficulty of exploring the indicators that quantify the security level of the Cloud system. In this paper, the GQM method is used to develop the set of SSLO parameters based on a deep analysis of the Cloud IaaS architecture and security services. This method was originally designed to transform the qualitative testing of software security to an empirical and measurable testing model. Centering the evaluation process on the security services will permit the comparison of CSP's security levels and allow for future Security-SLA standardization. With GQM, we define a measurement model on three levels as shown in Figure 7.5 : on the conceptual level, we describe the goal of the measurement ; on the operational level, we characterize the achievement of this goal by generating a set of questions ; and finally on the quantitative level, we develop a set of measurable parameters that answer these questions [42]. A set of security services and techniques is presented in Table 7.2, along with the developed SSLO parameters. Inspired by the three types of measurements [125] that NIST had set to evaluate the performance of an information security system (implementation, effectiveness, and impact), we divide the proposed SSLO parameters into the following categories based on their role :

— **Implementation SSLO parameters.** They intend to reflect the implementation and deployment of appropriate security mechanisms, policies, controls, and procedures on the different layers of the Cloud architecture. They normally takes a binary or boolean value : 1 (or yes) if the procedure is deployed, and 0 (or no) if not.

— **Performance SSLO parameters.** They help in monitoring the performance and effectiveness of the provisioned Cloud security services. This SSLO parameter is usually a real number or percentage that expresses a benefit aspect of security. Therefore,

Table 7.2 A set of SSLO parameters related to each security service in the Cloud Security-SLA. Parameter type I refers to implementation, P to performance, and C to cost.

Security service	Security techniques and mechanisms	SSLO parameters	Parameter type
Identity and Access Management	- Identity management - Multi-Factor authentication - Federated IDs (e.g., LDAP or OAuth) - Single Sign On (SSO) - Authorization and access control - Authorization policies - XACML access control - Access auditing and logging	- False acceptance and rejection rates of authentication mechanisms - Enforcement of policies on password strength and expiration - Blocking of invalid login attempts - Enabling of client certificate for SSL/TLS - Average response time of transactions - Implementation of risk-based entitlement decisions - Ability to use temporary access credentials - Frequency of review of system users and administrators' entitlements - Frequency of review of access control logs and accounts' activity	C I I I C I I P P
Network security	- IDS/IPS - DDoS mitigation - IPSec VPNs deployment - SSL protection - Firewalls - Traffic isolation	- Configuration of security groups - Configuration of Access Control Lists (ACLs) on virtual interface ports - Intrusions and injection attacks detection success rate - Intrusion detection false positives - Mean-time to discover and mitigate attack - Service latency or response time during attack - Network packets drop rate during attack - CPU load or processing usage of IDS - Network load or bandwidth overhead of IDS - Frequency of network penetration tests	I I P C C C C C C P
Data and storage security	- Secure storage schemes - Encryption techniques - Key management procedures - Data dispersion techniques - Data loss/leakage prevention - Data backup and restore - Data isolation - Data deletion - Secure data migration - Breach reporting and recovery	- Capability of open encryption methodologies - Capability of creation of a unique encryption key per tenant - Internal storage of encryption keys - Encryption key length - Enabling of HTTP Strict Transport Security (HSTS) - Database deployment with SSL protected transactions - Support of secure data deletion - Implementation of data loss/leakage prevention techniques - Percentage of key storage overhead - Storage node online latency in responding to read-write requests - Data backup frequency - Backup restoration time - Number of redundant backup sites	I I I P I I I I C C P C P
Virtualization security	- VMs' interference prevention - Hypervisor-level role-based access control - VM encryption - SSH secure communications - Malware detection - Secure VM migration	- Events monitoring and auditing by hypervisor - Capability of encryption of virtual storage - Validation of VMs - Implementation of encrypted live migration - Data destruction procedures after migration - Frequency of assessment of virtualization vulnerabilities - VM backup, restoration and clean-up capabilities - Slowdown in migration time due to secure migration - Time spent scanning VM images' vulnerabilities - Performance overhead due to running filters on VM images	I I I I I P I C C C
Physical security	- Physical security perimeters - Role-based access control systems	- Monitoring and controlling process service points - Isolation and monitoring of data storage physical points - Monitoring of environmental conditions that affect computer systems	I I I
Data and Computational integrity	- PDP techniques for data validation - Computing integrity checking techniques	- Deployment of applications with computing replication - Encryption of VM images during transport across hypervisor instances - Computation time by PDP techniques - Bandwidth overhead of PDP tokens' transmission - Processing overhead of accountable computation - False positives of computational faults	I I C C C C
Data and service availability	- Incident response plans - Data replication between Cloud nodes - DDoS attack detection system - DoS mitigation - Risk assessment plans - Vulnerability scan	- Multi-failure disaster recovery capability - Capability of infrastructure service fail-over to other providers - Configuration of SIEM incident reporting, analysis and alerting - Frequency and coverage of risk analysis - Frequency of reviewing and testing of business continuity plans - Data redundancy level - Mean-time of incident discovery and recovery - Average time between incident detection and reporting - Percentage of recurring incidents - Cloud failure rate and failure repair time - Time to recover failed tasks - System downtime due to incidents - FRC and DDoS attack detection time - Time to identify attack source - Processing overhead of detection mechanisms - Application response time under attack - Mean-time to mitigate or patch vulnerability	I I I P P P C C C C C C C C C C C C

Table 7.3 Notations used in our security evaluation model.

Notation	Description
$CSP = \{CSP_n,\ n \in \mathbb{N}^*,\ n \leq N\}$	the set of CSPs
$I_n = \{I_{n,j},\ j \in \mathbb{N}^*,\ j \leq J\}$ and $I_b = \{I_{b,j},\ j \in \mathbb{N}^*,\ j \leq J\}$	the sets of implementation SSLO parameters' values offered by CSP_n and baseline values respectively
$P_n = \{P_{n,k},\ k \in \mathbb{N}^*,\ k \leq K\}$ and $P_b = \{P_{b,k},\ k \in \mathbb{N}^*,\ k \leq K\}$	the sets of performance SSLO parameters' values offered by CSP_n and baseline values respectively
$C_n = \{C_{n,l},\ l \in \mathbb{N}^*,\ l \leq L\}$ and $C_b = \{C_{b,l},\ l \in \mathbb{N}^*,\ l \leq L\}$	the sets of cost SSLO parameters' values offered by CSP_n and baseline values respectively
$w_j^I,\ w_k^P$, and w_l^C	the weights of SSLO parameters $I_{b,j},\ P_{b,k}$, and $C_{b,l}$ respectively
$R = \{R_n,\ n \in \mathbb{N}^*,\ n \leq N\}$	the set of resources (VMs) that are shared by CSPs with the federation

security is usually high when the value of such parameter is high.

— **Cost SSLO parameters.** They express the cost of security implementation and the effect of deploying security mechanisms on the performance of the Cloud service. This SSLO parameter is normally a real number or percentage, usually expressing a cost aspect of security, such as the Cloud failure rate. Therefore, security is usually low when the value of such metrics is high.

Normally, when the value of a specific SSLO parameter which was agreed upon in the defined Security-SLA is not met by the CSP, a violation might occur and the customer would have the right to request a compensation as a penalty (e.g., financial). However, in this paper, violations of the Security-SLA will be measured according to the global satisfaction of SSLO parameters, as we will see next.

7.4.2 Security Level Evaluation

The development of SSLO parameters was the first step towards quantifying the Cloud security level. In this section, we first propose a method to evaluate the security level of CSPs. Then, taking into consideration their security reputations, which we compute relatively to their customers' satisfaction, we evaluate the security levels of the federations that could form between them. To evaluate the security levels of CSPs and generate a single numerical value that reflects their security status, we apply a coarse grained aggregation method that is based on comparing their provided SSLO values to a predefined baseline level, defined as follows.

Definition 1. *Cloud Security-SLA Baseline (CSB). It is a version of the Security-SLA that expresses the minimum required security level that a CSP should achieve when providing Cloud services.*

The CSB could be set according to the type of services, and is determined by third party

security experts. The baseline security level is expressed using the same SSLO parameters that we developed, and can be constructed by carefully considering previous Cloud customers' experiences with deploying services to the Cloud. For instance, if an implementation SSLO parameter was found to be not fully essential for the protection of the Cloud service, its value will remain 0. Information related to the CSB will be stored on the broker side and updated occasionally when new Cloud threats are discovered or new SSLO parameters need to be added. Therefore, the security level of CSPs is continuously evaluated against the baseline level to demonstrate compliance and keep the information stored on the brokers' side up to date. When the security baseline is defined and set, the security level of CSPs and their formed federations will be evaluated by measuring their deviation from the baseline level. We assume that the federations will form between the CSPs providing the same type of services, and therefore, the members will be evaluated against the same CSB. The principle motivation behind the definition of a security baseline is to build a reference level that will facilitate the evaluation and comparison of CSPs' security levels and help standardizing the security evaluation process in the Cloud. The notations used in the evaluation of the security level are presented in Table 7.3. The deviation of a Cloud service provider $CSP_n \in CSP$ from the baseline level, denoted by α_n, is computed as follows :

$$\alpha_n = \frac{\sum_{j=1}^{J} w_j^I (I_{n,j} - I_{b,j})}{J} + \frac{\sum_{k=1}^{K} w_k^P (P_{n,k} - P_{b,k})/P_{b,k}}{K} + \frac{\sum_{l=1}^{L} w_l^C (C_{b,l} - C_{n,l})/C_{b,l}}{L} \tag{7.1}$$

where J, K, and L are respectively the numbers of implementation, performance, and cost SSLO parameters defined in the security-SLA, and each term of Equation 7.1 reflects the average weighted relative deviation of the respective type of SSLO parameters from the baseline. To homogenize the evaluation, the values of SSLO parameters offered by CSPs are normalized with respect to the values given in the baseline, since these parameters have different units of measurement. The security level of CSP_n, denoted by SL_n, will be then expressed as follows :

$$SL_n = 1 + \alpha_n \tag{7.2}$$

A positive value of α_n reflects a security level that is higher than the baseline level, whereas a negative value of it expresses a security level that is lower than the baseline level. We consider a CSP secure if it provides a security level that is higher of or equal to the baseline level (e.g., ≥ 1), and insecure otherwise. For instance, considering a percentage like evaluation result, we would say that $CSP_n \in CSP$ has a security level of 20% over or under the defined

baseline if $SL_n = 1.2$ or 0.8 respectively. Therefore, CSP_n is considered to be secure in the first case and insecure in the second.
The weights of SSLO parameters in Equation 7.1 aim at determining their relative significance to the evaluation process, and are computed with the help of security experts based on several factors such as the Cloud service type. For instance, if the deployed Cloud service is a web server, the SSLO parameter “System downtime due to incidents" is more important than the SSLO parameter “False positives of computational faults" since the availability of the service in this case is more critical to the business than the integrity aspect.
Each CSP interacts with its customers by servicing their requested sets of workloads, which could be represented by a number of VMs of different sizes in terms of the offered resources. Each workload entails a set of security requirements that are mapped to specific SSLO values. This mapping can be performed by transforming the qualitative description of customer's requirements into the quantitative parameters of table 7.2. For instance, to ensure service availability, the requirement “Disruption of Cloud services" is proposed, along with five possible descriptive options : absolutely intolerant, intolerant, moderately tolerant, tolerant, and absolutely tolerant. These qualitative values can then be mapped to the quantitative SSLO parameter “System downtime due to incidents" by defining the correspondent intervals of service downtime in hours according to the service's characteristics and nature. What matters in our case, is that a requested Security-SLA will be constructed for each of the customer's workloads, including all SSLO values which it defined. Then, a required security level is computed for each workload relatively to the CSB using Equations 7.1 and 7.2.

7.4.3 Federation Security Evaluation

To evaluate the security level of a Cloud federation, we first start by evaluating the reputation of CSPs with respect to security, which is a parameter that reflects the degree to which CSPs are providing their customers with the promised security level. Let $CSC = \{CSC_h,\ h \in \mathbb{N}^*,\ h \leq H\}$ denotes the set of H Cloud Service Customers that are interacting with a specific CSP, and each customer requests the servicing of a set of workloads W_h. The security of a workload is considered to be satisfied by the CSP if, and only if, the security level that was provided by the CSP is higher than or equal to the workload's required security level, which was promised by the CSP. Thus, if we assume that customer CSC_h will only interact with the CSP if the latter had promised to satisfy the security levels required by its workloads, the security satisfaction Sat_h of the customer can be computed by simply dividing the number of workloads which required security was satisfied when serviced by the CSP by

the total number of workloads during the interaction, as follows :

$$Sat_h = \frac{|W_h^{Sat}|}{|W_h|} \tag{7.3}$$

where $W_h^{Sat} \subseteq W_h$ and W_h^{Sat} is the set of workloads which required security was satisfied by the CSP. Finally, in order to compute the reputation value rep_n of a Cloud service provider CSP_n, that is attributed to its security level, and which will indicate how much the CSP is being honest about the promised security levels, the security satisfaction of all its customers is considered. The following equation evaluates rep_n by averaging the security satisfaction of all the H customers of CSP_n :

$$rep_n = \frac{\sum_{h=1}^{H} Sat_h}{H} \tag{7.4}$$

Before evaluating the security level of a federation, we start by formally defining the federation formation concept in the context of coalition formation games as follows.

Definition 2. *Coalition partition. A coalition structure or partition is a set of M coalitions $\Pi = \{F_m,\ m \in \mathbb{N}^*,\ m \leq M\}$, where each $F_m \subseteq CSP$ is a disjoint coalition such as $\bigcup_{m=1}^{M} F_m = CSP$ and $F_m \cap F_y = \emptyset\ \forall\ y \neq m$.*

The coalitions in our case are called federations. When forming a federation, we assume that the CSPs are already fulfilling the constraints on workload servicing, that is, they are able to provide sufficient resource capacities, and we look beyond this criterion to verify the security status of the members. We define the security level function Sec for the formed federations as a real-valued function such that $Sec:\ F_m \rightarrow \mathbb{R}^+$ as follows :

$$Sec(F_m) = \frac{\sum_{CSP_n \in F_m} R_n rep_n SL_n}{\sum_{CSP_n \in F_m} R_n} \tag{7.5}$$

This function aims at computing the security level of a formed federation based on the security levels and reputations of its members. Multiplying the reputation value of a CSP by its declared security level instead of directly considering the latter compensates for the possible lack of integrity of the CSP when providing the information related to its security deployment and performance, by taking into account its customers' satisfaction. In our case, the resources are shared between CSPs at the VM granularity level, hence the security level of a federation is computed by averaging the security levels of all VMs that are shared with the federation. Therefore, the security level SL_n of a service provider CSP_n is multiplied by

the amount of resources (VMs) R_n which it supplies to the federation, since all its VMs are provided with the level SL_n, which corresponds to the provisioning of a particular Security-SLA, regardless of their types or sizes. Consequently, all CSPs in the federation F_m will provide their federated services with the same security level $Sec(F_m)$, since they are sharing the federation's resources.

7.5 The Security-based Cloud Federation Formation Game

When deciding to form a Cloud federation and sharing their resources, CSPs should consider the security factor, which will lead them to forming secure federations and reducing Security-SLA violations. Coalition formation is a major subject in multi-agent systems, and hedonic games are a popular category of the coalitional cooperative games, and in which profit allocation among the coalition members is not the main problem. In a hedonic game, the players are usually self-interested, and the stability property is guaranteed, that is, when the final partition of coalitions is formed, none of the players will have an incentive to leave its current coalition to join another. In the case of Cloud federation formation, the number of possible coalition structures is too large to permit an exhaustive search for the optimal solution and finding it is a NP-complete problem [104]. Thus, a hedonic game was adopted to model the problem. In this section, the security-based Cloud federation formation process is modeled as a hedonic coalitional cooperative game that aims at producing secure federations, and where each CSP acts as a selfish player when deciding to prefer a federation over another according to the security levels of its members. The appropriate preference function is defined and the properties of the game are analyzed.

7.5.1 Game Model

A coalitional game is a game-theoretical approach that models the interactions among players when they aim at forming groups, and generates a partition of coalitions over the set of players. In the game that we propose, the players are the Cloud service providers and the coalitions to be formed are the Cloud federations. The objective of the players is to join secure federations where the number of insecure CSPs is minimal. When the federation size increases, the probability of finding insecure members in the federation is higher. We introduce the following definitions.

Definition 3. *Grand federation. The grand federation G is formed when all CSPs decide to join one single federation, that is, $G = CSP = \{CSP_1, \ldots, CSP_N\}$.*

Definition 4. *Non-cohesive game. A coalitional game is considered to be non-cohesive [144] when producing a set of disjoint coalitions is preferred over forming a single coalition that groups all the players together.*

In our game, forming the grand federation will entail higher drops in the security levels of its members and increase the rates of Security-SLA violations, since secure and insecure CSPs will exist in the same federation. Therefore, we consider our game to be non-cohesive which goal is to generate a set of disjoint federations that separate secure CSPs from insecure ones. In contrary to Transferable Utility (TU) coalitional games where the utility of the coalition can be transfered and distributed between its members [105], our proposed game is considered of Non-Transferable Utility (NTU) since security can not be distributed among the federation members.

Definition 5. *Hedonic game. A hedonic game is a type of NTU coalitional games where the utility of a player in a coalition depends only on its members, and the formation of coalitions is based on the preferences that the players have over the set of possible coalitions [105].*

We consider our game to be hedonic since it verifies these two conditions. First, the security level of a CSP in a particular federation depends only on the CSPs that are members in that federation as shown in Equation 7.5, and second, we look at security as an enjoyable property which CSPs will consider as the basis for building their preferences over the set of federations.
At any given time, the set of CSPs is divided into a partition $\Pi = \{F_m,\ m \in \mathbb{N}^*,\ m \leq M\}$ with M federations, where each $F_m \subseteq CSP$ is a disjoint federation such as $\bigcup_{m=1}^{M} F_m = CSP$ and $F_m \cap F_y = \emptyset\ \forall\ y \neq m$. Given a federation partition Π, for any $CSP_n \in CSP$, we denote by $F_\Pi(n)$ the federation $F_m \in \Pi$ such that $CSP_n \in F_m$. Each coalition possesses a coalition value, which in our case is the security level of the formed federation that we evaluate using Equation 7.5 relatively to the defined CSB by taking into account the security levels of the federation members and the amount of resources each member shares with the federation. The Cloud federation formation game proposed in this paper is defined as follows.

Definition 6. *The security-based Cloud federation formation game. It is the pair $(CSP, \succeq)$, where CSP is the set of Cloud Service Providers in the game, and $\succeq_n$ is a reflexive, complete, and transitive preference relation on the set of all federations that CSP_n can form based on the security level.*

Based on this definition, for all $CSP_n \in CSP$ and for all $E, E' \in \Pi$, we define $\succeq_n$ as

$$E \succeq_n E' \ \Leftrightarrow\ U_n(E) \geq U_n(E') \tag{7.6}$$

where $U_n(E)$ and $U_n(E')$ are the utilities of CSP_n in federations E and E' respectively. The utility function can be defined as follows :

$$U_n(E) = \begin{cases} Sec(E) & \text{if } E \notin h_n \\ 0 & \text{otherwise} \end{cases} \tag{7.7}$$

where h_n is a history set where CSP_n stores the identities of the federations which was a part of in the past. Hence, the preference relation assigns a utility value of 0 for any federation that already exists in h_n, in order to avoid visiting the same federation more than one time during the federation formation process (a similar concept has also been adopted in [145, 146, 86]). Based on the defined preference relation, each CSP will compare the set of possible federations and indicate its intention to be a part of one of them, which means that it will prefer the federation that will grant the higher security level and avoid joining federations with insecure members. $E \succeq_n E'$ indicates that CSP_n prefers to be a member of federation E than to be a member of federation E', or at least prefers them both equally. The strict counterpart of the relationship denoted by $\succ_n$ implies that CSP_n strictly prefers to be a member of E over E'.

7.5.2 The Federation Formation Algorithm

In this section, we propose a hedonic federation formation algorithm that achieves the goal of our security-based federation formation game, and that can be implemented in a distributed fashion. The algorithm is illustrated in Algorithm 2, and takes as input the initial federation partition Π_c, the set R of resources that CSPs will share with the federations, and the security levels of CSPs computed using Equations 7.1 and 7.2, and outputs the final federation partition Π_f.

The following hedonic shift rule [145], which is selfishly executed by each CSP when moving between federations, is used as the basis of the proposed federation formation algorithm : given a federation partition Π on the set of service providers CSP and a preference relation $\succ_n$, any $CSP_n \in CSP$ decides to leave its current federation $F_\Pi(n)$ to join another one $F_m \in \Pi \cup \emptyset$ if, and only if, $F_m \cup \{CSP_n\} \succ_n F_\Pi(n)$, that is, if the utility of CSP_n in the new federation is higher than its utility in the current one. In other words, if the security level of the new federation is higher than the security level of the current one. The actions presented in the algorithm are executed asynchronously and independently by each CSP during the federation formation process. Starting from a federation partition Π_c, each service provider CSP_n evaluates the utility that it will acquire by joining a new federation F_m and compares it to the one it has from its current federation $F_{\Pi_c}(n)$, then applies the described shifting rule if

Algorithm 2 The security-based Cloud federation formation algorithm.

Input:
- The current federation partition $\Pi_c = \{F_1, \ldots, F_M\}$
- The set $R = \{R_1, \ldots, R_N\}$ of available resources
- The set $SL = \{SL_1, \ldots, SL_N\}$ of CSPs' security levels

Output:
- The final federation partition Π_f

1: **procedure** FEDERATIONFORMATION(Π_c, R, SL)
2: $h_n \leftarrow$ *empty set*
3: **for all** $CSP_n \in CSP$ **do**
4: **for all** federations $F_m \in \Pi_c \cup \emptyset$ **do**
5: evaluate $Sec(F_m \cup \{CSP_n\})$
6: **if** $U_n(F_m \cup \{CSP_n\}) > U_n(F_{\Pi_c}(n))$
$\Leftrightarrow F_m \cup \{CSP_n\} \succ_n F_{\Pi_c}(n)$ **then**
7: - leave the current federation $F_{\Pi_c}(n)$
8: - join the new federation $F_m \cup \{CSP_n\}$
9: - update the federation partition :
10: $\Pi_{c+1} = (\Pi_c \backslash \{F_{\Pi_c}(n), F_m\})$
11: $\cup (F_{\Pi_c}(n) \backslash \{CSP_n\}, F_m \cup \{CSP_n\})$
12: - Update history set : $h_n = h_n \cup \{F_{\Pi_c}(n)\}$
13: **else**
14: CSP_n remains in $F_{\Pi_c}(n)$:
15: $\Pi_{c+1} = \Pi_c$
16: **end if**
17: **end for**
18: **end for**
19: **return** $\Pi_f = \Pi_{c+1}$
20: **end procedure**

the utility of CSP_n in the new federation F_m exceeds its utility in $F_{\Pi_c}(n)$, and automatically updates its history set h_n by adding the federation $F_{\Pi_c}(n)$ that it had left (lines 6 to 12). Otherwise, CSP_n remains in its current federation and the partition Π_c remains unchanged. The federation formation procedure is repeated until all CSPs converge to a final partition Π_f that satisfies Nash-stability.

The execution of the algorithm is repeated periodically to reflect the updates of CSPs' security levels, reputation values, and the set of their available resources. However, it is deemed necessary to provide the appropriate mechanisms for : state retrieval (e.g., [147]), which permits to each CSP to obtain the current federation partition, and atomic state update (e.g., [148]), which guarantees that the current federation partition will remain unchanged while the CSP is executing the actions in the algorithm.

The main computational complexity of the proposed algorithm lies in the hedonic shift rule that the CSP will apply to switch from a federation to another, and depends on the number of federations in the partition Π_C. Let $|\Pi_C|$ be the number of federations in Π_C, the computational complexity can then be expressed by $O(|\Pi_C|)$. The worst computational complexity will occur in the case where Π_C contains N federations, that is, every CSP is acting alone in a separate coalition. Since the federation formation procedure is repeated until reaching the final partition, the computational complexity of the game will be defined by the Bell number in the worst case scenario (discussed in the following subsection).

7.5.3 Game Analysis

In this section, the properties of the proposed security-based federation formation game are analyzed. More specifically, we demonstrate the property of convergence of the federation formation algorithm to a final solution and the property of stability of the produced final solution.

Definition 7. *Nash-Stability. A partition of federations* Π *is considered Nash-stable if no CSP has an incentive to move from its current federation* $F_\Pi(n)$ *to join a different one, nor to act alone. That is, for* $\Pi = \{F_m,\ m \in \mathbb{N}^*,\ m \leq M\}, \forall\, CSP_n \in CSP,\ F_\Pi(n) \succeq_n F_m \cup \{CSP_n\}$ $\forall\ F_m \in \Pi \cup \emptyset$.

Definition 8. *Individual Stability. A partition of federations* Π *is considered individually stable if no CSP can benefit by moving from its current federation* $F_\Pi(n)$ *to another one without negatively affecting the members of the latter. That is,* $\nexists\, CSP_n \in CSP$ *and* $F_m \in \Pi \cup \emptyset$ *such that* $F_m \cup \{CSP_n\} \succ_n F_\Pi(n)$ *and* $F_m \cup \{CSP_n\} \succeq_{n'} F_m,\ \forall\ CSP_{n'} \in F_m$.

The following propositions are made with regard to the proposed federation formation game.

Proposition 1. *Convergence. Starting from any initial partition of federations* Π_0*, Algorithm 2 converges to a final partition* Π_f *consisting of a number of disjoint federations.*

Proof. Every application of the hedonic shift rule will transform the current partition Π_c into a new partition Π_{c+1} that has not been visited in the past (according to Equation 7.7), until reaching the final partition Π_f. Since the number of transformations is finite, and at most, is equal to the number of partitions defined by the N^{th} Bell number B_n for N CSPs as follows :

$$B_n = \sum_{i=0}^{N-1} \binom{N-1}{i} .B_i \quad \text{for } N \geq 1 \text{ and } B_0 = 1 \tag{7.8}$$

the sequence of transformations will always be limited and will converge to a final partition Π_f. □

Proposition 2. *Nash-Stability. Any final partition Π_f produced by Algorithm 2 is a Nash-stable federation partition.*

Proof. This proposition can be proved by contradiction. Let us assume that the final partition of federations Π_f is not Nash-stable. Then, there exists a service provider CSP_n that prefers to leave its current federation $\Pi_f(n)$ and join another one F_m, that is, $F_m \cup \{CSP_n\} \succ_n \Pi_f(n)$. Therefore, CSP_n will perform the shifting rule and the final partition will change to a new one Π'_f such as $\Pi'_f \neq \Pi_f$ which contradicts with Proposition 1 that states that the partition Π_f is the final outcome of the federation formation algorithm. Hence, we conclude that the algorithm always converges to a Nash-stable federation partition. □

Proposition 3. *Individual Stability. Any final partition Π_f that is produced by Algorithm 2 is individually stable.*

Proof. Since we have already proven that Algorithm 2 converges to a Nash-stable partition of federations, this implies that it also converges to an individually stable partition [105]. □

To illustrate the individual stability of the partitions generated by our proposed game, the following numerical example is considered. We study the stability of a security-based federation formation game between 7 players such that $SL = \{0.72, 1.1, 1.3, 0.55, 1.4, 1.2, 0.5\}$ and $R = \{100, 200, 150, 320, 240, 170, 450\}$. The initial partition of federations is shown in the first row of Table 7.4. Every row shows the partition Π_{c+1} that is generated at the end of every execution of Algorithm 2 by each of the CSPs, and the last row shows the final partition Π_f. As we notice in Π_f, CSP_1 and CSP_4 both have incentive to leave their current federation and join either one of the other two federations, since $\{CSP_2\} \cup \{CSP_1\} \succ_1 \{CSP_1, CSP_4\}$ and $\{CSP_3, CSP_5, CSP_6, CSP_7\} \cup \{CSP_1\} \succ_1 \{CSP_1, CSP_4\}$, and $\{CSP_2\} \cup \{CSP_4\} \succ_4 \{CSP_1, CSP_4\}$ and $\{CSP_3, CSP_5, CSP_6, CSP_7\} \cup \{CSP_4\} \succ_4 \{CSP_1, CSP_4\}$. If one of them decides to move from its current federation and join a new one, every member of the latter federation will be worsen off. For instance, the federation $\{CSP_3, CSP_5, CSP_6, CSP_7\}$ is preferred by each of its members over the federations $\{CSP_3, CSP_5, CSP_6, CSP_7, CSP_1\}$ and $\{CSP_3, CSP_5, CSP_6, CSP_7, CSP_4\}$ since it is more secure. Similarly, CSP_2 will prefer to stay non-cooperative over joining CSP_1 or CSP_4 in a federation. Therefore, the individual stability in the final partition is achieved.

7.6 Experimental Results and Analysis

In this section, we study the performance of the security-based Cloud federation formation game. First, we explain the experimental setup used in our simulation, then we analyze

Table 7.4 A numerical example of the execution of Algorithm 2.

Federation partition	Federations' security levels
$\{\{CSP_1, CSP_2\}, \{CSP_3, CSP_4\}, \{CSP_5, CSP_6, CSP_7\}\}$	$\{0.9733, 0.7894, 0.8895\}$
$\{\{CSP_2\}, \{CSP_3, CSP_4\}, \{CSP_5, CSP_6, CSP_7, CSP_1\}\}$	$\{1.1000, 0.7894, 0.8719\}$
$\{\{CSP_2\}, \{CSP_3, CSP_4\}, \{CSP_5, CSP_6, CSP_7, CSP_1\}\}$	$\{1.1000, 0.7894, 0.8719\}$
$\{\{CSP_2\}, \{CSP_3, CSP_4\}, \{CSP_5, CSP_6, CSP_7, CSP_1\}\}$	$\{1.1000, 0.7894, 0.8719\}$
$\{\{CSP_2\}, \{CSP_3\}, \{CSP_5, CSP_6, CSP_7, CSP_1, CSP_4\}\}$	$\{1.1000, 1.3000, 0.7914\}$
$\{\{CSP_2\}, \{CSP_3, CSP_5\}, \{CSP_6, CSP_7, CSP_1, CSP_4\}\}$	$\{1.1000, 1.3615, 0.6510\}$
$\{\{CSP_2\}, \{CSP_3, CSP_5, CSP_6\}, \{CSP_7, CSP_1, CSP_4\}\}$	$\{1.1000, 1.3125, 0.5437\}$
$\{\{CSP_2\}, \{CSP_3, CSP_5, CSP_6, CSP_7\}, \{CSP_1, CSP_4\}\}$	$\{1.1000, 0.9505, 0.5905\}$

the results. The goal is to show that when CSPs consider the security factor when forming federations, they tend to form federations that are more secure and refrain from joining federations that will cause high security loss. Consequently, security-based federation formation will reduce the rate of Security-SLA violations and their severity.

7.6.1 Experimental Setup

The security-based federation formation game is implemented on a 64-bit Windows 7 machine equipped with an Intel Core i7-3612QM CPU @ 2 :10 GHz Processor and 12 GB RAM. Since we haven't found in the literature any other approach that discussed Cloud federation formation based on security, and we found that other existing approaches are not directly comparable to ours, we decided to compare our model with the following two models : the grand federation model which consists on forming the grand federation between all CSPs without any consideration to any factor, and the random federation formation model which consists on randomly grouping CSPs together and forming federations of relatively bigger size than when considering a specific factor during formation (e.g, security). Throughout the simulation, we vary the percentage of insecure CSPs from 10% to 70% and compare the performance of the three models, starting from a randomly generated initial partition of federations. We implement the security-based federation formation algorithm using MATLAB, where CSPs are simulated as objects, each having an offered security level, a required security level defined by its customers, a reputation value, and a number of VMs to be shared with a federation.

Cloud Security-SLA is still a concept that needs to be continually developed to reach standardization. Security-SLAs do not exist today in the quantitative form that we propose in this paper, but they might exist, and not widely, in a more primary version. Either way, the

Table 7.5 Experimentation parameters.

Parameter	Description	Min	Max
$I_{n,j}$ and $I_{b,j}$	Implementation SSLO parameters	0	1
$P_{n,k}$ and $P_{b,k}$	Performance SSLO parameters	60	95
$C_{n,l}$ and $C_{b,l}$	Cost SSLO parameters	1	5
R_n	Number of shared VMs	100	400
rep_n	Reputation of CSPs	0.6	0.95

transparency of CSPs about what they state and include in their Security-SLAs is relatively shy, which makes it challenging to collect information related to security deployment and performance. Therefore, we decided to generate our own data for the sake of these experiments. The values of SSLO parameters for each CSP and for the baseline level, along with other experimentation parameters, are randomly generated as shown in Table 7.5. The experiment is done with a total of 100 CSPs and the evaluation of security level is performed based on a total of 120 parameters divided between the three categories : implementation, performance, and cost. To reduce complexity, we assume that all parameters have the same importance to the evaluation process, hence parameters' weights are not considered in our experiments.
The two main aspects that we consider in our performance study are the effect of the security-based federation formation on the initial security levels of CSPs, and the Security-SLA violations caused by the federation formation. We aim at showing that our model reduces the number of insecure federations by grouping secure CSPs together on one side and insecure ones together on the other side, and generating federations of small size. We show that this will result in lower rates and severity of Security-SLA violations.

7.6.2 Experimental Results

First, and before comparing the performance of our security-based federation formation model to the other two models, we study its efficiency in conserving the security of CSPs while forming the federations, by producing federations with security as close as possible to that of its members when they are acting alone. Figure 7.6 shows for every CSP, its initial security level when acting alone, and the security level of the final federation where it ended up. The figure was generated by executing our model on a set of 100 CSPs divided between 50% secure and 50% insecure, that is, 50 CSPs with a security level above 1 (above the baseline) and 50 CSPs with a security level below 1 (below the baseline). We notice that our model was successful in separating secure from insecure CSPs when forming federations. All secure

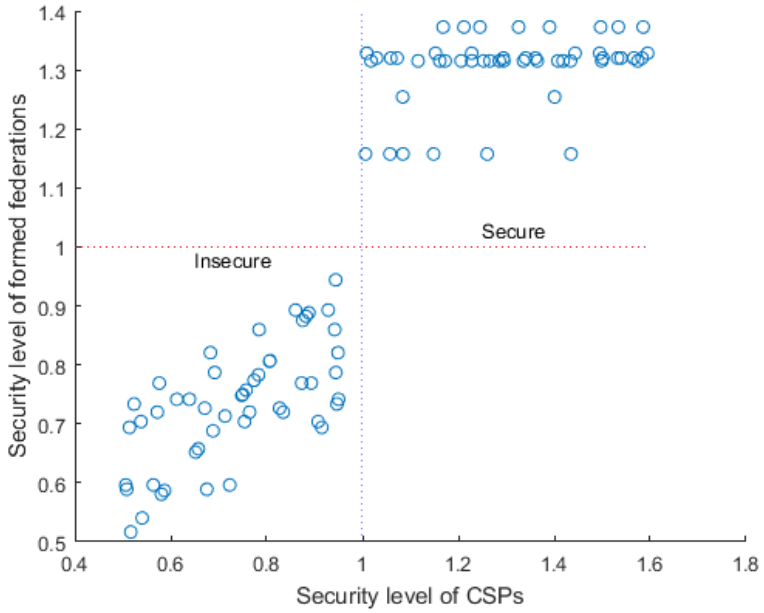
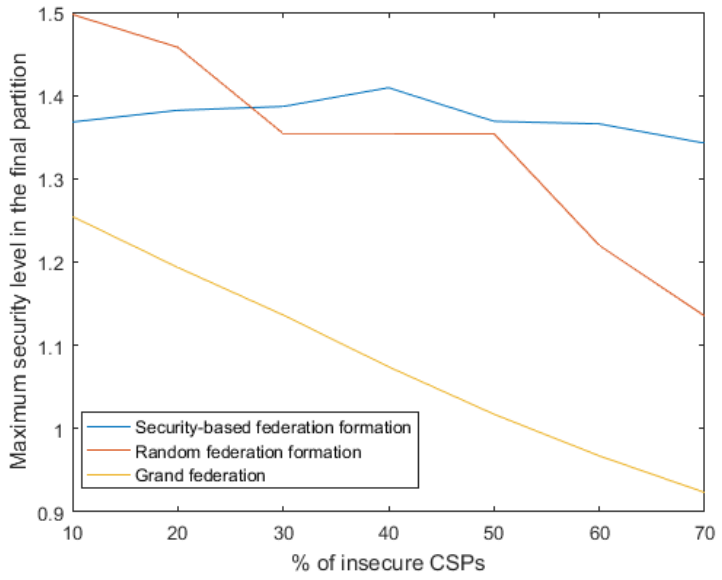
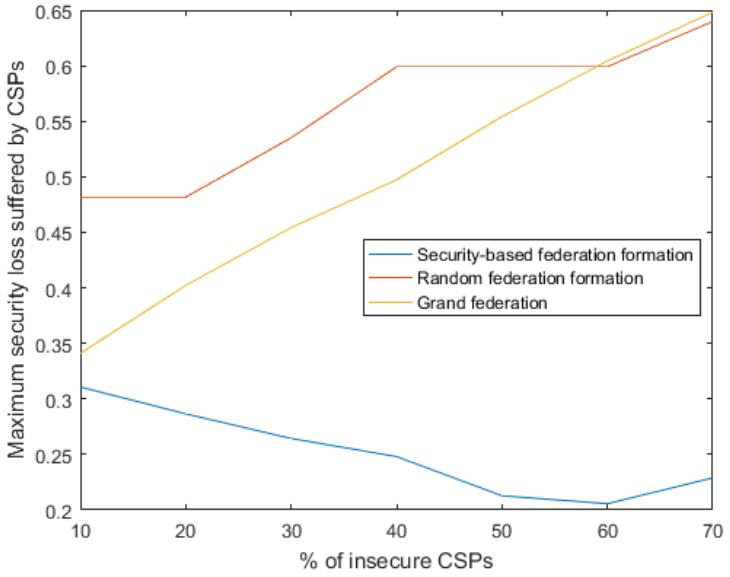


Figure 7.6 The difference between CSPs' initial security and their security in the final partition.

CSPs ended up joining secure federations while insecure ones ended up forming insecure federations. Even though, for some CSPs, the initial security level was not fully conserved, they eventually did not end up forming insecure federations. For instance, some CSPs initially had a security level of up to 1.5, and they ended up in federations which security level is between 1.3 and 1.4, which is relatively acceptable as long as they are still secure.
The effect of applying our security-based model on CSPs' security levels is studied in Figure 7.7. We illustrate in Figure 7.7a the maximum loss in security that a CSP can suffer when forming a federation while increasing the number of insecure CSPs in the initial partition. The results are shown for the three models : the proposed security-based model, the random formation model, and the grand federation model. As we can see in the figure, the security-based model helps producing federations while keeping a low level of security loss (less than 30%), whereas higher security loss occurred with the other two models. When the percentage of insecure CSPs increased, the loss level caused by applying our model decreased by up to 8% in the final partition, which is conform with the results in Figure 7.6, demonstrating that our model keeps grouping together the CSPs with relatively close security levels. The maximum occurred security loss increased with the other two models from 35% when applying random federation formation and 48% when forming the grand federation to reach 65% when the percentage of insecure CSPs reached 70%, which proves that these two models are not suitable for forming secure federations that respect the initial security levels of CSPs.



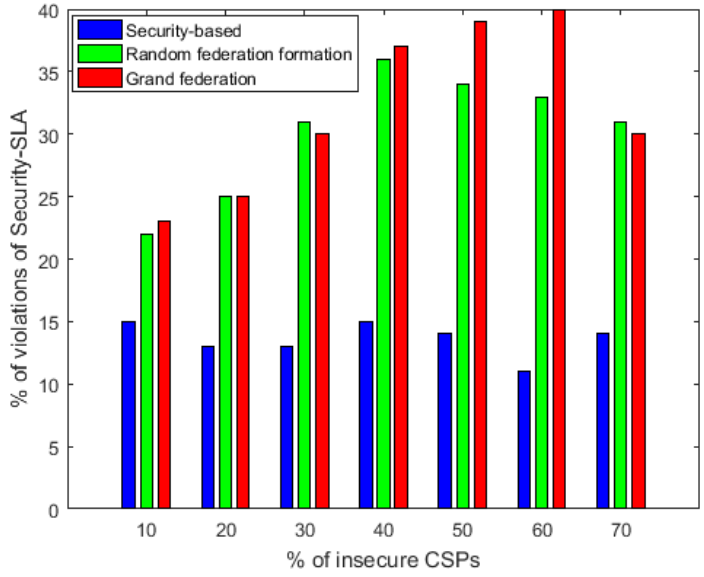
(a) Maximum security loss suffered by CSPs when joining federations.



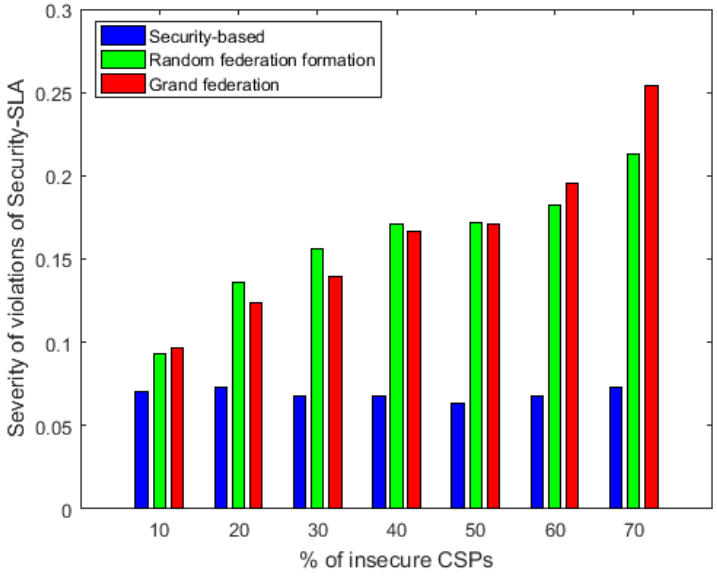
(b) Maximum security level in the formed federations.

Figure 7.7 The effects of federation formation on CSPs' security levels.

Figure 7.7b shows the maximum security level that exists in the federations of the final partition. First, it is clear and expected that the security level of the grand federation will keep dropping when the percentage of insecure members increases. Second, the main criteria that should be considered in this case to establish a fair comparison between the security-based and random formation models is the stability of the maximum security level. For instance,



(a) The variation of Security-SLA Violation rates between the three models.



(b) The variation of Security-SLA Violation severity between the three models.

Figure 7.8 Comparison of Security-SLA violation rates and severity between the three models.

the maximum security level obtained in the final partition when applying random formation rapidely decreased when the percentage of insecure members in the initial partition increased, whereas the security-based formation model conserved approximately the same maximum security level (up to 1.42) indifferently from the increase in the percentage of insecure members.

This shows that our model will keep generating secure federations by separating secure from insecure CSPs.

The variation in CSPs' initial security levels between being in a non-cooperative state and being in a federation is directly related to the occurred Security-SLA violations. Figure 7.8 illustrates the difference between the three models with regard to respecting the agreed upon Security-SLA. In Figure 7.8a, we show the variation in the rates of Security-SLA violations while increasing the percentage of insecure CSPs in the initial partition. The results demonstrate the efficiency of our model in producing federations that guarantee minimum violation rates (between 11 and 15%) comparing to the other two models (between 22 and 40%). The reason is that violations are correlated with the drop in CSPs' initial security levels, and usually occur when the security level of the federation where the CSP exists in the final partition is lower than the security level that was promised to its customers when accepting to service their workloads. For example, when the percentage of insecure CSPs reached 60%, the rate of violations was minimum (11%), which corresponds to the minimum loss in security (21%) as shown in Figure 7.7a.

Figure 7.8b measures the severity of occurred Security-SLA violations when applying the three models, and while varying the percentage of insecure CSPs in the initial partition. We define the violation severity by :

$$Violation\ Severity = \frac{provided\ security\ level}{promised\ security\ level} \tag{7.9}$$

where the provided security level is the same as the federation security level where the CSP exists in the final partition, and the promised security level is the one that the customer requested for its workload. Even though our model does not completely avoid security violations, it achieves federation formation while keeping a minimum and stable level of violation severity (maximum of 7%), whereas with the other two models, the severity level continued to increase with the increase in the percentage of insecure CSPs. This is mainly due to the fact that random federation and grand federation formation do not consider the security factor when grouping CSPs into federations, thus, secure CSPs end up joining federations with high number of insecure CSPs which affect the overall security level of the formed federation, and consequently, the provided security level of its members. In real life scenarios, the rate of violations is not as critical as their severity, since the severity of the violation is what controls the applied penalty cost. For instance, considering the cost SSLO parameter "Cloud failure rate", if the promised value was determined to be not higher than 5%, a provided value of 15% will result in a violation that is more severe than a violation resulted from a provided value of 7%, hence, the cost of the applied penalty will be higher in the first case.

In Figure 7.9, the average size of federations in the final partition Π_f is illustrated. As we can see, our model tends to group CSPs into federations of small size (up to 2 or 3 CSPs per federation), in contrast to forming the grand federation which will group all CSPs together in one single federation. The intention from measuring the size of the final formed federations is to show that our model distributes the CSPs onto federations according to their security levels, where CSPs of relatively close security levels end up together in the final federation. For instance, if the average federation size in the final partition was higher than the one showed in the figure, the probability to encounter insecure CSPs in a federation of secure ones would increase. Therefore, by considering the security factor, the proposed federation formation algorithm works in a way that conserves a stable federation size and refrains from grouping insecure CSPs with secure ones.

Finally, we aim at studying the execution time of the game. In Figure 7.10, we present the average results of fifty runs along with the standard deviations with respect to the number of CSPs involved in the game. We notice that the computational time is relatively small comparing to an exhaustive search approach. For instance, when the number of CSPs was equal to 8, the algorithm was exploring 57 federations on average instead of all the 255 possible federations before converging to the final partition. This demonstrates the scalability of our model when applied on a high number of players.

We also find it necessary to discuss other security and performance factors that would be influenced by the application of our proposed security-based federation formation model, and which we did not numerically consider in our experiments. For instance, by considering the security levels of CSPs when forming Cloud federations, the shared environment which CSPs will create in their federation will somewhat guarantee the same availability level that CSPs were providing in their respective Security-SLAs, along with the other security aspects that CSPs agreed on protecting. In addition, attack propagation will be limited since secure CSPs are now federating their workloads to each other, and sharing their performance and expertise in attack detection and mitigation, away from insecure environments where the probability of a threat occurrence is higher. Consequently, security risk will also be reduced, since the risk is directly proportional to the probability of threat occurrence [149]. Therefore, we can conclude that security-based Cloud federation formation will allow the federation members to safely and peacefully enjoy each other's company and share their resources in a more secure and efficient fashion.

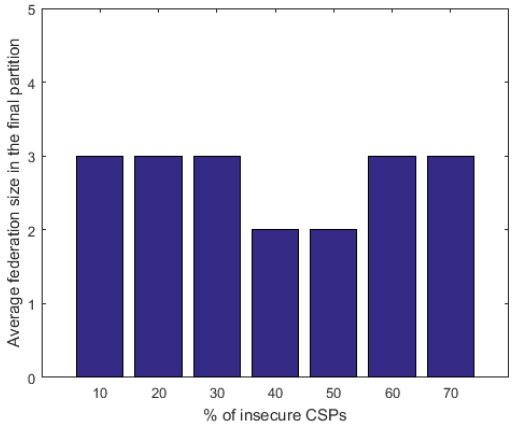


Figure 7.9 Federation average size in Π_f.

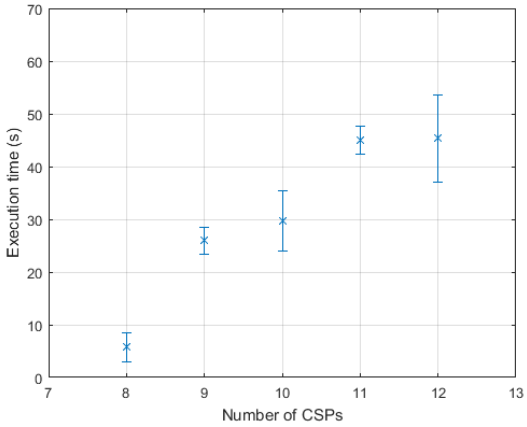


Figure 7.10 Computational time of the proposed federation formation algorithm.

7.7 Conclusion

In this paper, the Cloud federation formation problem was addressed from a new perspective. The originality of our work lies within three aspects : the quantitative Cloud Security-SLA parameters that we developed, the proposed method that evaluates the security level of CSPs and federations, and the security-based Cloud federation formation game that we modeled and studied. It is also important to note, that considering Cloud customers' security satisfaction and CSPs' reputation during the evaluation process introduced an added value to the accuracy of the model by compensating for the miss-leading information about CSPs' offered security levels. We found it fair enough to consider CSPs' security levels along with their provided shares of the federation resources when evaluating the security level of a federation. The proposed federation formation algorithm enabled the CSPs that offer relatively close security levels to be grouped together in the same federation, while reducing their loss in security and minimizing the rates and severity of occurred Security-SLA violations.
The construction of the Security-SLA baseline that we used as a reference level when evaluating security could be discussed with more details in future work. Building an efficient and measurable Security-SLA is also a critical mission that is far from being complete. We are also planning to consider, along with security, other significant factors that could influence the Cloud federation formation process (e.g., costs, QoS), and study the trade-off that could exist between these factors, since defining utility functions that combine multiple aspects altogether is quite challenging.

CHAPTER 8 ARTICLE 5 : SECURITY RISK-AWARE RESOURCE ALLOCATION AND PROVISIONING IN CLOUD COMPUTING

Talal Halabi and Martine Bellaiche
Submitted to *IEEE Transactions on Cloud Computing*, June 2018

Abstract

The interest in exploiting the advantages of Cloud Computing technology is on the rise. However, businesses and organizations still hesitate to fully migrate their services and data to the Cloud due to its lack of security assurance. Cloud Service Providers (CSP) are urged to exert the necessary efforts to boost their reputation and improve their trustworthiness. Unfortunately, the uniform implementation of advanced security solutions across all their data centers is not the ideal solution, since customers' security requirements are usually not monolithic. We believe that the answer lies in the deep integration of the security risk factor into the process of resource allocation and provisioning on Cloud infrastructures with the objective of reducing the potentiality of security threats and limiting security attacks. CSPs could offer their security solutions as explicit and heterogeneous configurations, and eventually evaluate the security risk that customers would introduce to their infrastructures according to these configurations. In this paper, the problem of resource allocation in an InterCloud setting is formulated from a security risk perspective. It is then solved using two different metaheuristics approaches from the family of evolutionary computation, which according to the results, showed the ability to function in online mode and demonstrated scalability.

8.1 Introduction

The adoption of Cloud Computing technology is on the rise. Many organizations are deciding to shift their businesses to the Cloud to benefit from the scalability, resilience, and cost reduction features. Recent statistics [1] showed that the total worldwide Cloud IT infrastructure revenue has almost tripled in the last couple of years, while the traditional IT infrastructure revenue continues to decline. According to a new study [11], it is expected that in only 15 months, 80% of all IT budgets will be invested in Cloud applications and solutions. However, the migration to the Cloud is still cautious and limited to specific types of applications due to the increased level of risk caused by outsourcing customers' services and data to third

parties. Every year, the Cloud market giants witness many security incidents and breaches causing Cloud outages and failures, which affects the production process and results in lost data and revenue [12].
Cloud Computing involves many potential security threats inherited from its architectural model and technical properties like virtualization, multi-tenancy, and data distribution. These threats include data breaches, data loss, and denial of service [70], and are usually less likely to cause serious problems and damages in traditional infrastructures. Security is one of the principal driving factors of the Cloud market, especially for businesses that deal with sensitive information. Cloud Service Providers (CSPs) are expected to maintain the security of the Cloud service and protect its availability. According to the study in [11], the rate of organizations that completely trust the public Cloud infrastructures today to protect their data is only at 23%. This shows that CSPs are urged to invest in more resources and efforts to deploy effective security solutions and provide advanced security features to increase the protection of the confidentiality, integrity, and availability aspects on their infrastructures.

8.1.1 Problem Definition

The CSP is usually responsible for securing her Cloud services and providing basic security capabilities. However, the implementation of advanced security solutions normally introduces additional costs to the CSP's budget, and does not necessarily generate higher profit. These costs include the price of security infrastructure (e.g., firewalls, antivirus software, Intrusion Detection Systems, etc), salaries of security architects, and the expenses of security training programs. Investing in security is not necessarily a priority for the CSPs, but has apparently become indispensable to boost the migration toward the Cloud business model. The return on security investment in an IT infrastructure is not usually measured in terms of the achieved revenue, but as a function of damage reduction, i.e, the decrease in loss expectancy due to security incidents and the rate of threat occurrence. According to the study in [14], the average total cost of a data breach is around 4 million $. This gives an idea about how crucial is to secure a Cloud infrastructure that hosts thousands of services and huge amounts of users' sensitive data. However, uniformly securing the Cloud infrastructure and increasing service charges might not be the ideal solution, since customers do not all have the same security requirements, and some of them might not be ready to invest more to have access to the secure services when they only require the deployment of basic security capabilities to protect their data. In public Cloud infrastructures, customers' nature could be very different, and the resources which they provision in the Cloud are usually for different purposes and will require distinct security levels.
To reduce the cost of security investments, the CSP might want to offer her security solutions

as explicit and heterogeneous configurations to be purchased by the customers to secure their services. This strategy will permit her to evaluate the security risk that customers would introduce to her infrastructure and grant her more control over the allocation of her resources by giving the priority to customers who are ready to invest in security and who will introduce lower security risk to the infrastructure. Deploying unsecured services will increase the security risk level within the CSP's infrastructure, since a single attack on a single service could be sufficient to bring down the whole Cloud data center due to increased data distribution and multi-tenancy, and cause irreversible damages to customers' data. This will also implies the violation of the Security Service Level Agreement (Security-SLA) [150, 151], which is the official contract that governs security management between service providers and consumers and helps in guarantying their respective rights, and violating it will entail applied penalties and compensations. Therefore, the integration of the security risk factor into the process of resource provisioning and allocation in the Cloud becomes the key to provide securer services and limit the potentiality of security threats.
Current Cloud resource allocation frameworks do not usually consider the security risk factor. Some of them are energy-aware, and others are profit-oriented, which aim at allocating as much requests as possible, regardless of how much secure the infrastructure will remain. Security is often compromised by the constraints on high resource utilization. If the security risk presented by customers' requests is efficiently evaluated, the CSP would be able to consider it in the allocation process and more importantly, to continuously monitor the security status of her infrastructure. The objective of maximizing the generated profit from selling the Cloud resources should be accompanied by the constraints on acceptable security risk levels within the Cloud. A CSP possessing several data centers would need to know how to optimally allocate her customers' requests, in a way that maximizes her returns without violating the critical security risk level that she had defined for her infrastructure. Besides, performing a security risk assessment of customers' requests before allocating them could help the CSP in differentiating between a malicious or untrusted and benign customer, and providing the allocation priority to the latter.

8.1.2 Contributions

What we propose in this work is an approach for security risk-aware resource allocation in the Cloud. Our contributions are described as follows :

— First, we identify the different security risk factors in a Cloud infrastructure using the Hierarchical Holographic Modeling (HHM) framework [46], which is a comprehensive theoretical framework for modeling complex systems, and propose a set of security parameters that could help the CSP in securing her Cloud environment.

— Second, we perform a relative evaluation of the security risk presented by customers' requests according to different security configurations that could be offered by the CSP. Based on this evaluation, we model the problem of security risk-aware resource allocation in an InterCloud infrastructure, which consists in maximizing the infrastructure's resource utilization while keeping its security risk level below a specific critical value.
— Third, we propose to solve the defined allocation problem using two different metaheuristics approaches from the family of evolutionary computation : the Genetic Algorithm (GA) and the Artificial Bee Colony (ABC) optimization.
— Finally, we implement the two algorithms and perform a set of experiments, which results showed that the algorithms are able to achieve a good approximation of the optimal solution and to perform an online allocation of customers' requests.

8.1.3 Paper Organization

The remainder of the paper is organized as follows. Section 8.2 discusses the literature review that is related to the addressed topic. Section 8.3 describes the proposed approach to security risk assessment. In Section 8.4, the problem of Cloud resource allocation is modeled from a security risk perspective. In Section 8.5, the defined problem is solved using two different Evolutionary Computational approaches. In Section 8.6, experimentation is conducted and results are analyzed. Finally, section 8.7 concludes the paper.

8.2 Literature Review

This paper aims at designing a resource allocation model for Cloud Computing infrastructures based on the integration of security risk evaluation. Security evaluation of Cloud systems is currently an active area of research. The research on risk assessment in Cloud Computing is also continuously evolving, and authors are trying to find efficient solutions that accurately estimate the Cloud security risk. Chopra et al. [44] studied the risk of migrating data and applications to the Cloud and performed a qualitative analysis of the associated risks prior to, during, and after the migration process. Tang et al. [45] modeled the risk identification phase in Cloud Computing based on the HHM framework [46] and used the fuzzy set theory to compute the probability of security risk. Ben Aissa et al. [47] proposed a model that estimates a system's security based on quantifying the costs to stakeholders using the mean failure costs metric. The challenge in their model lies in the difficulty of accurately quantifying the stakes, dependability, impact, and threat matrices that they define. Shameli and Cheriet [48] proposed a quantitative and iterative approach for evaluating the security

risks associated with the Cloud platform using a fuzzy multi-criteria decision making technique, but without identifying those risks. Tanimoto et al. [49] also extracted and analyzed the Cloud risk factors and used a risk matrix that classifies a risk into four kinds : transference, mitigation, acceptance, and avoidance, in order to approximate the asset, threat, and vulnerability values according to the generation frequency and degree of incidence. Finally, Djemame et al. [149] developed a risk assessment framework for the Cloud, which evaluates the risk during service deployment and in run-time, with more focus on performance and Quality of Service (QoS) than security.

The allocation of security resources in Cloud Computing has been studied in previous research, but not extensively. For instance, Liu and Lee [92] proposed a resource allocation algorithm for mobile Cloud Computing systems while providing a security guarantee. They used a semi Markov decision process to model the allocation problem and calculated the optimal allocation policy with the use of linear programming. In their security-aware allocation model, security implementation is supplied by an extra number of VMs according to customers' security requirements. Liang et al. [93] proposed a Security Service Admission Model (SSAM) also based on a Semi-Markov Decision Process to model the system reward for the CSP, which is the difference between the service incomes and the running expenses, while allocating the security requests of the customers in a mobile Cloud Computing setting. They divided the provided security services into two categories : normal security services which provide basic security mechanisms, and critical security services that provide more advanced security features.

Cloud profit maximization is often an essential factor when it comes to resource allocation. For instance, Goudarzi and Pedram [94] proposed a distributed solution to a SLA-based resource allocation problem, which maximizes the total profit in the system while considering the following three dimensions in the optimization : processing, data storage, and communication bandwidth. Also, Nezarat and Dastghaibyfard [95] proposed a game theoretical model to maximize profit in a Cloud environment. In their model, a combinatorial auction mechanism is used to select the winners among the competent users. Other factors such as energy consumption have also been a subject of attention when performing resource allocation in the Cloud. For example, Beloglazov et al. [152] proposed several energy-aware allocation algorithms which ensure efficient energy management within the Cloud's data centers while respecting the constraints on QoS. Other research applied the genetic algorithm to solve the energy-aware Cloud resource allocation problem. Quang-Hung et al. [153] proposed a genetic algorithm for a power-aware allocation of VMs in the Cloud. Portaluri et al. [154] also proposed a power-efficient tasks allocation model in the Cloud based on the genetic algorithm. Finally, Sharma and Guddeti [155] developed an improved genetic algorithm to solve the

multi-objective resource allocation problem in a green Cloud by creating enhanced initial solutions. However, none of the resource allocation frameworks has considered the security risk factor, which is our goal in this work.

8.3 Security Risk Evaluation

The resource allocation model presented in this paper is essentially based on the evaluation of security risk that accompanies the deployment of customers' services on the Cloud infrastructure. In this section, we propose an approach to identify the different security risk factors in a Cloud environment, which will lead us to define the security parameters that can help securing the Cloud infrastructure, and then perform a relative evaluation of the security risk presented by customers' requests. To identify the risk factors that may potentially affect the security of the Cloud service, the latter has to be visualized from different perspectives. To this end, the HHM framework [46] was adopted, which is a comprehensive theoretical framework that was originally conceived to model large-scale complex systems involving many components. The HHM Cloud security risk assessment model is presented in Figure 8.1. It helps us document and understand the different elements, objectives, and constraints in a Cloud Computing scenario. The security risk factors can be seen along the following four dimensions : the Cloud architecture, the security attributes, the risk temporal dimension, and the risk source. In this work, we aim at analyzing the security risk factors in a public Infrastructure-as-a-Service (IaaS) Cloud scenario, which usually poses the greatest security risk on customers' services and data. Hence, the Cloud deployment and service delivery models were not considered as explicit dimensions in the HHM model.
According to [6], the five security attributes that need to be provided in a Cloud environment are :

— Confidentiality, which concerns with protecting customers' sensitive information from unauthorized disclosure. Unlike traditional in house applications, Cloud-based applications require the implementation of more sophisticated security solutions to protect users' confidentiality since they are running on untrusted servers and networks. To protect data and service confidentiality, CSPs usually deploy Identity and Access Management (IAM) solutions and appropriate virtualization and network security, in addition to physically securing their sites and facilities.

— Integrity, which can be defined as the accuracy and validity of data and computations in regards with customers and business process' expectations. CSPs usually protect the data at rest, in transit, and during process using robust cryptography and leakage prevention techniques.

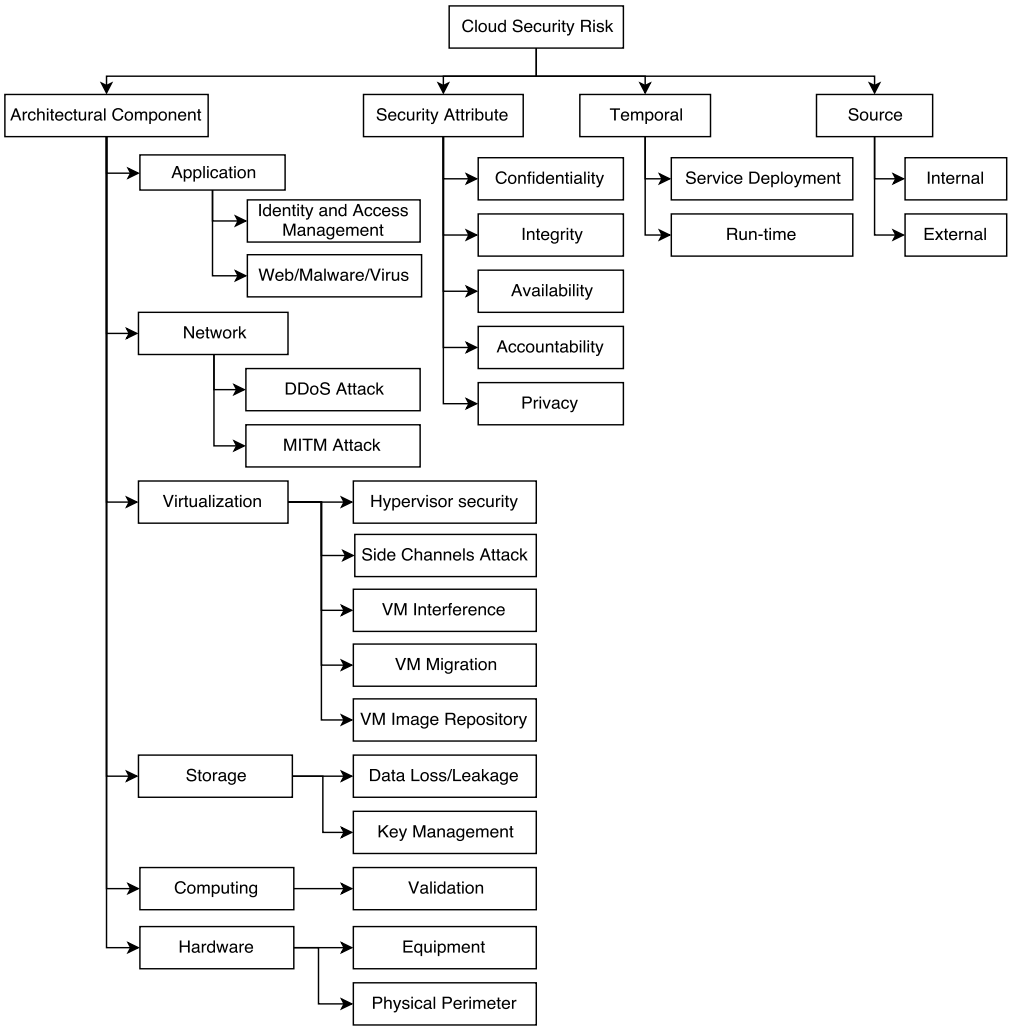


Figure 8.1 The proposed HHM model for Cloud security risk identification.

— Availability of the service and data while hosted on the CSP's side, which is usually maintained by the deployment of effective backup and recovery procedures, incident response plans, and Distributed Denial of Service (DDoS) mitigation solutions.
— Accountability, which stands for the ability of customers to detect dishonest computations and faults within the Cloud (e.g., MapReduce processing faults) and to ensure that their workload is being correctly processed. This is achieved by providing the customers with the ability to perform security audits and assessments in order to detect violations of the SLA. Accountability can also be seen as a measure that helps in keeping track of the integrity attribute.
— Privacy, which is interrelated with the confidentiality and integrity aspects. Privacy preservation techniques include Homomorphic encryption and trusted computing.

Table 8.1 shows which components of the Cloud architecture hold potential threats to each of

these security attributes, and Table 8.2 presents some of these threats with their associated damages to the Cloud service and data. For instance, on the network layer, the botnets that form with the aim of exploiting the vulnerabilities of the Cloud infrastructure, ARP spoofing, and DDoS, are the most feared threats by the CSPs. The virtualization layer also presents serious threats to the Cloud infrastructure. VMs could be compromised during creation, execution, replication, and migration. These threats are described in more detail in [129] and [156]. A security incident can also occur during service deployment like in the case of a compromised job migration, or in run-time. The security risk source is also a critical factor, since in the Cloud, threats to customers' data or services could be caused not only by an outside attack, but also by an inside intruder such as a malicious system administrator.

In order to limit the existence of these threats and increase the protection of users' data and

Table 8.1 Security attributes in the Cloud Computing architecture.

Security attribute	Architecture component: Application	Virtualization	Network	Computing	Storage	Hardware
Confidentiality	●	●	●	●	●	
Integrity				●	●	●
Availability	●	●	●		●	●
Accountability	●	●		●	●	
Privacy	●	●	●	●	●	

Table 8.2 Some of the threats to Cloud Computing.

Threat	Vulnerability	Risk source	Damaged assets
Cross-VM attack via Side Channels	VM co-residence	Neighboring VM	Data
Malicious SysAdmin	Loss of physical control over data	Malicious insider	Data
Data loss/manipulation	Loss of physical control over data	Cloud server, Server administrator	Data
Dishonest computation in remote servers	Outsourced computation	Cloud server	Computations
Direct and indirect DoS attacks	VM co-residence, Bandwidth under-provisioning	Malicious user, Neighboring VM	Service, Data
Economic Denial of Sustainability (EDoS)	Cloud pricing model	Malicious user	Service

Table 8.3 A set of security parameters that could help forming the security configurations.

Security service	Security parameters

Identity and Access Management	- Implementation of Multi-factor authentication protocols
	- Deployment of Single Sign On (SSO) authentication
	- Enforcement of policies on password strength and expiration
	- Blocking of invalid login attempts
	- Enabling of client certificate for SSL/TLS
	- Authorization and access control policies
	  - Implementation of risk-based entitlement decisions
	  - Enabling the use temporary access credentials
	  - Frequency of review of system users and administrators' entitlements
	  - Frequency of review of access control logs and accounts' activity
	- Implementation of XACML access control schemes

Network Protection	- Implementation of IDS/IPS - Deployment of DDoS mitigation solutions - Deployment of IPSec VPN networks between private clouds - Deployment of firewalls and traffic isolation techniques - Configuration of security groups and Access Control Lists - Configuration of web application scanners - Frequency of network penetration tests

Data Security and Privacy	- Deployment of secure storage schemes
	  - Encryption key length
	  - Enabling the internal storage of encryption keys
	  - Capability of open encryption methodologies
	  - Capability of creation of a unique encryption key per tenant
	  - Enabling of HTTP Strict Transport Security (HSTS)
	  - Database deployment with SSL protected transactions
	  - Support of data isolation
	- Implementation of data dispersion techniques
	- Support of secure data deletion
	- Implementation of data loss/leakage prevention techniques
	- Deployment of data backup and restore procedures
	  - Data backup frequency
	  - Backup restoration mean-time
	  - Number of redundant backup sites
	- Provisioning of dynamic malware analysis and detection

VM Security	- Deployment of VMs' interference prevention techniques - Enabling SSH secure communications between VMs - Implementation of encrypted VM live migration - Implementation of data destruction procedures after migration - Capability of VM backup, restoration, and clean-up - Provisioning of Hypervisor-level role-based access control - Capability of events monitoring and auditing by the hypervisor - Frequency of assessment of virtualization vulnerabilities
Integrity Verification	- Deployment of PDP techniques for data validation - Deployment of computing integrity checking techniques - Deployment of applications with computing replication
Service Availability	- Deployment of data replication between the Cloud nodes - Capability of Multi-failure disaster recovery - Capability of infrastructure service fail-over to other CSPs - Configuration of SIEM incident reporting, analysis and alerting - Frequency and coverage of risk analysis and assessment plans - Frequency of reviewing and testing of business continuity plans - Data redundancy level - Frequency of vulnerability scans
Physical Security	- Implementation of Role-based access control systems - Monitoring and controlling the business process' physical service points - Isolation and monitoring of data storage physical points - Monitoring of environmental conditions that affect computer systems

services, several security services are usually offered by the CSP. In Table 8.3, we show some of these services along with a set of security parameters that enable the deployment of these services. These parameters could form the basis of developing different security configurations that might be offered to the Cloud customers, who will select the configurations that best match the security requirements of their services and applications. The security profile of customers' applications is usually determined by the nature of the deployed services (e.g., web service, data storage, computational workload, etc), and the nature of their users (e.g., individuals, banks, governments, etc). Offering these heterogeneous security configurations will help in relatively evaluating the security risk that a customer's request would generate within the Cloud infrastructure. Security risk is usually defined as the likelihood of occurrence of a security incident that could have an impact on the service or asset. In a Cloud Computing environment, it could be expressed as a function of threats' occurrence probability, the degree of existence of vulnerabilities, and the impact of the threat on the deployed service :

$$Risk = F(Threats,\ Vulnerabilities,\ and\ Impact) \tag{8.1}$$

It is then safely reasonable to state that increasing the security level by providing more security capabilities will lower the security risk that a customer would introduce to the infrastructure when deploying her service, since the probability of threats' occurrence will decrease and the impact of an incident will be reduced. Let $\mathcal{SC} = \{SC_l \mid 0 \leq l \leq L\}$ be a set of L security configurations that the CSP offers to her customers, where SC_0 is the default security configuration that will imply the highest security risk level, and SC_L is the most

Table 8.4 An example of five different security configurations.

Security parameter	SC_0	SC_1	SC_2	SC_3	SC_4
Frequency of network penetrations tests	-	-	✔	✔	✔
Frequency of review of access logs	-	monthly	weekly	daily	daily
VM clean-up capabilities	-	-	✔	✔	✔
DDoS mitigation solutions	-	-	-	✔	✔
Encryption key length (bits)	128	128	128	128	256
Support of secure data deletion	-	-	-	✔	✔
Data backup frequency	monthly	weekly	weekly	daily	hourly
Encrypted VM live migration	-	-	-	-	✔

advanced security configuration that will generate the lowest security risk level. In other words, SC_0 is considered to be the configuration that provides basic security capabilities. Table 8.4 shows an example of how the proposed security parameters could be used to define five different security configurations.

In order to define the security risk level associated with a customer request according to the selected security configuration, we introduce a discretization function that aims at relatively evaluating the security risk levels with respect to the provided security configurations as follows :

$$SRL_l = (L - l)Risk/L \quad \forall SC_l \in \mathcal{SC} \tag{8.2}$$

where $Risk$ denotes the maximum value of security risk that could exist on the CSP's infrastructure and is assigned to the default security configuration SC_0 (i.e., $SRL_0 = Risk$). This value could be evaluated by the CSP by applying suitable risk assessment methodologies. Figure 8.2 shows an example of the assignment of security risk levels when five security configurations are offered. Providing advanced security configurations will introduce additional costs to the CSP's budget, hence the idea of offering these configurations to be sold to customers, and eventually allocating their requests according to their intention in securing their services. The evaluation of security risk generated by customers' requests is significant to the problem of resource allocation in the Cloud, since it will allow the CSP to manage her resources in a security risk-aware fashion and have more control over the security of her infrastructure.

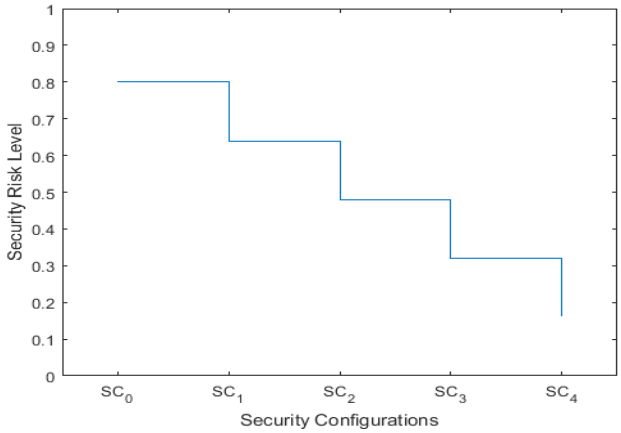


Figure 8.2 Definition of relative security risk levels in terms of security configurations.

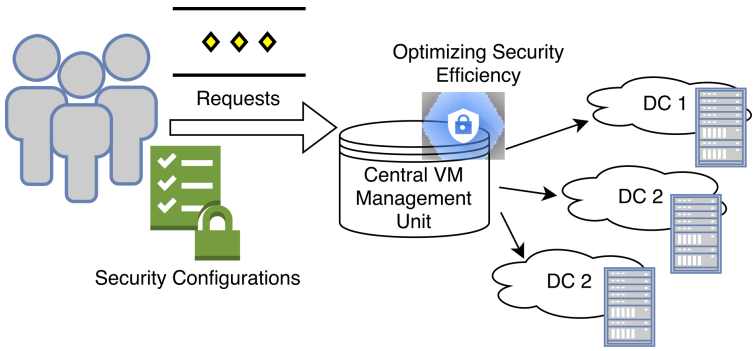


Figure 8.3 The security risk-aware resource allocation architecture.

8.4 Security Risk-aware Resource Allocation

In this section, the problem of Cloud resource allocation based on security risk evaluation is defined and modeled in an InterCloud setting. The Cloud Computing paradigm presents several concerns and challenges for CSPs, such as Quality of Service (QoS) guarantee, resource limitation, disaster-recovery planning, regional distribution of workload, and legal issues. To address these concerns, a CSP usually provides her services through the deployment of multiple data centers that are geographically distributed but securely interconnected. This concept is commonly known as InterCloud. The problem in this paper consists in allocating customers' requests to the Cloud's data centers in a way that optimizes the trade-off between security risk and resource utilization, i.e., the CSP needs to provision the resources required by the customers on her data centers while ensuring the safety of her infrastructure. The architecture of the allocation model is shown in Figure 8.3. The model could be implemented onto the VM central management unit that controls the VMs' placement on the Cloud's data centers and can be fully transparent to customers.

We consider a set of Cloud customers $\mathcal{I} = \{i \mid 1 \leq i \leq I\}$, each generating one resource provisioning request, and a set of data centers $\mathcal{DC} = \{j \mid 1 \leq j \leq J\}$ that belong to one CSP. Note that the proposed model could work in the case where the data centers belong to multiple CSPs, and the allocation will be performed by a third party (e.g., Cloud broker). Also, in a usual multi-tiers Cloud architecture, the service providing layer could be separate from the infrastructure providing layer. In this case, the service provider will only forward the received requests to the infrastructure provider which will perform the allocation of requests on her data centers. The Cloud resources are usually allocated in the form of VM instances. We consider K types of VMs that are offered by the CSP, each type offers a different combination of resources (i.e., number of CPU cores, memory amount, and storage capacity). The set of VM types is denoted by $\mathcal{VM} = \{k \mid 1 \leq k \leq K\}$ and the set of resource

types is denoted by $\mathcal{R} = \{r \mid 1 \leq r \leq R\}$. At the time of the allocation, each data center in $\mathcal{DC}$ has an available capacity of each resource type. We denote by $\mathcal{C} = \{C_j \mid 1 \leq j \leq J\}$ the set of vectors of available resource capacities on all data centers, where $C_j = (C_{j,1}, \ldots, C_{j,R})$. We denote a customer's request by $\theta_i = (S_i, \mathcal{F}_i)$, where $S_i = (S_{i,1}, \ldots, S_{i,K})$ is the vector containing the number of VMs that are required of each type $k \in \mathcal{VM}$, and $\mathcal{F}_i \in \mathcal{SC}$ is the security configuration that customer i is requesting to deploy along with the Cloud service. The total usage $u_{i,r}$ of a resource $r \in \mathcal{R}$ in the request of customer $i \in \mathcal{I}$ is calculated as follows :

$$u_{i,r} = \sum_{k \in \mathcal{VM}} S_{i,k}\alpha_{k,r} \quad \forall i \in \mathcal{I},\ \forall r \in \mathcal{R} \tag{8.3}$$

where $\alpha_{k,r}$ is the usage of r by the VM type $k \in \mathcal{VM}$.
We define a binary variable $x_{i,j} = \{0, 1\}$ that indicates if request θ_i will be allocated to data center $j \in \mathcal{DC}$. Let Q_i denotes the security risk level associated with the security configuration F_i that customer i is requesting. Eventually, all VMs provisioned to respond to customer i's request will generate the security risk level Q_i on the infrastructure. We define the Global Security Risk Level (GSRL) metric as a way to globally monitor the security risk level on each data center. It is computed as the average of security risk levels of all allocated VMs according to the allocation vector x as follows :

$$GSRL_j = \frac{\sum\limits_{i \in \mathcal{I}} Q_i \sum\limits_{k \in \mathcal{VM}} S_{i,k}x_{i,j}}{\sum\limits_{i \in \mathcal{I}} \sum\limits_{k \in \mathcal{VM}} S_{i,k}x_{i,j}} \quad \forall j \in \mathcal{DC} \tag{8.4}$$

The idea behind this estimation lies in the fact that a data center with higher number of high security risk users will pose higher attack probability, and the average security risk is the ideal metric to reflect this probability. During the allocation process, the global security risk level $GSRL_j$ on each data center $j \in \mathcal{DC}$ should not exceed a critical value β_j that is usually set by the CSP according to her understanding and evaluation of the infrastructure. This value could be defined based on the history of incidents and dynamically updated according to the monitoring activities, and of course, should be in correlation with the security risk levels that the CSP defines for her offered security configurations.
A usual objective of the Cloud resource allocation problem is to achieve optimal resource utilization. In our model, we aim at maximizing the amount of allocated VM instances while keeping the security risk level on the infrastructure below its critical value. In other words, we try to allocate as much requests as possible without harming the maximum security risk levels that the CSP's data centers could support. The goal here is to optimize the allocation

of the Cloud resources in a security context, that is, ensuring a high number of securely allocated VMs. This will reduce the rate of unprotected Cloud users and encourage the customers to purchase advanced security configurations with their deployed services in order to guarantee the allocation of their requests. The problem can be considered as a version of the VM Placement Problem (VMPP), which is usually described as a variant of the multi-dimensional Multiple-Knapsack Problem (MKP) [106] from the class of Bin Packing problems (BPP). We formulate the security risk-aware resource allocation problem as follows :

$$Max \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{DC}} \sum_{k \in \mathcal{VM}} S_{i,k} x_{i,j} \tag{8.5}$$

Subject to :

$$\sum_{i \in \mathcal{I}} u_{i,r} x_{i,j} \leq C_{j,r} \quad \forall j \in \mathcal{DC},\ \forall r \in \mathcal{R} \tag{8.6}$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{VM}} S_{i,k}(Q_i - \beta_j) x_{i,j} \leq 0 \quad \forall j \in \mathcal{DC} \tag{8.7}$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{VM}} S_{i,k} x_{i,j} \leq \tau \quad \forall j \in \mathcal{DC} \tag{8.8}$$

$$\sum_{j \in \mathcal{DC}} x_{i,j} \leq 1 \quad \forall i \in \mathcal{I} \tag{8.9}$$

$$x_{i,j} = \{0, 1\} \quad \forall i \in \mathcal{I},\ \forall j \in \mathcal{DC} \tag{8.10}$$

where Constraint 8.6 ensures that the allocated resources of each type on each data center will not exceed the available capacity of that type, and Constraint 8.7, which is equivalent to $GSRL_j \leq \beta_j,\ \forall j \in \mathcal{DC}$, will ensure that the global security risk level $GSRL_j$ on each data center j will always be lower than the critical value β_j. Constraint 8.8 ensures workload balancing between the CSP's data centers by implementing a workload spreading policy on the VM level. The constraint guarantees that the number of allocated VMs on each data center will not exceed a predefined amount τ set by the CSP (i.e., according to the received number of requests), which will permit to reduce the probability of overwhelming a specific data center and eventually, energy costs. Finally, Constraint 8.9 ensures that if a request θ_i is allocated, it will be allocated to only one data center in $\mathcal{DC}$. In this problem, the requests could be seen as the items and the data centers as the knapsacks, with the added dimension of security risk along with the traditional dimensions of resource capacities. The integration

of the security risk aspect in the optimization of resource allocation is the main contribution of this paper. The multi-dimensional MKP has proven to be NP-complete [106], and no solution in polynomial time exists for this problem. Formal methods that are used to solve Integer Linear Programming (ILP) problems can only scale up to small problem instances. In our case, the problem should be able to respond to a large number of requests in an online fashion. Therefore, we propose to use two different metaheuristics approaches to find an approximated solution to the problem.

8.5 Evolutionary Computation for Security Risk-aware Resource Allocation

In this section, we propose to solve the security risk-aware Cloud resource allocation problem using two different evolutionary computation approaches : the evolutionary GA, and the swarm-based ABC optimization. The goal is to allocate the resources in a way that guarantees acceptable security risk levels within the Cloud while optimizing resource utilization. The solution should also allocate customers' VMs on the Cloud's data centers in a way that guarantees the workload balancing property. In this work, we focus more on the security risk level while performing the allocation of resources, but other aspects (e.g., geographical, legal, response time, etc) could also be considered and the model will adapt accordingly by simply implementing additional constraints. The two proposed algorithms will be executed anytime a set of customer requests is placed on the Cloud controller server.

8.5.1 Applying Evolutionary Algorithms

First, the GA from the family of Evolutionary Algorithms (EA) [15] is adopted to solve the problem. The GA is a population-based meta-heuristic search algorithm that mimics the natural behavior of evolution, and is commonly used in automatic programming and machine and robot learning. It is suitable for the approximation of optimal solutions to complex optimization problems and could be considered computationally efficient. The GA usually starts by randomly generating an initial population of candidate solutions called chromosomes, and continuously applies the three genetic operators : selection, crossover, and mutation in

request	1	2	3	4	...	θ_{I-1}	θ_I
Data center	3	2	4	3	...	J	2

Figure 8.4 An example of the representation of the GA chromosome in our model.

order to improve the quality of the generated chromosomes with respect to the optimization problem's objective, until finally selecting the best solution to the problem. The process termination constraints usually involve the maximum number of iterations and early convergence, where the fittest chromosome can not be further improved.

In our model, a chromosome C is defined as an I-dimensional vector representing the assignment of customers' requests to the CSP's data centers. As shown in Figure 8.4, the value at a position i in the vector indicates the data center to which request θ_i will be allocated. For instance, in the shown example, request θ_1 will be allocated to data center 3 and request θ_2 will be allocated to data center 2. The main operation in the GA is the evaluation of chromosomes, which is based on computing their fitness values. Based on the objective function of our problem presented in Equation 8.5, we define the fitness function $f : \ C \rightarrow \mathcal{R}^+$ to evaluate the quality of each chromosome $C = (C_1, \ldots, C_I)$, as follows :

$$f(C) = \sum_{i \in \mathcal{I}, C_i \neq 0} \sum_{k \in \mathcal{VM}} S_{i,k} \tag{8.11}$$

The objective of the problem consists in maximizing the value of the fitness function, which represents the amount of VMs that chromosome C has allocated to customers.

The pseudo-code of applying the GA to the security risk-aware Cloud resource allocation problem is presented in Algorithm 3. The algorithm takes as inputs the following parameters : the set of customers' requests $\theta = \{\theta_i \mid i \in \mathcal{I}\}$; the set $\mathcal{C}$ of available resource capacities on all data centers ; the vector $GSRL = (GSRL_1, \ldots, GSRL_J)$ containing the values of global security risk levels on all data centers computed before the time of the new allocation ; the critical security risk values for all data centers ; and the workload spreading policy parameter τ. The algorithm needs to have knowledge about previous values of security risk levels and amounts of allocated VMs on the data centers to consider them while performing a new allocation, since it will be functioning in an online mode. The algorithm outputs the best allocation solution found by the GA denoted by C_{best}, which indicates which requests were allocated and to which data center. Eventually, the vector C_{best} will be used by the CSP to provision the resources.

First, the algorithm computes the usage of all resources $r \in \mathcal{R}$ by request $\theta_i \in \theta$ using Equation 8.3 and makes use of this information along with the security risk level Q_i associated with the request to evaluate the satisfaction of Constraints 8.6, 8.7, and 8.8 during the allocation process. Then, the process of GA starts by randomly generating a chromosome population of size P and evaluating the fitness value of each chromosome using Equation 8.11 (lines 7-9). The GA then proceeds to create a new child generation of chromosomes, also of

size P, by following a selection process and applying crossover and mutation operations on the chromosomes of the parent generation (lines 11-17). The roulette wheel selection method is usually adopted to select two chromosomes of the current population. Then, the two-point crossover method was used to enable the selected children chromosomes to inherit the good parts of their parents by exchanging their corresponding parts at a randomly generated position. For the mutation part that follows the crossover operation, we use the bit-flop mutation method where each child chromosome undergoes a modification at a particular randomly generated position in the vector. The crossover and mutation operations happen according to the crossover and mutation rates respectively, which determine their frequency of occurrence during the process.

After a new generation of chromosomes is formed, the fitness of each one is evaluated (lines 18-24) and the chromosome with the best fitness value is selected as the current solution ($C_{current}$). If this chromosome is fitter than the one selected from the previous generation, then it becomes the best solution to the problem (C_{best}), which will be returned by the algorithm once one of the termination constraints is satisfied. Since our optimization problem is constrained, we adopt the constrained handling method proposed by [157], which is based on applying a tournament selection operator instead of the greedy selection mechanism. With this approach, when two solutions are compared, the following rules are enforced : a feasible solution is always preferred over an infeasible one, the fittest solution is preferred in the case of two feasible solutions, and the solution with a smaller constraint violation is preferred in the case of two infeasible solutions.

The time complexity of Algorithm 3 is determined by the optimization part. For instance, the time complexity of one iteration of the GA is determined by the complexity of the three operations : selection, crossover, and mutation. The time complexity of the roulette wheel selection method is $O(P)$, where P is the population size, and that of the crossover and mutation operations is $O(I * P)$, where I is the number of requests. Thus, the time complexity of one iteration is $O(I * P)$. The total time complexity is $O(I * P * Q)$, where Q is the number of maximum iterations in the GA.

8.5.2 Applying Swarm Intelligence

Next, we choose to adopt the Artificial Bee Colony (ABC) algorithm from the family of swarm intelligence algorithms to find an approximated solution to the security risk-aware Cloud resource allocation problem. The algorithm was proposed by Karaboga and Basturk in [16] to solve complex computational problems, and is inspired by the natural intelligent behavior of honey bees. The main components of this model are :

— Food sources, which are evaluated by the forager bees according to several factors such

Algorithm 3 Pseudo-code of applying GA to the security risk-aware resource allocation problem

Input:
- The set of received requests θ
- The set of vectors of available resource capacities $\mathcal{C}$ on each data center
- The vector $GSRL$ computed before the execution of the algorithm
- The critical security risk values $\beta_j,\ \forall j \in \mathcal{DC}$
- τ, the workload spreading policy parameter

Output:
- The best allocation solution C_{best}

for all $\theta_i \in \theta$ **do**
  for all $r \in \mathcal{R}$ **do**
    Compute $u_{i,r}$
  end for
end for
$C_{best} \leftarrow$ *empty vector*
- Generate the initial population of chromosomes of size P
- Compute the fitness of each chromosome C

while termination constraints are not satisfied **do**

  while the size of the new generation $\neq P$ **do**
    - Select two chromosomes using roulette wheel
    - Apply two-point crossover on the chromosomes
    - Apply bit-flop mutation on child chromosomes
  end while
  - Evaluate the fitness of the new generation
  - Apply the tournament selection operator to select the fittest chromosome as the current best solution $C_{current}$
  if $f(C_{current}) > f(C_{best})$ **then**
    $C_{best} \leftarrow C_{current}$
  end if
end while
return C_{best}

as the closeness to the hive, the taste of the nectar, and the abundance of energy.

— Employed bees, which are forager bees assigned to exploit specific food sources. They share their information about these sources with the rest of the hive through the waggle dance.

— Unemployed bees, which could be of two types : scout bees that randomly search for

food sources, or onlooker bees that use the information given by the employed bees to try to find potential food sources.

In the ABC algorithm, the positions of the food sources constitute the candidate solutions to the optimization problem and their qualities represent the fitness values of these solutions. The algorithm initially generates a population of random food source positions (solutions) of size S. The solution's structure is similar to that of the chromosomes generated by the GA in the previous section. It is an I-dimensional vector that maps customers' requests to the data centers. We use the same fitness function $f(.)$ that was used to evaluate the GA chromosomes to evaluate a food source $F_k = (F_{k,1}, \ldots, F_{k,I})$, as follows :

$$f(F_k) = \sum_{i \in \mathcal{I}, F_{k,i} \neq 0} \sum_{k \in \mathcal{VM}} S_{i,k} \quad 1 \leq k \leq S \tag{8.12}$$

The pseudo-code of applying the ABC algorithm to the security risk-aware Cloud resource allocation problem is presented in Algorithm 4. The inputs of the algorithm are the same of Algorithm 3, and the output is the best selected food source denoted by F_{best}. The algorithm first computes the usage of resources by each request. After the generation and evaluation of an initial population of food sources (lines 6-7), the process of the ABC algorithm will involve three stages :

— First, the artificial employed bees introduce modifications to the positions of the sources using the following equation :

$$F'_{k,i} = F_{k,i} + \phi_{k,i}(F_{k,i} - x_{k',i}) \quad 1 \leq k \leq S \tag{8.13}$$

where F'_k is the newly produced solution, F_k is the old one, k' and i are randomly chosen indexes between 1 and S and 1 and I respectively, and $\phi_{k,i}$ is a random number in the interval $[-1, 1]$ that controls the generation of neighbor food sources by the bee after visually comparing two different food positions. After evaluating the fitness value of the produced solution, the employed bee applies a local selection process to choose between the new solution and the one it already has in its memory (lines 10-15). The tournament selection operator is also applied in this algorithm to implement the feasibility of Constraints 8.6, 8.7, and 8.8.

— Second, the employed bees share their information about the produced food sources with the artificial onlooker bees, which, after watching the dances, perform a global probabilistic selection process by evaluating the selection probability of each food source F_k according to its relative fitness value as follows :

$$P(F_k) = \frac{f(F_k)}{\sum_{z=1}^{S} f(F_z)} \quad 1 \leq k \leq S \tag{8.14}$$

After selecting a food source, the artificial onlooker bee becomes employed and thus, performs the same process as in stage one (lines 16-23).

— Third, a random selection process that is performed by artificial scout bees, which randomly generate solutions to replace the abandoned food sources. In our model, a random vector solution F_k of size I is produced such that the values of $F_{k,i},\ 1 \leq i \leq I$, are in the interval $[0, J]$. These solutions are produced when older solutions can not be further improved. This is controlled by the ABC algorithm's *limit* parameter that controls solutions' abandonment. The scout bees also apply the tournament selection operator to choose the fittest between the produced solutions and the old ones, and correspondingly update their memories (lines 24-28).

This process is repeated until the Maximum Cycle Number (MCN) is attained, which is the basic control parameter used in the ABC algorithm.

The time complexity of Algorithm 4 is determined according to that of the ABC algorithm. In a single iteration, the time complexity of what is performed by the employed and onlooker bees is $O(S * limit)$, where S is the population size and *limit* is the abandonment controller parameter. On the other hand, the scout bees perform a searching process of time complexity $O(S)$. Therefore, the total time complexity of the algorithm is $O(S * limit * MCN)$, where MCN represents the maximum number of iterations in the ABC algorithm.

8.6 Experimentation and Results

In this section, a set of evaluation experiments is conducted and results are analyzed. We implement the proposed algorithms in MATLAB, and use a Mixed-Integer Linear Programming solver to find the exact optimal solution to the security risk-aware resource allocation problem. The solver first tries to reduce the problem size, then uses heuristics to solve an initial relaxed problem and produce an initial feasible solution, and finally applies a Branch and Bound algorithm [158] to perform an exhaustive search for the optimal solution.

8.6.1 Experimental Setup

The experiments aim at evaluating the performance of the proposed algorithms in terms of the quality of their produced solutions and their computational efficiency, and also demonstrating how the proposed model helps in better controlling the security risk level within the Cloud.

Algorithm 4 Pseudo-code of applying ABC optimization to the security risk-aware resource allocation problem

Input:
- The set of received requests θ
- The set of vectors of available resource capacities $\mathcal{C}$ on each data center
- The vector $GSRL$ computed before the execution of the algorithm
- The critical security risk values $\beta_j,\ \forall j \in \mathcal{DC}$
- τ, the workload spreading policy parameter

Output:
- The best allocation solution F_{best}

for all $\theta_i \in \theta$ **do**
  for all $r \in \mathcal{R}$ **do**
    Compute $u_{i,r}$
  end for
end for
- Generate the initial population of S food sources
- Compute the fitness value of each food source F
$cycles \leftarrow 1$
while $cycles \leq MCN$ **do**
  Employed bees :
  - Produce new food sources and evaluate their fitness
  - Apply the tournament selection operator to choose the best food source according to feasibility
  Onlooker bees :
  - Compute the selection probabilities associated to the food sources advertised by employer bees
  - Apply the tournament selection operator to choose the best food source according to feasibility
  - Onlooker bees become employed. Repeat related steps
  Scout bees :
  - Replace abandoned food source positions with new randomly generated ones
  - Memorize in F_{best} the best food source found so far
  $cycles \leftarrow cycles + 1$
end while
return F_{best}

We simulate the customers as objects, each requesting a number of VMs of different types and a specific security configuration. We consider five different security configurations in our experiment and assign the corresponding security risk levels according to Equation 8.2 with

Table 8.5 VM types offered by Amazon EC2.

	Small (k=1)	Medium (k=2)	Large (k=3)	ExtraLarge (k=4)
CPU cores (r=1)	1	2	4	8
Memory (GB) (r=2)	1.7	3.75	7.5	15
Storage (GB) (r=3)	160	410	850	1690

Table 8.6 GA and ABC optimization parameters.

GA		ABC	
Population size P	100	Population size S	90
Number of generations	1000	Number of iterations	1000
Crossover rate	0.6	Abandonment limit	60
Mutation rate	0.1		
Crossover type	Two point		
Mutation type	Bit-flop		

Table 8.7 Achieved fitness value by GA and ABC for different population sizes.

P, S		I			
		50	100	150	200
60	GA	550.23	1300	2310.12	1942.72
	ABC	620.65	1590.32	2378.25	3280.39
70	GA	632.44	1390.14	2356.75	2098.46
	ABC	699.78	1587.12	2458.89	3219.29
80	GA	750.13	1450.69	2380.33	2044.26
	ABC	720.19	**1613.67**	2446.57	3350.48
90	GA	**870.51**	1499.71	2356.79	2150.65
	ABC	**780.15**	1602.37	**2489.44**	**3402.64**
100	GA	800.89	**1511.22**	**2402.75**	**2197.33**
	ABC	772.97	1597.35	2470.35	3375.46

a maximum value of security risk $Risk = 0.8$ (on a scale between 0 and 1). We consider the three standard types of Cloud resources in our experimentation, the number of CPU cores, the amount of memory (GB), and storage capacity (GB), and four different types of offered VMs like the ones offered by Amazon EC2. The four types are presented in Table 8.5 along with their parameters $\alpha_{k,r}$ which correspond to the usage of resource type $r \in \mathcal{R}$ by the VM type $k \in \mathcal{VM}$. For each request, the types of VMs are randomly selected and a random number between 0 and 10 is generated to simulate the number of requested VMs of each type. This interval is fixed for all evaluation scenarios. Our focus in this paper is oriented toward the study of security risk during resource allocation regardless of whether customers' requests are small or large. In our experimentation, we mainly highlight the dependence of the size of our problem on the number of received requests and the number of Cloud data centers involved in the assignment. In all the experiments we set the value of τ (workload balancing threshold) to $40I/J$, where I and J are respectively the number of requests and the number of Cloud data centers. Finally, on each data center $j \in \mathcal{DC}$, we set the vector of resources $C_j = (200CPU, 1000GB, 200TB)$.

Table 8.6 shows the optimization parameters that we set for the GA and ABC algorithm. These parameters are closely related to the performance of the two algorithms, hence we had to determine some of them experimentally. In [159], Jong et al. provided the most used values for GA parameters. However, since our problem works with a large search space, a small population size might not provide a good approximation of the solution. Therefore, we performed extensive experimentation to determine the ideal value of population size for both GA and ABC. We executed both algorithms for different combinations of the number of data centers J and the number of received requests I while varying the two parameters P and S and evaluated the achieved fitness values. Our decision was taken in terms of the fitness value since the quality of the solution is more interesting to our problem than the allocation time, as long as the execution is still computationally affordable. Table 8.7 shows one of the evaluation cases that led us to fixing the population size to 100 for GA and 90 to ABC. Similarly, other experiments were conducted to determine the optimal GA mutation rate and the ABC abandonment limit parameter for our problem. A high mutation rate or a small abandonment limit will cause the algorithms to perform a random search. The experimentation determined that a mutation rate of 0.1 and an abandonment limit of 60 were the values required for the algorithms to achieve good quality solutions. Finally, and following extensive experimentation trials, the GA number of generations and the ABC number of iterations were selected from the interval [50,1000] according to the size of the problem. The values were evaluated in terms of several factors such as the number of fitness evaluations without improvement and population convergence.

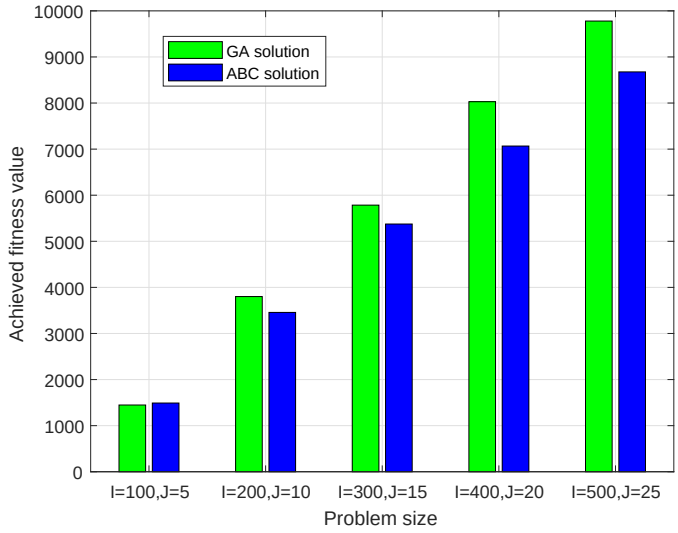


Figure 8.5 Value of the fitness achieved by GA and ABC.

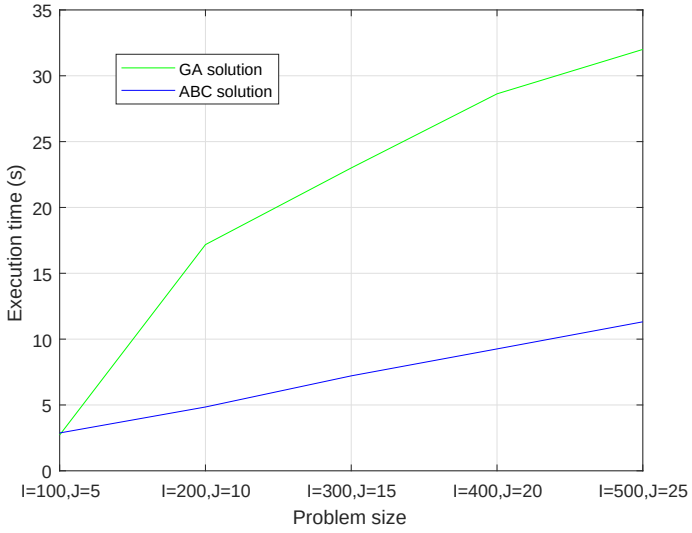


Figure 8.6 Execution time of the algorithms.

8.6.2 Performance of the GA and ABC

In a first scenario, we aim at comparing the performance of the two proposed algorithms. We run both algorithms for different problem sizes. The number of customers' requests I varies between 100 and 500 with an incremental step of 100, and the number of data centers J varies between 5 and 25 with an incremental step of 5. The value of critical security risk β_j is set to 0.5, $\forall j \in \mathcal{DC}$. The results are the average of fifty runs. Figure 8.5 shows the

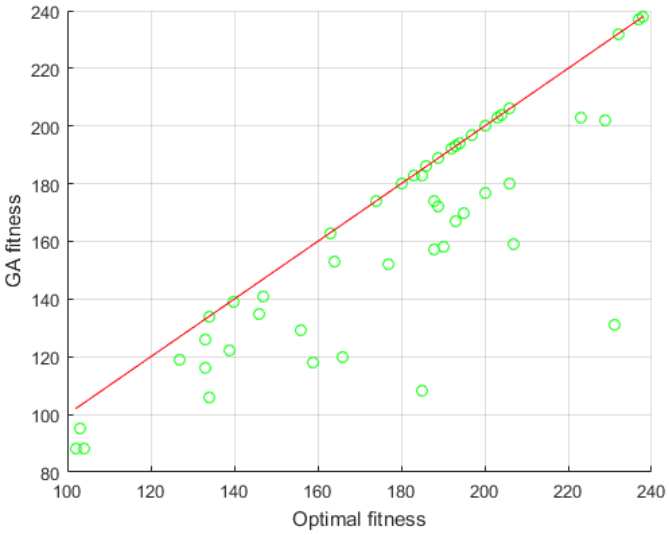


Figure 8.7 Comparison of GA solutions with the optimal ones.

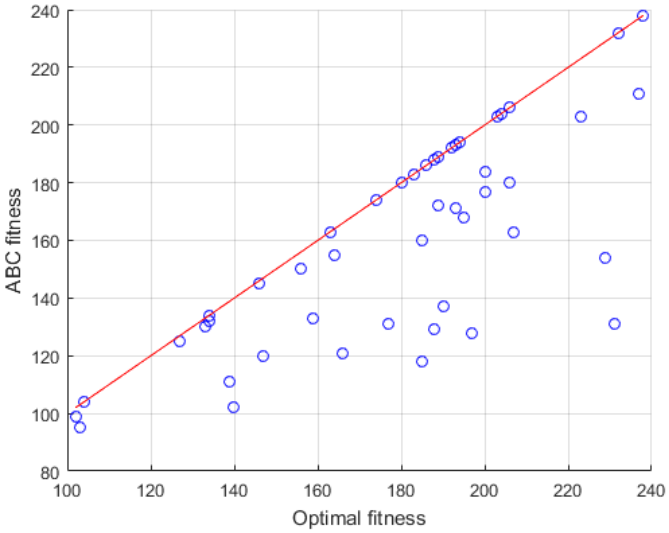


Figure 8.8 Comparison of ABC solutions with the optimal ones.

value of the achieved fitness (the amount of allocated VMs) by the two algorithms. As we see in the figure, both algorithms nearly achieved the same fitness when the problem size was small. However, the GA achieved higher fitness values than the ABC algorithm in all other cases. In Figure 8.6, the execution time of the two algorithms is measured and illustrated for the same scenarios. The ABC algorithm shows lower execution time than the GA (less than half) during the allocation. This gives an advantage to the ABC solution. However, both approaches can be implemented in online mode, since their execution time was far lower

than that of finding the optimal solution, which was taking hours even for small problem instances. Both approaches are able to solve the problem in polynomial time, and will be able to efficiently cope with the Cloud's scalability property, and the continuous fluctuation in the amount of received requests. The choice between both approaches should be evaluated in lights of the acceptable trade-off between solution quality and computational efficiency.
In the second scenario, we evaluated the quality of the solutions generated by both the GA and ABC algorithm with respect to the optimal solution. Another experiment was conducted to determine how close were the heuristic solutions to the optimal one. Considering that MATLAB was taking hours to find the optimal solution to the problem and sometimes causing memory overflow in the case of large-scale problems, we decided to conduct the experiment on a small problem instance, where the number of data centers $J = 10$ and the number of requests $I = 10$. We kept the value of critical security risk the same as in the first case. In this scenario, fifty runs were executed. Figure 8.7 and Figure 8.8 compare the fitness achieved by the two algorithms with that of the optimal solution, and the line $y = x$ in the figures represents the optimal solutions. The closer the points are to this line, the closer is the performance of the two evolutionary computation solutions are to the optimal one. We notice that the GA achieves closer solutions to the optimal one, and in most of the fifty cases, the solutions generated by the GA were closely distributed around the optimal ones. On the other hand, the solutions generated by the ABC algorithm were distributed relatively further away from the line $y = x$. The GA optimality gap in this scenario was in the interval [0.011,0.433] with an average of 0.0845, whereas that of ABC was in the interval [0.007,0.433] with an average of 0.0963. We can conclude that both algorithms are able to generate good quality solutions with slight differences. However, the GA showed to generate better quality solutions than ABC for large size problems.

8.6.3 Model Security Analysis

In the third scenario, we conduct an experiment that shows the role of the critical security risk parameter in the proposed model. We set the number of data centers J to 10 and the number of requests I to 500, and varied the parameter β_j, $\forall j \in \mathcal{DC}$ from 0.1 to 0.5 with an incremental step of 0.1. Although that in our experiments we set the same value of critical security risk on all data centers, this is not always the case in a real InterCloud setting, where each data center could set a different value of critical security risk depending on many factors such as the nature of the deployed services or geographical localization. This is actually an important feature of our model, which will permit to distribute the security risk among the data centers according to their ability to support it, and grant the CSP more flexibility over controlling and managing the allocation of her resources. The goal of this scenario is to show

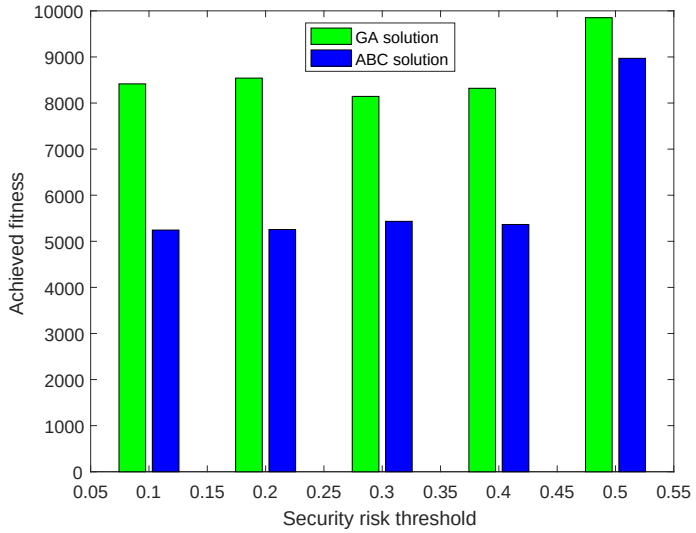


Figure 8.9 Value of the fitness function in terms of the critical security risk value β_j.

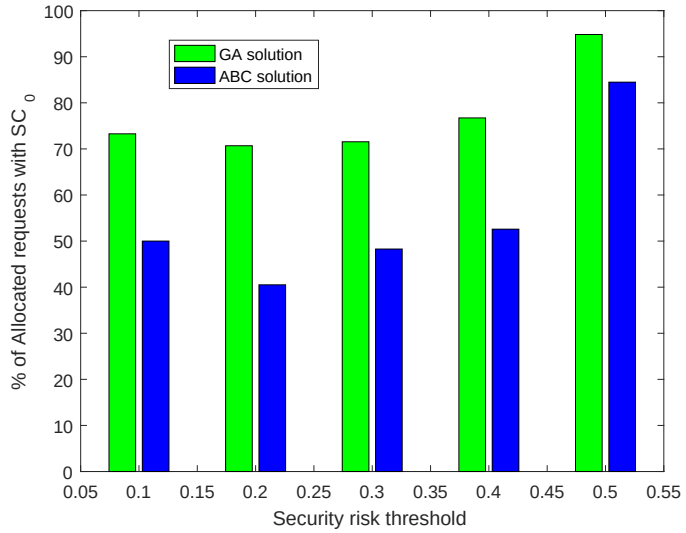


Figure 8.10 Percentage of allocated requests received with the default security configuration SC_0.

how the value of the critical security risk could affect the security of the system and how could the CSP determine the adequate trade-off to deliberate between achieved profit and security risk awareness.

igure 8.9 shows the achieved fitness value by the two proposed metaheuristics approaches with respect to the critical security risk value. It is normal to see the GA algorithm achieving higher fitness values than ABC since we are dealing with a large problem instance in this

case. However, what matters here is the relative change in the fitness value achieved by each of the two algorithms. When the critical security risk value increases, the fitness value also increases. For instance, in the case of ABC, the fitness value when β_j was equal to 0.5 was almost the double of that achieved when $\beta_j = 0.4$. This is due to the fact that according to Constraint 8.7, more requests will be allocated when high critical security risk values are supported by the data centers and resource utilization will boost, and eventually, CSP's generated profit will increase. To determine the adequate trade-off between security risk and profit, the CSP will need to carefully define the critical security risk value that her data centers could handle. On the other hand, encouraging customers to buy the offered security configurations will surely help in slowly decreasing the global security risk level on the data centers and permitting the allocation of more requests. By keeping an eye on the GSRL metric that we have defined, the CSP will have more control over the security status of her infrastructure and services and could be able to determine the appropriate settings for the allocation model on the fly according to particular situations. For example, a CSP who notices a strange activity occurring within the monitored network traffic will decide to take effective preventive measures in real-time and immediately decrease the value of supported critical security risk on the data center in order to avoid allocating the requests of malicious or untrusted users or limit the damages of a potential security attack.

Figure 8.10 also stresses this point. In the figure, we see the changes in the rate of allocated requests that have received with the default security configuration SC_0. Both GA and ABC show a stable performance when the critical security risk value is between 0.1 and 0.4. For instance, on average, the ABC algorithm was allocating only up to 45% of requests with SC_0. This goes back to the fact that in order to respect Constraint 8.7 while maximizing the amount of allocated resources, the algorithm will give the priority to those requests that will introduce lower security risk to the infrastructure and keep the value of the GSRL metric small. Hence, it will more likely allocate requests received with advanced security configurations before allocating the requests of customers who are not interested in deploying security solutions along with their Cloud services. This will help in encouraging customers to request advanced security configurations when placing their requests in order to guarantee the allocation. On the other hand, when the value of β_j was equal to 0.5, nearly all the requests that included SC_0 were allocated by the GA, which again, indicates that the value of critical security risk parameter has an important role in the proposed allocation model and should be carefully set to provide the CSP with the desired security status of the system.

8.7 Conclusion

The adoption of Cloud Computing is still slow due to the lack of security and reliability. However, this could change if the CSPs decide to start taking into consideration the security risk that customers could introduce to their infrastructures, and allocate their resources accordingly. This will enable them to reduce the probability of failures and attacks, limit the rate and severity of Security-SLA violations, increase their trustworthiness, and attract more customers. In this work, we first identified the security risk factors in a public IaaS Cloud environment and proposed several security parameters based on which the CSP could offer different security configurations to be purchased by her customers to secure their deployed services. According to their requests, the security risk that customers would present to the Cloud infrastructure was relatively evaluated. Then, we proposed a model for resource allocation in an InterCloud setting that integrates the security risk factor in order to help the CSP in keeping her infrastructure secure to some extent. By implementing this model on her VM central management unit, the CSP will have more control over the security status of her environment. Customers who are willing to invest in security solutions will more likely be given the priority of allocation, and the rate of insecure VMs will be reduced. The allocation problem was solved using two evolutionary computation approaches : the GA and ABC optimization. Results showed that the proposed metaheuristics are able to generate acceptable solutions and could perform in an online mode. In the future, we plan on shedding the light on effectively estimating the critical security risk level that a data center could support using robust Artificial Intelligence architectures.

CHAPTER 9 GENERAL DISCUSSION

In this chapter, we first recall the research objectives that we declared at the beginning of the thesis and evaluate the extent to which they were achieved. Then, we describe the scientific and industrial impacts of our research through the presented contributions. Finally, we discuss about the limitations of our work.

9.1 Objectives achievement

The main objective of this thesis was to provide a Cloud security evaluation model that could be successfully integrated into the fundamental problems of service provisioning and management in Cloud Computing. These include : service selection, application deployment, service placement, resource allocation, and workload federation. More specifically, the following objectives were attained :

— **Evaluate the security level of a Cloud Computing infrastructure.** In the first phase of our work in this thesis, the importance of security quantification in the evaluation process was emphasized, and a set of quantitative security metrics that reflect to some extent the security level of Cloud services was created, following a thorough analysis of the major security vulnerabilities and threats in the Cloud. Then, these metrics were embedded into a security evaluation methodology in which the security services provided by CSPs were numerically assessed to enable a relative evaluation of their capabilities. Afterwards, a set of SSLO parameters was proposed, based on the defined security metrics, to establish a standard version of the Cloud Security-SLA, based on which security offerings can be successfully evaluated and compared. The quantitative modeling of customers' security requirements and CSPs' security capabilities paved the way for the design of a solid service evaluation and matching framework, and made possible the integration of the security element into the definition of the problems described above.

— **Enable Cloud customers to perform an optimized security-driven service selection.** To this end, a security-oriented Cloud service selection framework was designed based on the new Security-SLA model. The framework consists in numerically evaluating the satisfaction of customers' security requirements by CSPs' security offerings with respect to the three security attributes of the CIA triad security model : confidentiality, integrity, and availability, and modeling the service selection problem

as a multi-objective optimization problem. The objective of the problem is to find the Security-SLA that adheres the most to the customer's security requirements and preferences, by effectively deliberating on the trade-off among the three security attributes. The solution consisted in performing an exhaustive search over the set of feasible points and returning a set of efficient points, from which one point is eventually selected according to the weights assigned by the customer to the three attributes. The evaluation and selection model was designed with the objective of highlighting the importance of effectively appraising the different Cloud security aspects that are critical to the deployed service. Indeed, the proposed solution was more focused on demonstrating the applicability of the model and emphasizing the decision making process, regardless of how suitable it is from a computational efficiency perspective. However, to be implementable online and respond to a high number of service deployment requests, the multi-objective optimization problem that was defined in the context of our service selection framework needs to be solved heuristically, especially if it is integrated into a service placement model at the resource allocation level. Also, to complete this work, the satisfaction of other security-related aspects that are of special relevance to Cloud Computing services such as accountability and privacy should be effectively modeled and incorporated into the service selection problem.

— **Enable security-based Cloud service placement.** In the second phase of our work, the Cloud service placement problem, which is usually formulated in the context of QoS optimization or energy-awareness, was approached from a security perspective. The goal was to ensure service security satisfaction during the assignment to the Cloud data centers. In our approach, the service placement problem was defined as a multi-objective optimization problem in which the security requirements of provisioned services were modeled in terms of three different aspects : cost, performance, and risk. The reason for which we have followed a different path in defining the objectives than the one we set while designing the security-oriented service selection framework lies in the fact that service placement is now occurring at the server granularity level, and requires careful attention to the trade-off that needs to be evaluated between the cost and effectiveness of security implementation. The problem was then solved optimally using a mathematical solver. The proposed approach proved to be effective in limiting the violations of Security-SLA, by placing the services on the servers that adhere the most to their requirements. In this work, our focus was oriented towards providing rigorous security modeling that reflects the reality of Cloud Computing infrastructures. However, to provide a scalable solution to the service placement problem, a computationally efficient approach needs to be further elaborated.

— **Create and maintain secure Cloud federations.** The integration of security into the process of federation formation in the Cloud forms an intrinsic part of this thesis. The goal was to enable CSPs to continue to satisfy the security requirements of their customers' workload even when the latter is executed outside their own infrastructures. To this end, we designed two different security-aware online federation formation frameworks. In the first one, the security risk of Cloud Computing infrastructures was assessed according to a set of criteria and incorporated into a cooperative game model that drives the federation formation process. In the second, we evaluated the security level of CSPs with respect to a security baseline that we defined based on the Security-SLA. This allowed for the formed federation to be less dependent on a particular user's security requirements, capable of supplying multiple and heterogeneous service federation requests between CSPs, and eventually more durable. The framework adopts a hedonic coalitional game model in which CSPs specify their preference relationships in terms of the security level of formed federations. The proposed models showed computational efficiency, and demonstrated their ability to reduce the rate and severity of Security-SLA violations when federating service workload between Cloud Computing data centers. With further combination of QoS requirements and profitability modeling, the proposed approaches can be implemented on current Cloud infrastructures to ensure service protection during the federation operations.

— **Support the CSPs to perform security-aware resource allocation.** In the last phase of our work, we addressed the challenges to security integration at the level of resource allocation and provisioning in the Cloud. To support the CSPs in reducing the security risk on their infrastructures, we designed a resource allocation model that integrates the assessment of customers' security risk in terms of the required security implementation specified in their resource provisioning requests. The objective of the model is to maximize resource utilization while keeping the security risk level of the Cloud infrastructure below the critical value. The security risk-aware resource allocation problem was modeled as a linear optimization problem and solved using two different metaheuristics approaches from the family of evolutionary computation : GA and ABC. Both approaches demonstrated their ability to achieve an acceptable approximation of the optimal solution and to be implementable in online mode. The proposed allocation model can be implemented in an InterCloud setting as a solution to limit security risk and reduce threat probability within the data centers.

— **Support the CSPs in reducing the cost of security investment.** To this end, we approached the problem of security integration in the Cloud from a financial angle.

We mathematically modeled the process of secure resource allocation in the Cloud in an auction-based context. In the model, customers are asked to show their valuation of the security of the Cloud resources in their resource provisioning requests. To solve the linear optimization problem while ensuring the truthfulness of the bidders, we proposed a DSIC online mechanism that aims at maximizing customers' social welfare and reducing the cost of security investment by allocating the secure resources to the customers who valuate their security the most. The mechanism implements a greedy-based allocation procedure and a truth-inducing payment rule. The mechanism is computationally efficient and achieves acceptable approximation of the optimal solution usually achieved through the offline VCG truthful mechanism.

9.2 Contributions and impact

Our contributions in this thesis can be summed up to fit into the following two main areas :

— **Enabling reliable evaluation of Cloud Computing security**. Some major contributions were presented in this thesis in light of this objective. These include the definition of standard evaluation criteria and metrics, and the design of robust evaluation frameworks, bearing in mind the high relevance of the quantification aspect of the evaluation. The proposed evaluation approaches have the advantage of being miscellaneous, that is, built on the viable assumption that the defined problem can be tackled from different perspectives to provide realistic and practical solutions. The problem of Cloud security evaluation is critical to the adoption of the Cloud Computing paradigm, and the solutions proposed in this thesis will certainly advance the research in this area. Indeed, the elaboration of standard, quantitative, and measurable security metrics constitutes the first step towards the creation of an effective and trustworthy security evaluation platform that could carry a pivotal role in the process of commercialization of Cloud services. Such platform would be of high importance to all Cloud stakeholders, from service providers and aggregators, to consumers and users.

— **Supporting the security of Cloud services through integration.** Security integration in the context of our work consists in describing the fundamental problems that follow the Cloud service lifetime from a security angle. These include : service selection, placement, provisioning, and federation. Some of the problems were tackled from a high level perspective to provide plausible approaches that can be further elaborated to fit into specific contexts and adapt to particular scenarios. Others were addressed at the low level of the Cloud Computing architecture to reflect more on the realism that accompanies this integration. As demonstrated throughout the thesis, the

proposed solutions could be effectively implemented on current Cloud infrastructures to enable securer Cloud-based service hosting, and eventually rise the adoption of the Cloud Computing paradigm.

9.3 Limitations

The security evaluation approaches proposed in this thesis will require complete cooperation from the CSPs in order to produce effective results and demonstrate their full powers. This cooperation entails a high level of exposure and transparency, which could in some cases, be against CSPs' business policies. Indeed, when we talk about evaluation, that naturally means that data will be collected for assessment. With the absence of these data, the evaluation of the security of Cloud Computing services will stand sterile. This is actually more like a challenge to the success of our proposed solutions than a limitation.

As we mentioned earlier in this chapter, a scalable solution that is able to cope with the high variability in Cloud traffic needs to be designed in the context of the defined security-based service placement problem. Moreover, most of the problems that we identified in this thesis were approached only from a security perspective, since this was our goal from the beginning. However, to be complete, other factors that are essential to the definition of these problems need to be integrated. For instance, to provide a veritable solution to the Cloud federation formation problem, the security element, which we mainly highlighted throughout this thesis, should be combined with QoS parameters which reflect the real motivation behind the federation process. Our goal was to provide and maintain secure Cloud federations, regardless of their achieved performance. Combining the two aspects and creating a unified model is not a straightforward task, since the evaluation of the trade-off among them will be quite challenging.

Another limitation of the proposed solution in the context of security-aware federation formation lies in the adoption of a security baseline, based on which the evaluation of CSPs' security levels was conducted. Although the existence of such baseline, especially if the latter is standardized according to the Cloud Security-SLA, is of high relevance to the problem of Cloud security evaluation, its creation is undoubtedly highly complex. It will require serious efforts to specify this reference security level that should, by definition, reflect the right degree of security capabilities required to protect Cloud services according to their characteristics and delivery models.

CHAPTER 10 CONCLUSION AND RECOMMENDATIONS

The adoption of Cloud Computing technology is on the rise. However, the extent to which its security is assured is still questionable. In this thesis, we presented a novel description of Cloud Computing security in terms of service requirements. We started by designing a security evaluation methodology that covers the main security aspects and architectural components in the Cloud and embeds a set of quantitative metrics that can effectively assess the security levels of CSPs. A standard and measurable Cloud Security-SLA was then proposed and used in the evaluation of security offerings to pave the way for the design of a solid service selection framework in which the trade-off among the different security aspects is rigorously assessed. Security quantification played a major role in placing the proposed security evaluation approaches at the level where they could finally be trusted, adopted, and automated. It also allowed for the successful integration of the security element into the definition of the fundamental problems of service placement, federation, and provisioning in the Cloud. Indeed, the Cloud service placement problem, the Cloud federation formation problem, and the Cloud resource allocation and provisioning problem were all redefined in a security context to empower the security of Cloud services throughout their whole lifetime. The goal was to provide securer Cloud-based service hosting in which security satisfaction is always emphasized. The proposed solutions proved their effectiveness in responding to customers' security requirements and limiting Security-SLA violations, and their ability to be implementable on current Cloud Computing infrastructures.

The following research avenues could be further explored based on the contributions presented in this thesis :

— The defined security metrics could be described more formally using a SLA-specific language to facilitate their collection and integration, and render possible the implementation and automation of the proposed evaluation procedures.

— Low level security metrics and quantitative indicators could be further elaborated to avoid the definition of abstract SSLOs and allow for the effective monitoring of Security-SLAs. The latter can be achieved by integrating automated measurements and designing robust violation prediction models that incorporate the emerging concept of data-driven security.

— Combining QoS and security requirements while defining the problems of Cloud service placement, federation formation, and resource allocation is another research avenue that needs to be closely investigated to provide realistic and adoptable solutions. For

some of these problems, performance is key, and the optimization of QoS parameters enjoys the highest priority, which makes the exploration of the adequate trade-off between security and performance a quite challenging task.

— The design of a federation-specific Security-SLA management framework that enables effective protection at the workload level is also a challenging mission that is not very straightforward. It requires to closely evaluate the security compatibility of CSPs' infrastructures with respect to services' security requirements. This research avenue is worth exploring and matching theory could play a vital role in this area.

— Finally, the integration of artificial intelligence techniques into the problems of optimization of security satisfaction could be the next and most remarkable step in the domain of security of online services. This includes the design of model-free approaches that are intrinsically based on learning from collected data and past experiences to optimize service security and provide on-demand access to security resources.

REFERENCES

[1] C. Tweaks, "Driven by expansion in public cloud," 2017, (visited on 2018-01-16). [Online]. Available : https://cloudtweaks.com/2017/10/worldwide-cloud-infrastructure-revenues-grow-25-8-second-quarter-2017/

[2] J. Dean and S. Ghemawat, "MapReduce : simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[3] AMAZON, "Amazon EC2," 2010, (visited on 2016-03-21). [Online]. Available : https://aws.amazon.com/fr/ec2/

[4] J. Su, "Hybrid Cloud vs. Multi-Cloud : What's the Difference ?" 2017, (visited on 2018-03-11). [Online]. Available : https://www.bmc.com/blogs/hybrid-cloud-vs-multi-cloud-whats-the-difference/

[5] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. M. Sadjadi, and M. Parashar, "Cloud federation in a layered service model," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1330–1344, 2012.

[6] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 843–859, 2013.

[7] H. Tianfield, "Security issues in cloud computing," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on.* IEEE, 2012, pp. 1082–1089.

[8] K. Hashizume, D. G. Rosado, E. Fernández-Medina, and E. B. Fernandez, "An analysis of security issues for cloud computing," *Journal of internet services and applications*, vol. 4, no. 1, p. 5, 2013.

[9] C. S. Alliance, "CSA Security, Trust and Assurance Registry (STAR)," (visited on 2014-08-18). [Online]. Available : https://cloudsecurityalliance.org/star

[10] ——, "The treacherous twelve cloud computing top threats in 2016," 2016, (visited on 2014-08-18). [Online]. Available : https://cloudsecurityalliance.org/star

[11] Forbes, "2017 state of cloud adoption and security," 2017, (visited on 2018-02-11). [Online]. Available : https://www.forbes.com/sites/louiscolumbus/2017/04/23/2017-state-of-cloud-adoption-and-security/#5788f24f1848

[12] J. Tsidulko, "The 10 biggest cloud outages of 2017," 2017, (visited on 2018-04-19). [Online]. Available : http://www.crn.com/slide-shows/cloud/300089786/the-10-biggest-cloud-outages-of-2017-so-far.htm/pgno/0/9

[13] ——, "The 10 biggest cloud outages of 2016," 2016, (visited on 2018-04-19). [Online]. Available : https://www.crn.com/slide-shows/cloud/300083247/the-10-biggest-cloud-outages-of-2016.htm

[14] P. I. LLC, "2016 cost of data breach study : Global analysis." 2016, (visited on 2018-02-11). [Online]. Available : https://www.forbes.com/sites/louiscolumbus/2017/04/23/2017-state-of-cloud-adoption-and-security/#5788f24f1848

[15] A. E. Eiben and J. E. Smith, "Introduction to evolutionary computing (natural computing series)," *Publisher : Springer-Verlag New York, LLC*, 2008.

[16] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems," in *International fuzzy systems association world congress*. Springer, 2007, pp. 789–798.

[17] J. Siegel and J. Perdue, "Cloud services measures for global use : the service measurement index (SMI)," in *SRII Global Conference (SRII), 2012 Annual*. IEEE, 2012, pp. 411–415.

[18] S. Wang, Z. Liu, Q. Sun, H. Zou, and F. Yang, "Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing," *Journal of Intelligent Manufacturing*, vol. 25, no. 2, pp. 283–291, 2014.

[19] W. Tang and Z. Yan, "CloudRec : A Mobile Cloud Service Recommender System Based on Adaptive QoS Management," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1. IEEE, 2015, pp. 9–16.

[20] M. Tang, X. Dai, J. Liu, and J. Chen, "Towards a trust evaluation middleware for cloud service selection," *Future Generation Computer Systems*, vol. 74, pp. 302–312, 2017.

[21] S. Ristov and M. Gusev, "A methodology to evaluate the trustworthiness of cloud service providers' availability," in *EUROCON 2015-International Conference on Computer as a Tool (EUROCON), IEEE*. IEEE, 2015, pp. 1–6.

[22] S. K. Garg, S. Versteeg, and R. Buyya, "A framework for ranking of cloud computing services," *Future Generation Computer Systems*, vol. 29, no. 4, pp. 1012–1023, 2013.

[23] J. Dodgson, M. Spackman, A. Pearman, and L. Phillips, "Multi-criteria analysis : a manual. department for communities and local government : London," 2009.

[24] V. Casola, A. R. Fasolino, N. Mazzocca, and P. Tramontana, "An AHP-based framework for quality and security evaluation," in *Computational Science and Engineering, 2009. CSE'09. International Conference on*, vol. 3. IEEE, 2009, pp. 405–411.

[25] M. Whaiduzzaman, A. Gani, N. B. Anuar, M. Shiraz, M. N. Haque, and I. T. Haque, "Cloud service selection using multicriteria decision analysis," *The Scientific World Journal*, vol. 2014, 2014.

[26] S. Wibowo, H. Deng, and W. Xu, "Evaluation of cloud services : A fuzzy multi-criteria group decision making method," *Algorithms*, vol. 9, no. 4, p. 84, 2016.

[27] Z. ur Rehman, F. K. Hussain, and O. K. Hussain, "Towards multi-criteria cloud service selection," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*. IEEE, 2011, pp. 44–48.

[28] C. S. Alliance, "CSA Security, Trust & Assurance Registry (STAR," (visited on 2014-09-11). [Online]. Available : https://cloudsecurityalliance.org/star/#_overview

[29] ——, "CSA cloud security alliance cloud controls matrix (CCM), v1.2," (visited on 2014-09-11). [Online]. Available : https://cloudsecurityalliance.org/research/initiatives/cloud-controls-matrix

[30] S. S. Rizvi, T. A. Bolish, and J. R. Pfeffer III, "Security evaluation of cloud service providers using third party auditors," in *Proceedings of the Second International Conference on Internet of things and Cloud Computing*. ACM, 2017, p. 106.

[31] S. Ristov, M. Gusev, and M. Kostoska, "A new methodology for security evaluation in cloud computing," in *MIPRO, 2012 Proceedings of the 35th International Convention*. IEEE, 2012, pp. 1484–1489.

[32] "ISO/IEC 27001 :2013, Information Security Management Systems - Requirements," 2013.

[33] S. Ristov and M. Gusev, "Security evaluation of open source clouds," in *EUROCON, 2013 IEEE*. IEEE, 2013, pp. 73–80.

[34] A. Akinbi, E. Pereira, and C. Beaumont, "Evaluating security mechanisms implemented on public Platform-as-a-Service cloud environments case study : Windows Azure," in *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*. IEEE, 2013, pp. 162–167.

[35] T. Probst, E. Alata, M. Kaâniche, V. Nicomette, and Y. Deswarte, "An approach for security evaluation and analysis in cloud computing," in *Safecomp 2013 FastAbstract*, 2013, p. NC.

[36] A. Abuhussein, F. Alsubaei, S. Shiva, and F. T. Sheldon, "Evaluating security and privacy in cloud services," in *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual*, vol. 1. IEEE, 2016, pp. 683–686.

[37] L. Lin, T. Liu, J. Hu, and J. Zhang, "A privacy-aware cloud service selection method toward data life-cycle," in *Parallel and Distributed Systems (ICPADS), 2014 20th IEEE International Conference on*. IEEE, 2014, pp. 752–759.

[38] G. F. Yu Bengong, Wang Liu, "Security evaluation model for the enterprise cloud services based on grey fuzzy AHP," in *COMPUTER MODELLING & NEW TECHNOLOGIES*, 2014, pp. 239–244.

[39] J. Luna, H. Ghani, D. Germanus, and N. Suri, "A security metrics framework for the cloud," in *Security and Cryptography (SECRYPT), 2011 Proceedings of the International Conference on.* IEEE, 2011, pp. 245–250.

[40] O. Mirković, "Security evaluation in cloud," in *Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on.* IEEE, 2013, pp. 1088–1093.

[41] C. A. Da Silva, A. S. Ferreira, and P. L. de Geus, "A methodology for management of cloud computing using security criteria," in *Cloud Computing and Communications (LATINCLOUD), 2012 IEEE Latin America Conference on.* IEEE, 2012, pp. 49–54.

[42] R. Van Solingen, V. Basili, G. Caldiera, and H. D. Rombach, "Goal Question Metric (GQM) approach," *Encyclopedia of software engineering*, 2002.

[43] M. Jouini and L. B. A. Rabai, "Comparative study of information security risk assessment models for cloud computing systems," *Procedia Computer Science*, vol. 83, pp. 1084–1089, 2016.

[44] A. Chopra, P. Prasad, A. Alsadoon, S. Ali, and A. Elchouemi, "Cloud computing potability with risk assessment," in *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2016 4th IEEE International Conference on.* IEEE, 2016, pp. 53–59.

[45] H. Tang, J. Yang, X. Wang, and Q. Zhou, "A research for cloud computing security risk assessment," *The Open Cybernetics & Systemics Journal*, vol. 10, no. 1, 2016.

[46] Y. Y. Haimes, "Hierarchical holographic modeling," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 9, pp. 606–617, 1981.

[47] A. B. Aissa, R. K. Abercrombie, F. T. Sheldon, and A. Mili, "Quantifying security threats and their potential impacts : a case study," *Innovations in Systems and Software Engineering*, vol. 6, no. 4, pp. 269–281, 2010.

[48] A. S. Sendi and M. Cheriet, "Cloud computing : A risk assessment model," in *Cloud Engineering (IC2E), 2014 IEEE International Conference on.* IEEE, 2014, pp. 147–152.

[49] S. Tanimoto, R. Sato, K. Kato, M. Iwashita, Y. Seki, H. Sato, and A. Kanai, "A study of risk assessment quantification in cloud computing," in *Network-Based Information Systems (NBiS), 2014 17th International Conference on.* IEEE, 2014, pp. 426–431.

[50] E. Cayirci, A. Garaga, A. S. De Oliveira, and Y. Roudier, "A risk assessment model for selecting cloud service providers," *Journal of Cloud Computing*, vol. 5, no. 1, p. 14, 2016.

[51] "MUSA : MULTI-CLOUD SECURE APPLICATIONS," (visited on 2018-04-13). [Online]. Available : http://www.tut.fi/musa-project/

[52] V. Casola, A. De Benedictis, M. Rak, and U. Villano, "SLA-based secure cloud application development : The SPECS framework," in *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2015 17th International Symposium on.* IEEE, 2015, pp. 337–344.

[53] V. Casola, A. De Benedictis, and M. Rak, "Security monitoring in the cloud : An SLA-based approach," in *Availability, Reliability and Security (ARES), 2015 10th International Conference on.* IEEE, 2015, pp. 749–755.

[54] V. Casola, A. De Benedictis, J. Modic, M. Rak, and U. Villano, "Per-service security SLA : a new model for security management in clouds," in *Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE), 2016 IEEE 25th International Conference on.* IEEE, 2016, pp. 83–88.

[55] C. A. B. De Carvalho, R. M. de Castro Andrade, M. F. de Castro, E. F. Coutinho, and N. Agoulmine, "State of the art and challenges of security SLA for cloud computing," *Computers & Electrical Engineering*, vol. 59, pp. 141–152, 2017.

[56] M. A. T. Rojas, N. M. Gonzalez, F. Sbampato, F. Redigolo, T. C. M. de Brito Carvalho, K. K. Nguyen, and M. Cheriet, "Inclusion of security requirements in SLA lifecycle management for cloud computing," in *Evolving Security and Privacy Requirements Engineering (ESPRE), 2015 IEEE 2nd Workshop on.* IEEE, 2015, pp. 7–12.

[57] Y. Rahulamathavan, P. S. Pawar, P. Burnap, M. Rajarajan, O. F. Rana, and G. Spanoudakis, "Analysing security requirements in cloud-based service level agreements," in *Proceedings of the 7th International Conference on Security of Information and Networks.* ACM, 2014, p. 73.

[58] S. A. de Chaves, C. B. Westphall, and F. R. Lamin, "SLA perspective in security management for cloud computing," in *Networking and Services (ICNS), 2010 Sixth International Conference on.* IEEE, 2010, pp. 212–217.

[59] J. Luna, N. Suri, M. Iorga, and A. Karmel, "Leveraging the potential of cloud security service-level agreements through standards," *IEEE Cloud Computing*, vol. 2, no. 3, pp. 32–40, 2015.

[60] P. H. Meland, K. Bernsmed, M. G. Jaatun, A. Undheim, and H. N. C. Martínez, "Expressing cloud security requirements in deontic contract languages." in *CLOSER*, 2012, pp. 638–646.

[61] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, "Web services agreement specification (WS-Agreement)," in *Open grid forum*, vol. 128, no. 1, 2007, p. 216.

[62] M. L. Hale and R. Gamble, "Building a compliance vocabulary to embed security controls in cloud SLAs," in *Services (SERVICES), 2013 IEEE Ninth World Congress on*. IEEE, 2013, pp. 118–125.

[63] A. Guesmi and P. Clemente, "Access control and security properties requirements specification for clouds' seclas," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, vol. 1. IEEE, 2013, pp. 723–729.

[64] M. A. Rojas, N. M. Gonzalez, F. V. Sbampato, F. F. Redígolo, T. Carvalho, K. W. Ullah, M. Näslund, and A. S. Ahmed, "A framework to orchestrate security sla lifecycle in cloud computing," in *Information Systems and Technologies (CISTI), 2016 11th Iberian Conference on*. IEEE, 2016, pp. 1–7.

[65] C. A. Da Silva and P. L. de Geus, "An approach to security-SLA in cloud computing environment," in *Communications (LATINCOM), 2014 IEEE Latin-America Conference on*. IEEE, 2014, pp. 1–6.

[66] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, "Security SLAs for federated cloud services," in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*. IEEE, 2011, pp. 202–209.

[67] A. Taha, R. Trapero, J. Luna, and N. Suri, "AHP-based quantitative approach for assessing and comparing cloud security," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*. IEEE, 2014, pp. 284–291.

[68] J. Luna, A. Taha, R. Trapero, and N. Suri, "Quantitative reasoning about cloud security using service level agreements," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 457–471, 2017.

[69] S.-H. Na and E.-N. Huh, "A broker-based cooperative security-SLA evaluation methodology for personal cloud computing," *Security and Communication networks*, vol. 8, no. 7, pp. 1318–1331, 2015.

[70] R. Los, D. Shackleford, and B. Sullivan, "The notorious nine cloud computing top threats in 2013," *Cloud Security Alliance*, 2013.

[71] J. Zhengwei, D. Ran, L. Zhigang, W. Xihong, and L. Baoxu, "A meta-synthesis approach for cloud service provider selection based on secsla," in *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on.* IEEE, 2013, pp. 1356–1360.

[72] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition : A systematic literature review," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3809–3824, 2014.

[73] H. Bouzary and F. F. Chen, "Service optimal selection and composition in cloud manufacturing : a comprehensive survey," *The International Journal of Advanced Manufacturing Technology*, pp. 1–14, 2018.

[74] O. Wenge, U. Lampe, and R. Steinmetz, "QoS-and Security-aware Composition of Cloud Collaborations." in *CLOSER*, 2014, pp. 578–583.

[75] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "Security-aware service composition with fine-grained information flow control," *IEEE Transactions on services computing*, vol. 6, no. 3, pp. 330–343, 2013.

[76] M. Albanese, S. Jajodia, and C. Molinaro, "A logic framework for flexible and security-aware service composition," in *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC).* IEEE, 2013, pp. 337–346.

[77] A. Guesmi, P. Clemente, F. Loulergue, and P. Berthomé, "Cloud resources placement based on functional and non-functional requirements," in *e-Business and Telecommunications (ICETE), 2015 12th International Joint Conference on*, vol. 4. IEEE, 2015, pp. 335–342.

[78] F. Xiang, Y. Hu, Y. Yu, and H. Wu, "QoS and energy consumption aware service composition and optimal-selection based on Pareto group leader algorithm in cloud manufacturing system," vol. 22, no. 4. Springer, 2014, pp. 663–685.

[79] J. Zhou and X. Yao, "A hybrid approach combining modified artificial bee colony and cuckoo search algorithms for multi-objective cloud manufacturing service composition," vol. 55, no. 16. Taylor & Francis, 2017, pp. 4765–4784.

[80] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, "Cloud federation," *Cloud Computing*, vol. 2011, pp. 32–38, 2011.

[81] M. R. Assis and L. F. Bittencourt, "A survey on cloud federation architectures : identifying functional and non-functional properties," *Journal of Network and Computer Applications*, vol. 72, pp. 51–71, 2016.

[82] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, "Thunder in the clouds : Security challenges and solutions for federated clouds," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on.* IEEE, 2012, pp. 113–120.

[83] H. Li, C. Wu, Z. Li, and F. C. Lau, "Profit-maximizing virtual machine trading in a federation of selfish clouds," in *INFOCOM, 2013 Proceedings IEEE.* IEEE, 2013, pp. 25–29.

[84] N. Samaan, "A novel economic sharing model in a federation of selfish cloud providers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 12–21, 2014.

[85] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky : Formation game and mechanism," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 14–27, 2015.

[86] M. Guazzone, C. Anglano, and M. Sereno, "A game-theoretic approach to coalition formation in green cloud federations," in *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on.* IEEE, 2014, pp. 618–625.

[87] M. M. Hassan, M. Abdullah-Al-Wadud, A. Almogren, S. Rahman, A. Alelaiwi, A. Alamri, M. Hamid *et al.*, "QoS and trust-aware coalition formation game in data-intensive cloud federations," vol. 28, no. 10. Wiley Online Library, 2016, pp. 2889–2905.

[88] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Towards trustworthy multi-cloud services communities : A trust-based hedonic coalitional game." IEEE, 2016.

[89] M. Cardosa, M. R. Korupolu, and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in *Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on.* IEEE, 2009, pp. 327–334.

[90] D. Lu, J. Ma, C. Sun, X. Ma, and N. Xi, "Credit-based scheme for security-aware and fairness-aware resource allocation in cloud computing," *Science China Information Sciences*, vol. 60, no. 5, p. 52103, 2017.

[91] V. Nandina, J. M. Luna, C. C. Lamb, G. L. Heileman, and C. T. Abdallah, "Provisioning security and performance optimization for dynamic cloud environments," in *Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on.* IEEE, 2014, pp. 979–981.

[92] Y. Liu and M. J. Lee, "Security-aware resource allocation for mobile cloud computing systems," in *Computer Communication and Networks (ICCCN), 2015 24th International Conference on.* IEEE, 2015, pp. 1–8.

[93] H. Liang, D. Huang, L. X. Cai, X. Shen, and D. Peng, "Resource allocation for security services in mobile cloud computing," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*. IEEE, 2011, pp. 191–195.

[94] H. Goudarzi and M. Pedram, "Maximizing profit in cloud computing system via resource allocation," in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*. IEEE, 2011, pp. 1–6.

[95] A. Nezarat and G. Dastghaibyfard, "A game theoretical model for profit maximization resource allocation in cloud environment with budget and deadline constraints," *The Journal of Supercomputing*, vol. 72, no. 12, pp. 4737–4770, 2016.

[96] L. Mashayekhy, M. M. Nejad, D. Grosu, and A. V. Vasilakos, "An online mechanism for resource allocation and pricing in clouds," *IEEE transactions on computers*, vol. 65, no. 4, pp. 1172–1184, 2016.

[97] H. Zhang, H. Jiang, B. Li, F. Liu, A. V. Vasilakos, and J. Liu, "A framework for truthful online auctions in cloud computing with heterogeneous user demands," *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 805–818, 2016.

[98] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge university press, 2007.

[99] Microsoft, "The STRIDE Threat Model," (visited on 2017-05-13). [Online]. Available : https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

[100] E. S. Frederic de Vaulx and R. Bohn, "NIST Special Publication 500-307 Cloud Computing Service Metrics Description," 2015.

[101] K. Deb, "An ideal evolutionary multi-objective optimization procedure," *IPSJ Transactions on Mathematical Modeling and Its Applications*, vol. 45, pp. 1–11, 2004.

[102] E. K. Lee, "Optimization with multiple objectives," *Industrial and Systems Engineering, Georgia Institute of Technology, Washington DC*, 2002.

[103] H. D. Sherali and A. L. Soyster, "Preemptive and nonpreemptive multi-objective programming : Relationship and counterexamples," *Journal of Optimization Theory and Applications*, vol. 39, no. 2, pp. 173–186, 1983.

[104] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, "Coalition structure generation with worst case guarantees," *Artificial Intelligence*, vol. 111, no. 1-2, pp. 209–238, 1999.

[105] A. Bogomolnaia and M. O. Jackson, "The stability of hedonic coalition structures," *Games and Economic Behavior*, vol. 38, no. 2, pp. 201–230, 2002.

[106] H. Kellerer, U. Pferschy, and D. Pisinger, "Knapsack problems. 2004," 2003.

[107] D. C. Parkes, "Online mechanisms," 2007.

[108] "National Institute of Standards and Technology (NIST)," (visited on 2017-05-13). [Online]. Available : http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html

[109] I. M. Khalil, A. Khreishah, and M. Azeem, "Cloud computing security : A survey," *Computers*, vol. 3, no. 1, pp. 1–35, 2014.

[110] K. Weins, "Cloud computing trends : 2014 state of the cloud survey," 2014, (visited on 2017-05-17). [Online]. Available : http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2014-state-cloud-survey

[111] A. Reed, C. Rezek, and P. Simmonds, "Security guidance for critical areas of focus in cloud computing v3. 0," *Cloud Security Alliance*, 2011.

[112] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of network and computer applications*, vol. 34, no. 1, pp. 1–11, 2011.

[113] N. Gonzalez, C. Miers, F. Redigolo, M. Simplicio, T. Carvalho, M. Näslund, and M. Pourzandi, "A quantitative analysis of current security concerns and solutions for cloud computing," *Journal of Cloud Computing : Advances, Systems and Applications*, vol. 1, no. 1, p. 11, 2012.

[114] R. Bhadauria and S. Sanyal, "Survey on security issues in cloud computing and associated mitigation techniques," *arXiv preprint arXiv :1204.0764*, 2012.

[115] J. Sen, "Security and privacy issues in cloud computing," *Architectures and protocols for secure information technology infrastructures*, pp. 1–45, 2013.

[116] J. E. Bryson, "NIST Special Publication 800-53 Revision 4 Security and Privacy Controls for Federal Information Systems and Organizations JOINT TASK FORCE TRANSFORMATION INITIATIVE," 2012.

[117] "Cloud security alliance group CSA-GRC stack," (visited on 2014-09-11). [Online]. Available : https://cloudsecurityalliance.org/research/

[118] C. A. Da Silva and P. L. De Geus, "Return on security investment for cloud computing : a customer perspective," in *Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems*. ACM, 2015, pp. 156–160.

[119] J. Luna Garcia, R. Langenberg, and N. Suri, "Benchmarking cloud security level agreements using quantitative policy trees," in *Proceedings of the 2012 ACM Workshop on Cloud computing security workshop*. ACM, 2012, pp. 103–112.

[120] O. W. A. S. P. (OWASP), "OWASP Top 10 Application Security Risks - 2017," 2017, (visited on 2018-05-13). [Online]. Available : https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project

[121] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM conference on Computer and communications security*. Acm, 2007, pp. 598–609.

[122] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing." *HotCloud*, vol. 9, no. 9, p. 3, 2009.

[123] U. Kumar and B. N. Gohil, "A survey on intrusion detection systems for cloud computing environment," *International Journal of Computer Applications*, vol. 109, no. 1, 2015.

[124] A. Banka, A. Saravgi, M. Sain, and H. J. Lee, "Exploration of security parameters to evaluate saas," in *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*. IEEE, 2013, pp. 1–6.

[125] E. Chew, M. M. Swanson, K. M. Stine, N. Bartol, A. Brown, and W. Robinson, "Performance measurement guide for information security," Tech. Rep., 2008.

[126] M. Rouse, "Security Information and Event Management (SIEM)," 2012.

[127] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring : A survey," *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013.

[128] Statista, "Market growth forecast for public cloud services worldwide from 2011 to 2020," 2015, (visited on 2016-02-19). [Online]. Available : https://www.statista.com/statistics/203578/global-forecast-of-cloud-computing-services-growth/

[129] T. Halabi and M. Bellaiche, "Towards quantification and evaluation of security of cloud service providers," *Journal of Information Security and Applications*, vol. 33, pp. 55–65, 2017.

[130] ——, "Evaluation and selection of Cloud security services based on Multi-Criteria Analysis," in *Computing, Networking and Communications (ICNC), 2017 International Conference on*. IEEE, 2017, pp. 706–710.

[131] M. Rak, N. Suri, J. Luna, D. Petcu, V. Casola, and U. Villano, "Security as a service using an SLA-based approach via SPECS," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, vol. 2. IEEE, 2013, pp. 1–6.

[132] V. Casola, M. Rak, and G. Alfieri, "A cloud application for security service level agreement evaluation." in *CLOSER*, 2014, pp. 299–307.

[133] R. Trapero, J. Modic, M. Stopar, A. Taha, and N. Suri, "A novel approach to manage cloud security SLA incidents," *Future Generation Computer Systems*, vol. 72, pp. 193–205, 2017.

[134] T. Halabi and M. Bellaiche, "How to Evaluate The Defense Against DoS and DDoS Attacks in Cloud Computing : A Survey and Taxonomy," *International Journal of Computer Science and Information Security*, vol. 14, no. 12, p. 1, 2016.

[135] G. Narzisi, "Classic Methods for Multi-Objective Optimization," 2008.

[136] W. Wei, J. Du, T. Yu, and X. Gu, "Securemr : A service integrity assurance framework for mapreduce," in *Computer Security Applications Conference, 2009. ACSAC'09. Annual*. IEEE, 2009, pp. 73–82.

[137] B. B. G. Abadi and M. G. Arani, "Resource management of iaas providers in cloud federation," *International Journal of Grid and Distributed Computing*, vol. 8, no. 5, pp. 327–336, 2015.

[138] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud : Utility-oriented federation of cloud computing environments for scaling of application services," in *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 2010, pp. 13–31.

[139] MathWorks, "Mixed-integer linear programming (MILP) - MATLAB intlinprog."

[140] M. Chiregi and N. J. Navimipour, "A new method for trust and reputation evaluation in the cloud environments using the recommendations of opinion leaders' entities and removing the effect of troll entities," *Computers in Human Behavior*, vol. 60, pp. 280–292, 2016.

[141] I. Petri, T. Beach, M. Zou, J. D. Montes, O. Rana, and M. Parashar, "Exploring models and mechanisms for exchanging resources in a federated cloud," in *Cloud Engineering (IC2E), 2014 IEEE International Conference on*. IEEE, 2014, pp. 215–224.

[142] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization : new insights," *Structural and multidisciplinary optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[143] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, "Cloud federation," *Cloud Computing*, vol. 2011, pp. 32–38, 2011.

[144] M. J. Osborne, *An introduction to game theory*. Oxford university press New York, 2004, vol. 3, no. 3.

[145] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes, "A selfish approach to coalition formation among unmanned air vehicles in wireless networks," in *Game Theory*

for Networks, 2009. GameNets' 09. International Conference on. IEEE, 2009, pp. 259–267.

[146] ——, "Hedonic coalition formation for distributed task allocation among wireless agents," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1327–1344, 2011.

[147] G. Coulouris, J. Dollimore, and T. Kindberg, "Distributed systems : Concepts and design : Pearson education," 2005.

[148] A. D. Kshemkalyani and M. Singhal, *Distributed computing : principles, algorithms, and systems.* Cambridge University Press, 2011.

[149] K. Djemame, D. Armstrong, J. Guitart, and M. Macias, "A risk assessment framework for cloud computing," *IEEE Transactions on Cloud Computing*, vol. 4, no. 3, pp. 265–278, 2016.

[150] T. Halabi and M. Bellaiche, "A broker-based framework for standardization and management of Cloud Security-SLAs," *Computers & Security*, vol. 75, pp. 59–71, 2018.

[151] ——, "Towards security-based formation of cloud federations : A game theoretical approach," *IEEE Transactions on Cloud Computing*, 2018.

[152] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[153] N. Quang-Hung, P. D. Nien, N. H. Nam, N. H. Tuong, and N. Thoai, "A genetic algorithm for power-aware virtual machine allocation in private cloud," in *Information and Communication Technology-EurAsia Conference.* Springer, 2013, pp. 183–191.

[154] G. Portaluri, S. Giordano, D. Kliazovich, and B. Dorronsoro, "A power efficient genetic algorithm for resource allocation in cloud computing data centers," in *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on.* IEEE, 2014, pp. 58–63.

[155] N. K. Sharma and R. M. R. Guddeti, "Multi-objective resources allocation using improved genetic algorithm at cloud data center," in *Cloud Computing in Emerging Markets (CCEM), 2016 IEEE International Conference on.* IEEE, 2016, pp. 73–77.

[156] M. A. Khan, "A survey of security issues for cloud computing," *Journal of network and computer applications*, vol. 71, pp. 11–29, 2016.

[157] T. Blickle and L. Thiele, "A comparison of selection schemes used in genetic algorithms," 1995.

[158] J. Clausen, "Branch and bound algorithms-principles and examples," *Department of Computer Science, University of Copenhagen*, pp. 1–30, 1999.

[159] K. A. De Jong and W. M. Spears, "An analysis of the interacting roles of population size and crossover in genetic algorithms," in *International Conference on Parallel Problem Solving from Nature*. Springer, 1990, pp. 38–47.

[160] G. Ataş and V. C. Gungor, "Performance evaluation of cloud computing platforms using statistical methods," *Computers & Electrical Engineering*, vol. 40, no. 5, pp. 1636–1649, 2014.

[161] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017.

[162] H. Aziz, F. Brandt, and P. Harrenstein, "Fractional hedonic games," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 5–12.

[163] T. Halabi and M. Bellaiche, "Service assignment in federated cloud environments based on multi-objective optimization of security," in *Future Internet of Things and Cloud (FiCloud), 2017 IEEE 5th International Conference on*. IEEE, 2017, pp. 39–46.

[164] S. Zaman and D. Grosu, "An online mechanism for dynamic VM provisioning and allocation in clouds," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 2012, pp. 253–260.

[165] M. M. Nejad, L. Mashayekhy, and D. Grosu, "Truthful greedy mechanisms for dynamic virtual machine provisioning and allocation in clouds," *IEEE transactions on parallel and distributed systems*, vol. 26, no. 2, pp. 594–603, 2015.

APPENDIX A

ARTICLE 6 : EVALUATION AND SELECTION OF CLOUD SECURITY SERVICES BASED ON MULTI-CRITERIA ANALYSIS MCA

Talal Halabi and Martine Bellaiche

International Conference on Computing, Networking and Communications, pp. 706-710, 2017.

Abstract

Security is still the main obstacle that is preventing businesses from moving towards the Cloud, which makes choosing the right Cloud Service Provider (CSP) a critical decision. We propose in this paper a methodology for the evaluation and selection of Cloud security services based on a Multi-Criteria Analysis (MCA) process using a set of evaluation criteria and quantitative metrics. We then give a general overview of the design and requirements of a future implementation of a Cloud security evaluation architecture.

sectionIntroduction Cloud Computing technology introduces many technical and financial advantages into the IT and business domains, such as resource pooling, rapid elasticity, and measured service. However, security concerns are still retarding the Cloud adoption especially for companies that deal with private information and confidential data due to the Cloud's specific features that introduce new vulnerabilities and threats within its different layers such as the very large scale and the virtualization of resources. Thus, selecting the appropriate level of security when migrating businesses to the Cloud is a key factor in order to maintain the protection and performance efficiency of the service, and reduce the violation of security contracts.

Previous research has tackled the issue of Cloud security quantification and the evaluation and selection of security services. However, the current Cloud lacks the existence of an efficient security evaluation technique that is based on standard criteria and quantitative metrics which cover all parts of its architecture (virtualization, network, storage, etc.). When moving critical businesses and sensitive information to the Cloud, the protection against different kinds of threats and attacks becomes essential. Security in the Cloud is usually provided through the following four aspects :

— Cloud confidentiality, which concerns with protecting the sensitive information from unauthorized disclosure using strong authentication techniques, adequate Identity and Access Management solutions (IAM), protection of the web interface and network, and also the hardware and virtualization layers.
— Cloud integrity, which concerns with accuracy and completeness of information and computation in regards with business requirements and expectations, and is provided by protecting the data at rest, in transit, and while processed, both cryptographically and physically from loss or manipulation.
— Cloud availability, which concerns with information being operational and accessible whenever required by the business process, and is maintained by implementing effective incident management plans (response and recovery), application change management procedures, and protection against Denial of Service (DoS) attacks (Intrusion Detection/Prevention Systems (IDS/IPS)).
— Cloud accountability and compliance with security standards (e.g., ISO 27001 and HIPAA [32]), which concerns with keeping track of actions related to security responsibilities and violation of regulations and Security Service Level Agreements (Security-SLA) through security audit and vulnerability assessment plans.

Evaluating Cloud security consists in measuring the level of each of these services in the context of specific criteria. The contribution of this paper is threefold :

— First, we define a set of hierarchically structured security evaluation criteria inspired by the nature of security mechanisms implemented in a Cloud environment, and a set of quantitative metrics that can effectively measure the level of security in the context of each criterion.
— Second, we propose a security evaluation methodology based on a Multi-Criteria Analysis (MCA) [23] process that considers both objective and subjective weighting strategies, in order to fully benefit from the nature of evaluation data and consider the preferences and requirements of Cloud consumers at the same time.
— Third, we give a general overview of the design and requirements of the implementation of a Cloud security service evaluation and selection architecture.

This paper is organized as follows. Section A.1 discusses related work. Section A.2 discusses the proposed evaluation criteria and metrics. In section A.3 the evaluation and selection methodology is elaborated. The requirements of the implementation are explained in section A.4. Finally, section A.5 concludes the paper.

A.1 Related Work

NIST [108] and CSA [9] are currently putting a lot of efforts to reach Cloud security standardization through different projects like the Cloud Computing Service Metrics Description [100], and the Cloud Controls Matrix (CCM) [29]. Da Silva et al. [41, 65] proposed an approach to Security-SLA based on a security metrics hierarchy that describes the security level in a Cloud Computing environment. Mirković [40] proposed a measurable model for the Cloud by defining a system of metrics based on the ISO 27001 standard security controls [32]. However, these proposals did not consider all critical security aspects and services in the Cloud.
Zhengwei et al. [71] constructed a quantifiable Cloud-oriented Security-SLA following a metrics modeling method. Na and Huh [69] proposed a security-based Cloud service selection method through weights evaluation using the Analytical Hierarchy Process (AHP) method. Taha et al. [67] also used the AHP technique to benchmark Cloud security based on the Security-SLA provided by the Consensus Assessments Initiative Questionnaire (CAIQ) of the CSA. However, these proposals did not discuss about the evaluation metrics that can quantify and measure the security of CSPs.

A.2 Evaluation Criteria and Metrics

We define in this section a set of critical criteria based on which a thorough security evaluation in Cloud Computing could be conducted. We consider three different aspects while defining this set : 1) determining if CSPs are providing sufficient security for their infrastructures and services, 2) characterizing the effective performance of Cloud security services, and 3) describing the effect of security implementation on the performance of Cloud services and overall cost. In order to quantitatively evaluate the Cloud security services in the context of each of the criteria, a set of measurable metrics was also defined, taking into consideration the nature of Cloud security mechanisms and procedures that are usually deployed. Table A.1 shows some of the developed evaluation metrics. We describe in the following the proposed criteria and Figure A.1 shows their hierarchical structure.

— Performance criterion, which describes if the security mechanisms are functioning correctly and effectively in order to meet the desired outcomes. It can be evaluated based on the effectiveness of security mechanisms in terms of accuracy, functionality, and response time, and their failure which can cause data or financial losses, and violations of the Security-SLA, since the shared nature of the Cloud introduces many new threats to the security of consumers' applications.

— Cost criterion, which characterizes the impact of implementing security mechanisms on the Cloud resources and applications' performance. A security mechanism can sometimes cause performance degradation to the Cloud service which can be reflected as a financial loss. Information related to performance and cost criteria can be provided through multiple sources such as security auditing and testing tools, vulnerability scanning and risk assessment systems, and the (SIEM) system [126].
— Implementation criterion, which helps demonstrating the progress in implementing information security programs, controls, and related policies and procedures, and can be evaluated based on three sub-criteria : continuity, dynamicity, and flexibility. CSPs can

Table A.1 An example set of metrics for evaluation of Cloud security services.

Performance
Effectiveness False Acceptance Rate and False Rejection Rate of authentication mechanisms ; frequency of review of access control logs and accounts' activity ; injection attacks and malicious activities detection rate by web application scanning ; mean-time to discover and mitigate network attacks ; false positives rate ; Latency ; Packet drop rate ; frequency of update of anti-malware and anti-virus programs ; strength of encryption keys and block ciphers ; mean-time of incident recovery ; recovery time of failed tasks ; Cloud failure rate ; frequency and coverage of risk assessments and vulnerability scans. **Failure** % Data loss or application offline-time due to authentication failure ; financial loss or application offline-time due to incidents.
Cost
Resource overhead Processing time, CPU load or processing usage, memory overhead, and bandwidth overhead of IDS ; percent of computational, communication, and storage overhead of data protection mechanisms ; computation time of PDP techniques ; bandwidth overhead of tokens transmission in PDP techniques ; financial cost of fault tolerance and data replication. **Impact on application performance** Computational complexity and online latency of encryption/decryption by Data Loss Prevention techniques ; mean-time to mitigate and patch vulnerability ; mean-time to complete configuration changes.
Implementation
Dynamicity % Applications deployed with : enforcement of Multi-Factor authentication for access of privileged users and remote access ; enforcement of policies on password strength and expiration, and blocking of invalid login attempts ; techniques for protection of sensitive configurations from unauthorized access. **Flexibility** % Applications deployed with : implementation of risk-based entitlement decisions ; implementation of key management procedures and internal storage of encryption keys ; mechanisms for VMs interference prevention and instances isolation (against cross-VM attack via Side Channels) ; implementation of role-based access control at the hypervisor level. **Continuity** % Applications deployed with : events monitoring and audit by hypervisor ; implementation of security incident response plans ; configuration of SIEM incident reporting, analysis and alerting ;

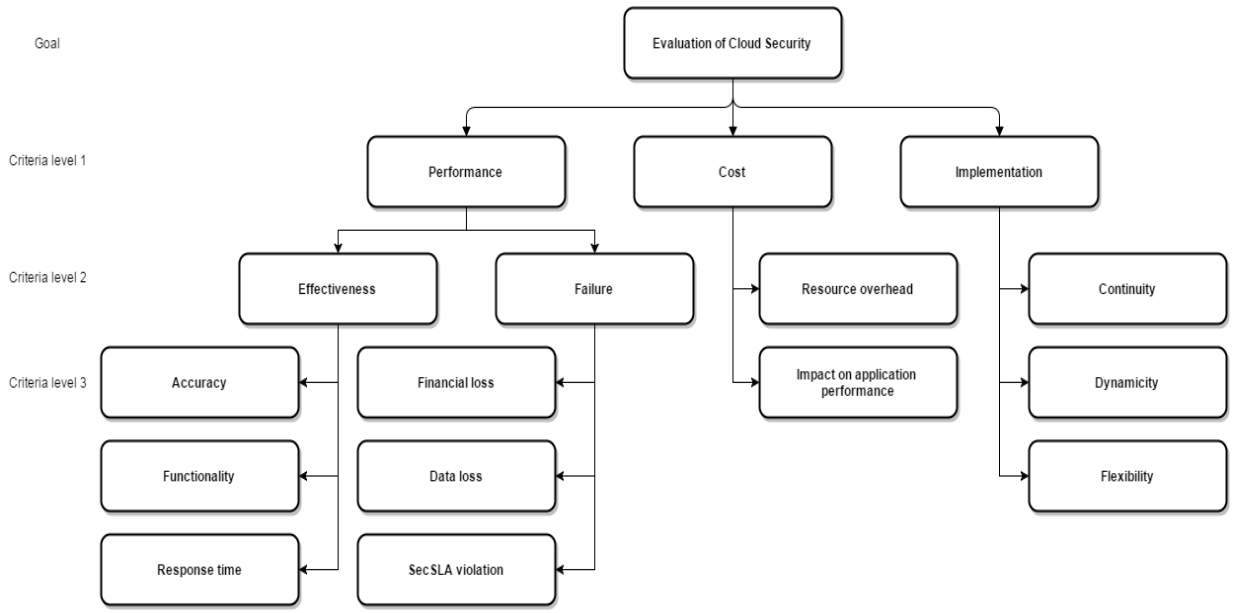


Figure A.1 The set of Cloud security evaluation criteria.

provide the necessary information about the metrics related to this criterion through the technical details and statistics concerning the deployment of their services.

A.3 The Evaluation and Selection Methodology based on MCA

The problem of evaluation and selection of Cloud security services is considered as an MCA [23] problem since it is based on multiple evaluation criteria structured in a hierarchical manner. We describe in the following the different steps of the proposed methodology.

A.3.1 Defining the MCA context

The objective of the evaluation of Cloud security services is twofold : first, it will help CSPs to better understand the security situation of their Cloud environments and compare the levels of their security services, and second, it will allow Cloud consumers to evaluate and select the security services that best match their requirements. The MCA problem can be seen as a complex problem with multiple and interdependent evaluation criteria. For this purpose, we decided to use the (AHP) method [23] which simplifies complex and ill-structured problems by arranging the evaluation factors in a hierarchical structure. At the top level is the goal which is Cloud security evaluation in our case and the lower levels correspond to the evaluation criteria and sub-criteria, as shown in figure A.1.

A.3.2 Subjective Weighting of Criteria

To perform the weighting of criteria during the selection process, we decided to use the subjective judgment of AHP to express the relative importance of criteria based on the preferences and requirements of Cloud consumers by constructing a Pair-wise Comparisons Matrix (judgment matrix) for each of the criteria at each level of the hierarchy. For a criterion with l sub-criteria, the matrix has the following form :

$$A = \begin{pmatrix} & & & \\ a_{11} & a_{12} & \dots & a_{1l} \\ & & & \\ a_{21} & a_{22} & \dots & a_{2l} \\ & & & \\ & \cdots\cdots\cdots & & \\ & & & \\ a_{l1} & a_{l2} & \dots & a_{ll} \\ & & & \end{pmatrix} \tag{A.1}$$

where each element $a_{ij},\ (i, j = 1, 2, \dots, l)$ of the matrix represents the relative significance of any two sub-criteria c_i and c_j under a specific criterion. A local weight vector W is then constructed for each criterion that does not belong to the bottom levels of the hierarchy by calculating the normalized eigenvector corresponding to the maximum eigenvalue λ_{max} of the judgment matrix with the following equation :

$$AW = \lambda_{max} W \tag{A.2}$$

A.3.3 Computing Security Values for Bottom Levels Criteria

The evaluation process based on the proposed hierarchy starts by evaluating the Cloud security services in the context of the bottom levels' criteria using the defined quantitative metrics. The steps at this stage are detailed in the following. Since we have three levels in the hierarchy, we use the indexation (o, p, q) to refer to a criterion node where $o,\ p$, and q are the indexes of the criteria on levels 1, 2, and 3 respectively.

A.3.3.1 Collection and Normalization of Metrics

We start by constructing a security matrix for each of the criteria on the bottom levels using the metrics developed in the previous section. This matrix contains the values of metrics for

each CSP under evaluation and has the following form

$$SM_{o,p,q} = \begin{pmatrix} sm_{11} & \cdots & sm_{1M} \\ \vdots & \ddots & \vdots \\ sm_{N1} & \cdots & sm_{NM} \end{pmatrix} \tag{A.3}$$

where M is the number of metrics that are used to measure the security in the context of the corresponding criterion, and N is the number of CSPs under evaluation. Every element $sm_{nm},\ (n = 1, 2, \ldots, N;\ m = 1, 2, \ldots, M)$ of the matrix represents the value of metric m provided by CSP n.

Definition : The ideal metric value could be the lowest or the highest among the values depending on the metric nature. A metric is called positive when a higher value of it indicates higher security level (e.g., attack detection rate), and negative when a higher value of it indicates lower security level (e.g., Cloud failure rate).

Qualitative metrics can also be quantified using special techniques or mapping functions. Then, normalization of metrics is performed in order to facilitate the evaluation process since metrics (collected raw data) have different units of measurement. Therefore, the values of metrics are transformed to be relative to the ideal metric value by applying the following formulas on the elements of the security matrix :

$$rsm_{nm} = \begin{cases} \frac{sm_{nm} - min_m}{max_m - min_m} & \text{if metric m is positive} \\ \frac{max_m - sm_{nm}}{max_m - min_m} & \text{if metric m is negative} \\ 1 & \text{if } max_m = min_m \end{cases} \tag{A.4}$$

where max_m and min_m are respectively the maximum and minimum values of metric m in the matrix SM, and $rsm_{nm},\ (n = 1, 2, \ldots, N;\ m = 1, 2, \ldots, M)$ represents the relative value of metric m for the CSP n.

Using this normalization process, a Relative Security Matrix (RSM) is constructed for each

criterion at the bottom level of the hierarchy. This matrix has the following form :

$$RSM_{o,p,q} = \begin{pmatrix} rsm_{11} & \cdots & rsm_{1M} \\ \vdots & \ddots & \vdots \\ rsm_{N1} & \cdots & rsm_{NM} \end{pmatrix} \tag{A.5}$$

A.3.3.2 Objective Weighting of Metrics

In order to express the relative significance of collected data to the evaluation process, we compute a specific weight for each of the metrics. These weights are called objective and are generated using the standard deviations of normalized data using the following formula :

$$w_m = \frac{\sigma_m}{\sum_{m=1}^{M} \sigma_m} \tag{A.6}$$

where w_m and σ_m are respectively the objective weight and standard deviation of metric m. This method reflects transparency and objectivity, and helps automating the evaluation process since it requires no human intervention and depends more on the collected data.

A.3.3.3 Computing Security Values

After computing the relative security matrix for each criterion at the bottom level of the hierarchy, we compute the Relative Security Vectors (RSV) by combining security matrices and weight vectors. The (RSV) of each criterion is obtained by the formula :

$$RSV_{o,p,q} = RSM_{o,p,q} \star W_{o,p,q} \tag{A.7}$$

where $W_{o,p,q}$ is the weight vector of the metrics under the criterion (o, p, q) and the vector $RSV_{o,p,q}$ contains the relative security values of CSPs with respect to criterion (o, p, q).

A.3.4 Computing Overall Security

After computing the relative security matrix of each criterion at the bottom levels of the hierarchy, we start evaluating security in the context of each criterion at the upper levels by combining relative security matrices and weight vectors. The RSVs of sub-criteria are combined as columns to form the RSM of the corresponding upper criterion. This procedure

of calculation using equation A.7 is repeated until the RSV of the goal (overall security value) is obtained [160].

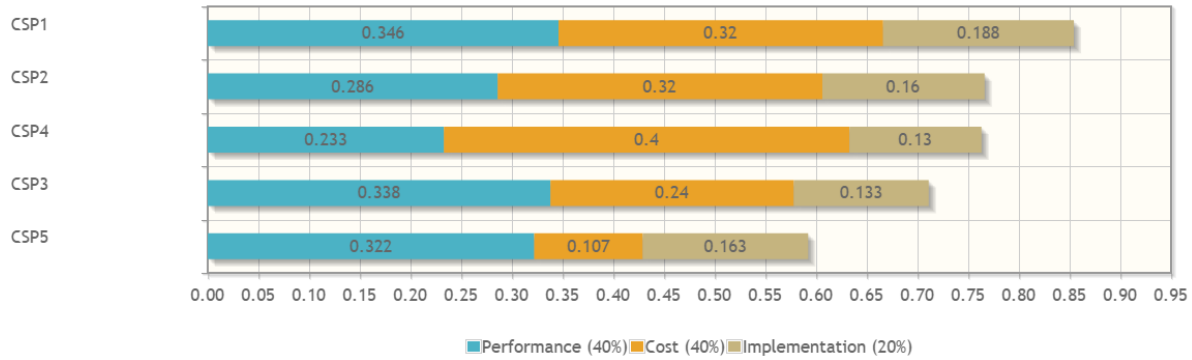
A.3.5 Sensitivity Analysis

Decision making involves managing trade-offs or compromises among a number of criteria that are in conflict with each other. The last step of the evaluation process involves conducting a sensitivity analysis to study the effects of evaluation data and weights of criteria on the overall evaluation results. First, Cloud service providers, consumers, and other key players can be consulted to ensure that the MCA evaluation process includes the criteria that are of concern to all stakeholders. Second, following the proposed methodology to examine how the ranking of CSPs might change under different values of metrics or weights [23] is necessary to determine the evaluation success.
We consider in this section a numerical example of an evaluation scenario in order to clearly explain our methodology. Five CSPs are being evaluated in this scenario and the values of evaluation metrics are provided. After passing through all the previous steps of the evaluation process (numerical data and computational steps are not shown here due to space limitation), we conduct a sensitivity analysis using an MCA program to show the importance of criteria weighting in the selection process. Figure A.2 shows how the ranking of CSPs and their evaluation scores change when using two different vectors of weights for the criteria. For instance, if the cost criterion is more important than the other criteria as in Figure A.2b, the consumer should choose CSP4 since she outperforms the other CSPs. As a result, this analysis helps Cloud consumers in determining the adequate trade-off during the decision making process. The value of the overall security/cost also plays a role at the end of the service selection process.

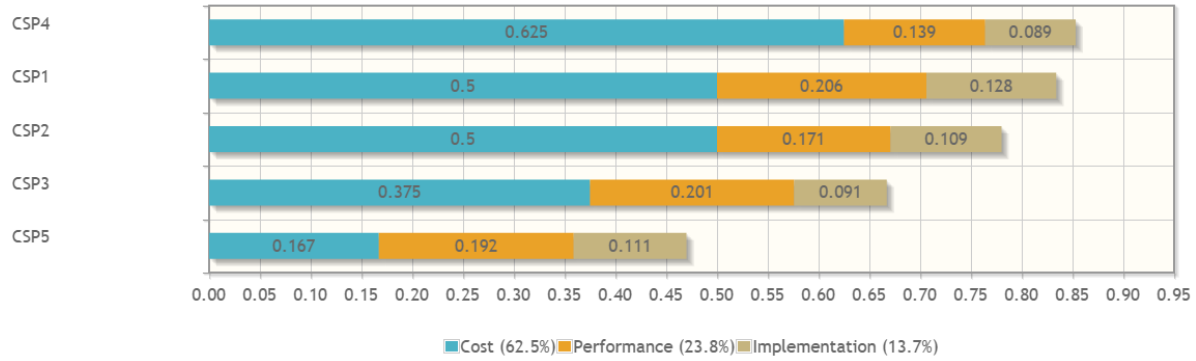
A.4 Architecture General Overview

In this section, we discuss in general the design of a Cloud security service evaluation and selection architecture based on the methodology proposed in the previous section. The essential components of this architecture with their roles and interactions are explained as follows :

— Service registration component : it collects the descriptions of offered security services and the necessary information about evaluation metrics from CSPs.
— CIAQ and security update component : it keeps track of updated information within the (CAIQ) [9] about CSPs' security information and communicates it with the security information database. It also tracks security information updates through se-



(a) Evaluation results using the first vector of weights.



(b) Evaluation results using the second vector of weights.

Figure A.2 Evaluation results in terms of weight vectors.

veral sources, such as experts' assessments, providers' statements, users' opinions, and security certificates.

— Security information database : it holds all the data collected from the above listed sources.

— Security evaluation and decision making component : it retrieves the required evaluation data from the database and applies the described MCA methodology.

— Security Manager and Recommender : it retrieves the information related to security requirements and preferences from the Cloud consumers and returns security recommendations and processed decisions.

— Evaluation Results Database : it holds the results of the evaluation process and communicates it with CSPs in the form of performance tables or diagrams.

Such architecture could be implemented onto the Cloud proxy as one of the Cloud discovery and description services to provide standard evaluation of security services.

A.5 Conclusion

Security is the main reason for which Cloud adoption is still slow. The existence of a successful security evaluation architecture in the Cloud will help consumers in assessing the security level of offered services before migrating their businesses to the Cloud. In this paper, we base the security evaluation process on a set of well-defined and structured criteria that could be evaluated using measurable metrics. The MCA-based evaluation methodology that we proposed takes into consideration both the nature of evaluation data and the preferences of the consumers. We also described the design elements of a Cloud security service evaluation and selection architecture. In our future work, we are planning to deploy a prototype implementation of this architecture to evaluate its effectiveness and usability.

APPENDIX B

ARTICLE 7 : A COOPERATIVE GAME FOR ONLINE CLOUD FEDERATION FORMATION BASED ON SECURITY RISK ASSESSMENT

Talal Halabi, Martine Bellaiche, and Adel Abusitta

IEEE 5th International Conference on Cyber Security and Cloud Computing (IEEE CSCloud 2018), pp. 83-88, 2018.

Abstract

Cloud federations allow Cloud Service Providers (CSPs) to deliver more efficient service performance by interconnecting their Cloud environments and sharing their resources. However, the security of the federated Cloud service could be compromised if the resources are shared with relatively insecure and unreliable CSPs. In this paper, we propose a Cloud federation formation model that considers the security risk levels of CSPs. We start by quantifying the security risk of CSPs according to well defined evaluation criteria related to security risk avoidance and mitigation, then we model the Cloud federation formation process as a hedonic coalitional game with a preference relation that is based on the security risk levels and reputations of CSPs. We propose a federation formation algorithm that enables CSPs to cooperate while considering the security risk introduced to their infrastructures, and refrain from cooperating with undesirable CSPs. According to the stability-based solution concepts that we use to evaluate the game, the model shows that CSPs will be able to form acceptable federations on the fly to service incoming resource provisioning requests whenever required.

B.1 Introduction

The adoption of Cloud Computing technology is on the rise. Many organizations are deciding to shift their businesses to the Cloud to benefit from the scalability, resilience, and cost reduction characteristics. Recent statistics [1] showed that the total worldwide Cloud IT infrastructure revenue has almost tripled in the last four years. However, the security risk introduced due to outsourcing customers' services and data to third parties still constitute an obstacle to the full migration to the Cloud. Cloud Computing involves many security

threats inherited from its architectural model and technical properties like virtualization, multi-tenancy, and data distribution. These threats include data breaches, data loss, and denial of service [70]. Security is one of the principal driving factors of the Cloud market, especially for businesses that deal with sensitive information. Thus, Cloud Service Providers (CSPs) are expected to maintain the security of the Cloud service and data and protect their availability.

Cloud Computing also introduces important challenges for CSPs, such as performance guarantee, resource limitation, disaster-recovery planning, regional distribution of workloads, and legal issues. To address these problems, the concept of Cloud federation was born. It allows a CSP to flexibly and transparently outsource a portion of its users' requests to other independent CSPs, especially when the available amounts of resources can not cope with the dynamic nature of the Cloud workload and the variability in users' requests for data and computing-intensive applications. The typical federation model between two CSPs is depicted in Figure B.1. The federation can exist for different service delivery models : Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). It usually occurs along two different dimensions : horizontal, taking place at matching layers of the Cloud Stack (e.g., under the form of shared Virtual Machine (VM) instances), and vertical, spanning multiple layers in order to service the additional requests on one specific layer through delegation [5]. By interconnecting their Cloud infrastructures and optimizing the management of their workload, federated CSPs maintain higher performance and Quality of Service (QoS) levels, and improve cost-effectiveness and energy efficiency [5].

In order to form a Cloud federation and allow CSPs to find their potential federation candidates, the state-of-the-art has proposed several models, mainely based on profit maximization, trustworthiness, or the optimization of QoS parameters. For instance, Li et al. [83] proposed an algorithm for VMs trading in a Cloud federation using an auction-based scheduling mechanism that maximizes the profit of the federation members. Samaan [84] designed a resource sharing strategy in a federation that increases the revenue based on game theory. In [85],

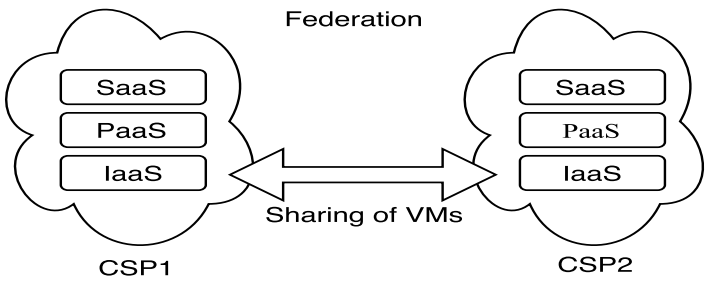


Figure B.1 Federation model between two CSPs.

Mashayekhy et al. introduced a federation formation game that allows CSPs to maximize their profit. Guazzone et al. [86] used the cooperative game theory to develop an algorithm that allows the formation of federations while maximizing the profit of CSPs and reducing the energy cost. Hassan et al. [87] proposed a federation formation mechanism using a trust-based cooperative game theory that allows CSPs to maximize their profit and minimize the Service Level Agreement (SLA) penalty cost by federating with trustworthy and reliable CSPs. In [88], Abdel Wahab et al. also proposed a trust-based hedonic coalitional game that permits the formation of multi-cloud communities of trustworthy services.

The security risk that a CSP can introduce to the federation when joining its members was not considered in the previous federation formation models. For instance, cooperating with a federation that maximizes profit and reduces cost could be very beneficial to CSPs, but does not guarantee service safety. Forming a federation with CSPs that have high security risk levels could increase the probability of security breaches and failures, and hurt the reputation of the federation members by increasing the rate of Security-SLA violations, which is the official contract that governs security management between providers and customers [150]. Therefore, to guarantee security satisfaction and limit the rate of security violations, the security risk level of the federation members should be efficiently assessed before the formation of the federation. This factor could be eventually combined with multiple QoS parameters (i.e., response time and availability) to provide more efficient and secure federations.

The formation of Cloud federations represents a decision making situation that can be mathematically modeled as a combinatorial optimization problem. Cooperative game theory plays a major role in this area, especially when multiple decision makers are involved. It provides effective high-level approaches to describe the strategies and payoffs of the players. Game theory has been applied to solve different kinds of optimization and allocation problems in the areas of wireless networking, web servicing, and computer system security. In this paper, we look at the Cloud federation formation problem from a security risk perspective. To the best of our knowledge, this is the first work to combine federation formation with security risk assessment in the Cloud. Our contributions can be described as follows :

— First, we propose a security risk assessment approach with which we quantitatively evaluate the security risk levels of CSPs according to several criteria.

— Second, taking into account the computed security risk levels, we propose a Cloud federation formation algorithm based on a hedonic coalitional game with a security risk preference relation that satisfies the stability property [105], that is, none of the CSPs in the final partition of federations has incentive to leave its current federation to join another one that is more desirable.

The properties of the federation formation game are then analyzed, and its performance is

evaluated. The remainder of the paper is structured as follows. Section B.2 describes the security risk assessment approach. Section B.3 describes the security risk-based federation formation model. In Section B.4, the performance of the model is evaluated. Finally, section B.5 concludes the paper.

B.2 Cloud Security Risk Assessment

The research on risk assessment in Cloud Computing is continuously evolving, and authors are trying to find efficient solutions that could accurately estimate the Cloud security risk. For instance, Chopra et al. [44] studied the risk of migrating data and applications to the Cloud and performed a qualitative analysis of the associated risks prior to, during, and after the migration process. Tang et al. [45] modeled the Cloud risk identification phase using the Hierarchical Holographic Modeling framework and used the fuzzy set theory to compute the probability of security risk. Ben Aissa et al. [47] proposed a model that estimates system security by quantifying the costs to stakeholders using the mean failure costs metric. The challenge to their model is the difficulty in accurately quantifying the defined stakes, dependability, impact, and threat matrices. Sendi and Cheriet [48] proposed a quantitative iterative approach to evaluate the security risk associated with the Cloud platform using a fuzzy multi-criteria decision making technique. Finally, Djemame et al. [149] developed a risk assessment framework that evaluates the risk during Cloud service deployment and in run-time, with more focus on performance and QoS than security.
In this paper, we propose an approach to quantify the security risk level of a Cloud infrastructure based on a set of evaluation criteria that describe the implementation of the required security measures and procedures to provide two aspects of security risk management in the Cloud : Security Risk Avoidance (SRA) and Security Risk Mitigation (SRM). In an IT infrastructure, security risk is normally defined as a function of threats' occurrence probability,

Table B.1 Some common security threats to Cloud Computing.

Threat	Vulnerability	Risk source	Damaged assets
Cross-VM attack via Side Channels	VM co-residence	Neighboring VM	Data
Malicious SysAdmin	Loss of physical control over data	Malicious insider	Data
Data loss or manipulation	Loss of physical control over data	Cloud server, server administrator	Data
Dishonest computation in remote servers	Outsourced computation	Cloud server	Computations
Direct and indirect DoS attacks	VM co-residence, bandwidth under-provisioning	Malicious user, neighboring VM	Service, Data
Economic Denial of Sustainability (EDoS)	Cloud pricing model	Malicious user	Service

existing vulnerabilities, and the impact of a security incident on the data or service. In a Cloud system, it could be present on the VM level, the physical host level, and the whole infrastructure level. Table B.1 presents some common threats to Cloud Computing with their associated damages to the Cloud service and data. For instance, on the network layer, the botnets that form with the aim of exploiting the vulnerabilities of the Cloud infrastructure, ARP spoofing, and Distributed Denial of Service (DDoS), are the most feared threats by the CSPs. The virtualization layer also presents serious threats to the Cloud infrastructure. VMs could be compromised during creation, execution, replication, and migration. These threats are described in more detail in [156].

The security risk evaluation criteria were developed according to the set of security evaluation metrics proposed in [129], and are presented in Table B.2. Their principal objective is to assess how and how much a CSP's infrastructure is capable of protecting the following three security attributes defined by the CIA triad security model :

— Confidentiality, which concerns with protecting customers' sensitive information from unauthorized disclosure. Unlike traditional in house applications, Cloud-based applications require the implementation of more sophisticated security solutions to protect users' confidentiality since they are running on untrusted servers and networks. To protect data and service confidentiality, CSPs usually deploy Identity and Access Management (IAM) solutions and appropriate virtualization and network security, in addition to physically securing their sites and facilities.

— Integrity, which can be defined as the accuracy and validity of data and computations in regards with customers and business process' expectations. CSPs usually protect the data at rest, in transit, and during process using adequate cryptography and leakage prevention techniques.

— Availability, which describes the ability of the service to be accessible whenever required by the users. This attribute is usually maintained by the deployment of efficient backup and recovery procedures, incident response plans, and DDoS mitigation solutions.

In our model, security risk assessment is controlled by a central unit (e.g., a Cloud broker) which collects all the required information about the evaluation criteria and quantitatively estimates a security risk level for the assessed Cloud infrastructure. We assume the implicit cooperation and transparency of CSPs when it comes to providing these information since forming desirable federations falls within their interest. Standardizing the evaluation criteria will help in automating the security risk assessment process and providing a fair evaluation of the involved CSPs. We consider a set $\mathcal{N} = \{n \mid 1 \leq n \leq N\}$ of N CSPs to be assessed, and a number K^A of SRA evaluation criteria and a number K^M of SRM evaluation criteria.

Table B.2 A set of Cloud security risk evaluation criteria.

	Evaluation criteria
SRA	- Activation of Multi-Factor authentication
	- Enforcement of policies on password strength
	- Implementation of risk-based access entitlement
	- Configuration of security groups
	- Configuration of Access Control Lists on virtual ports
	- Enabling of client certificate for SSL/TLS
	- Configuration of web application scanners
	- Provisioning of dynamic malware analysis and detection
	- Capability of open encryption methodologies
	- Database deployment with SSL protected transactions
	- Support of secure data deletion
	- Implementation of data loss/leakage prevention techniques
	- Use of data dispersion techniques for storage
	- Support of data isolation
	- Providing VM image scanning and filtering before creation
	- Implementation of Secure VM and data migration
	- Implementation of Hypervisor-level access control
	- Capability of encryption of virtual storage
	- Deployment of data destruction procedures after migration
	- Implementation of PDP techniques [121] for data validation
	- Implementation of computational integrity verification [122]
	- Deployment of applications with computing replication
SRM	- Configuration of IDS/IPS
	- Deployment of DDoS mitigation solutions
	- Providing VM backup, restoration and clean-up capabilities
	- Enabling events monitoring and auditing by hypervisor
	- Providing data replication between Cloud nodes
	- Providing Multi-failure disaster recovery capability
	- Configuration of SIEM [126] incident reporting and alerting

The security risk level SR_n of a CSP n is computed as follows :

$$SR_n = \alpha(1 - \frac{Sat_n^A}{K^A}) + \beta(1 - \frac{Sat_n^M}{K^M}) \quad \forall n \in \mathcal{N} \tag{B.1}$$

where Sat_n^A and Sat_n^M are respectively the numbers of criteria that n satisfies with respect to SRA and SRM, and α and β are the weights that the broker assigns to SRA and SRM respectively during the assessment process ($\alpha+\beta = 1$). Hence, the security risk level is always in the interval [0,1], and is directly related to the implementation of security capabilities represented in the evaluation criteria of Table B.2. Increasing the level of integration of these security measures into the Cloud infrastructure will lead to the reduction of threat vector, and eventually presenting a lower security risk level. This security risk assessment approach will allow to perform a relative evaluation of CSPs' security risk according to a reference level, which is required in the context of our cooperation model. Next, the security-risk based federation formation game is presented.

B.3 The Security Risk-based Cloud Federation Formation Game

Coalition formation is a major subject in multi-agent systems, and hedonic games form a popular category of the coalitional cooperative games, in which profit allocation among the coalition members is not the main problem. In a hedonic game, the players are usually self-interested, and the stability property is guaranteed, that is, when the final partition of coalitions is formed, none of the players will have an incentive to leave its current coalition to join another. In the case of Cloud federation formation, the number of possible coalition structures is too large to permit an exhaustive search for the optimal solution and finding it is a NP-complete problem [104]. Thus, a hedonic game was adopted to solve the problem.

B.3.1 Game Model

In the proposed game, the players are the CSPs and the coalitions to be formed are the Cloud federations. The objective of the CSPs is to join federations with desirable members according to the security risk levels and reputations, and each CSP acts as a selfish player when deciding to prefer a federation over another. We start by formally defining the federation formation concept from the perspective of coalitional games.

Definition 9. *Federation partition. A federation structure or partition is a set of M federations $\Pi = \{F_m,\ m \in \mathcal{N}^*,\ m \leq M\}$, where each $F_m \subseteq \mathcal{N}$ is a disjoint federation such as $\bigcup_{m=1}^{M} F_m = \mathcal{N}$ and $F_m \cap F_y = \emptyset\ \forall\ y \neq m$.*

When forming a federation, we assume that the CSPs are already fulfilling the constraints on workload servicing, that is, they are able to provide sufficient resource capacities, and we look beyond this criterion to verify the security risk status of the members. The grand federation G is formed when all CSPs decide to cooperate in one single federation, that is, $G = \mathcal{N} = \{1, \ldots, N\}$. We consider our game to be non-cohesive [144] which goal is to generate a set of disjoint federations that group the CSPs according to their security risk levels, since forming the grand federation might not provide the optimal security risk level for its members. The game is also considered to be of Non-Transferable Utility (NTU) since security risk can not be distributed among the federation members. The definition of a hedonic game is given by the following :

Definition 10. *Hedonic game. A hedonic game is a type of NTU coalitional games where the utility of a player in a coalition depends only on its members, and the formation of coalitions is based on the preferences that the players have over the set of possible coalitions [105].*

We consider our game to be hedonic since it verifies these two conditions. We define the function $v_n : \mathcal{N} \rightarrow \mathcal{R}$ as the valuation function of a CSP n, which assigns a real value to each of the other CSPs in $\mathcal{N}$, as follows :

$$v_n(n') = \frac{1}{SR_{n'}(1 - rep_{nn'})} \quad \forall n, n' \in \mathcal{N} \tag{B.2}$$

where $rep_{nn'}$ denotes the security reputation of a CSP $n' \in \mathcal{N}$ normalized to the interval]0,1[according to CSP n, and is computed based on the previous interactions between the two CSPs and the ratings provided by the customers. This parameter is integrated in the model to compensate for the possible lack of integrity of the CSP when providing the information related to the security risk evaluation criteria. The ratings reflect how much the CSP was honest about satisfying customers' security requirements in the past, and are regularly updated. The Collaborative Filtering approach, which is widely used in the development of recommender systems, could be employed in our case to exploit the Cloud customers' ratings if an appropriate platform for security satisfaction evaluation is created. In the case where no ratings or only a small number of ratings are available for a particular CSP during the evaluation process, approaches based on the baseline models or deep learning techniques could be adopted to solve the incomplete information problem [161].
The valuation function implies that a CSP will highly valuate another if the security risk level of the latter is low and its security reputation is high, with the assumption of $v_n(n) = 0$ [162]. This function can be extended to valuate the different possible federations that could form as follows :

$$v_n(F_m) = \frac{\sum\limits_{n' \in F_m} v_n(n')}{|F_m|} \quad \forall n \in \mathcal{N},\ \forall F_m \in \Pi \tag{B.3}$$

where $|F_m|$ is the cardinality of the federation. This implies that a CSP valuates a particular federation based on its valuations of its members. The Cloud federation formation game proposed in this paper is defined as follows.

Definition 11. *The security risk-based Cloud federation formation game. It is the pair* $(\mathcal{N}, \succeq)$*, where* $\mathcal{N}$ *is the set of CSPs in the game, and* $\succeq_n$ *is a reflexive, complete, and transitive preference relation on the set of all federations that* $n \in \mathcal{N}$ *can form based on the security risk level.*

Based on this definition, for all CSPs $n \in \mathcal{N}$ and for all $E, E' \in \Pi$, we define $\succeq_n$ as follows :

$$E \succeq_n E' \ \Leftrightarrow \ v_n(E) \geq v_n(E') \tag{B.4}$$

This game belongs to the family of *Fractional Hedonic Games (FHG)* [162]. $E \succeq_n E'$ indicates that n prefers to be a member of federation E than to be a member of federation E', or at least prefers them both equally. Based on the preference relation, each CSP will compare the set of possible federations and indicate its intention to be a part of one of them, which means that it will prefer the federation that valuates the most. Note that the individual rationality of the players is assumed, that is, every CSP prefers to form a federation instead of acting alone non-cooperatively, since the goal here is to form a federation and respond to a resource provisioning request.

B.3.2 The Federation Formation Algorithm

The security risk-based federation formation algorithm is presented in Algorithm 5. It is executed by the broker in case a resource provisioning request that requires a federation between multiple CSPs to be formed is received. The algorithm takes as inputs the initial federation partition Π_i and CSPs' security risk levels computed using Equation B.1 and security reputation values, and outputs the final federation partition Π_f. For any CSP $n \in \mathcal{N}$, we denote by $F_\Pi(n)$ the federation $F_m \in \Pi$ such that $n \in F_m$. Given a current federation partition, a CSP faces three options : 1) stay in the current federation ; 2) merge with any of the other federations ; or 3) split from the current federation.

The algorithm makes use of the two following comparison rules which are based on the defined preference relation to permit to decide if a federation is more preferred than the other :

— **Merge rule** : this rule is based on the hedonic shift rule defined in [145], which is selfishly executed by each CSP when moving between federations. A CSP n will decide to merge with a federation $F_m \in \Pi_i \cup \emptyset$ and form a new federation if, and only if, $F_m \cup \{n\} \succeq_n F_\Pi(n)$, that is, if the valuation of the new federation by n is higher than its valuation of its current federation.

— **Split rule** : a federation $F_z \in \Pi_i$ will decide to split into two federations F_x and F_y such that $F_z = F_x \cup F_y$ if $\exists n \in F_x$ such that $F_x \succeq_n F_z$ or $\exists n \in F_y$ such that $F_y \succeq_n F_z$.

In other words, the merge rule is applied when a CSP decides that joining a new federation is more preferred than staying in the current one. On the other hand, the split rule is applied on a federation if at least one of its sub-federations will have higher valuation by at least one of its members. An iterative sequence of the merge and split rules will continue to run according to the improvement in the valuations of CSPs until the partition converges to its final and stable form Π_f. The split rule is essential in this case to avoid early convergence of the federation formation algorithm to a solution that is far from the optimal one [85]. Note that all the visited federations that have been later split were conserved by the algorithm in

Algorithm 5 Pseudo-code of the federation formation algorithm.

Input:
- The initial federation partition $\Pi_i = \{F_1, \ldots, F_M\}$
- The vector $SR = (SR_1, \ldots, SR_N)$ of CSPs' security risk levels
- The vector $rep_n = (rep_{n1}, \ldots, rep_{nN})$ of CSPs' security reputation values $\forall n \in \mathcal{N}$

Output:
- The final federation partition Π_f

1: **repeat**
2: **for all** CSP $n \in \mathcal{N}$ **do**
3: Apply merge rule and update federations in Π_i
4: **end for**
5: **for all** non-singleton $F_m \in \Pi_i$ **do**
6: Apply split rule and update federations in Π_i
7: Store the visited federations in a history set
8: **end for**
9: **until** no split occurs
10: **return** Π_f

a history set to avoid visiting them again by the players.

During the merge operations, the highest complexity occurs when all CSPs in Π_i are being non-cooperative, that is, acting alone in separate federations. In this case, every CSP will try to merge with all the others. The first CSP will at most perform $(N-1)$ attempts to merge, the second will perform $(N-2)$ merge attempts at most, and so on. So the total number of possible merge attempts in the first run would be $N(N-1)/2$ in the worst case. Since in practice, and after the first run, CSPs may continue to work as a federation, the future merge operations will have to deal with less than N federations in the partition. On the other hand, the complexity of the split operation is determined by the k^{th} Bell number where k is the dimension of the federation, which could be computationally affordable in our case since the split process is performed on federations of relatively small sizes.

B.3.3 Game Analysis

In this section, the properties of the proposed security risk-based federation formation game are analyzed. More specifically, we demonstrate the property of convergence of the federation formation algorithm to a final solution and the property of stability of the produced final solution.

Definition 12. *Nash-Stability. A partition of federations* Π *is considered Nash-stable if no CSP has an incentive to move from its current federation* $F_\Pi(n)$ *to join a different one, nor to act alone. That is, for* $\Pi = \{F_m,\ m \in \mathcal{N}^*,\ m \leq M\}$, $\forall\ n \in \mathcal{N},\ F_\Pi(n) \succeq_n F_m \cup \{n\}$ $\forall\ F_m \in \Pi \cup \emptyset$.

Definition 13. *Individual Stability. A partition of federations* Π *is considered individually stable if no CSP can benefit by moving from its current federation* $F_\Pi(n)$ *to another one without negatively affecting the members of the latter. That is,* $\nexists n \in \mathcal{N}$ *and* $F_m \in \Pi \cup \emptyset$ *such that* $F_m \cup \{n\} \succ_n F_\Pi(n)$ *and* $F_m \cup \{n\} \succeq_{n'} F_m$, $\forall\ n' \in F_m$.

The following propositions are made with regard to the proposed federation formation game, and their proofs are inspired by the work in [86] and [87].

Proposition 4. *Convergence. Starting from any initial partition of federations* Π_i*, Algorithm 5 converges to a final partition* Π_f *consisting of a number of disjoint federations.*

Proof. Every application of the merge and split rules will transform the current partition into a new partition that has not been visited in the past, until reaching the final partition Π_f. Since the number of transformations is finite, and at most, is equal to the number of partitions defined by the N^{th} Bell number B_n for N CSPs, the sequence of transformations will always be limited and will converge to a final partition. □

Proposition 5. *Nash-Stability. Any final partition* Π_f *produced by Algorithm 5 is a Nash-stable federation partition.*

Proof. This proposition can be proved by contradiction. Let us assume that the final partition of federations Π_f is not Nash-stable. Then, there exists a CSP n that prefers to leave its current federation $\Pi_f(n)$ and join another one F_m, that is, $F_m \cup \{n\} \succ_n \Pi_f(n)$. Therefore, n will perform the split rule and the final partition will change to a new one Π'_f such as $\Pi'_f \neq \Pi_f$ which contradicts with Proposition 1 that states that the partition Π_f is the final outcome of the federation formation algorithm. Hence, we conclude that the algorithm always converges to a Nash-stable federation partition. □
According to [105], this implies that Algorithm 5 also converges to an individually stable partition of federations.

B.4 Performance Evaluation

Since the information needed to perform a security risk assessment of CSPs is not yet fully available due to lack of transparency, we randomly generate our data to evaluate the performance of the model. Each CSP n is assigned a security risk value SR_n between 0.2 and

Table B.3 An example of CSPs' valuations.

CSPs	1	2	3	4	5
1	0	12.35	8.49	8.5	7.75
2	26.32	0	4.68	6.61	13.68
3	5.48	8.82	0	3.9	12.92
4	4.46	8.42	7.87	0	6.64
5	7.52	6.73	6.09	7.22	0

0.8, and a vector rep_n of security reputation values between 0.3 and 0.9, which evaluates its interactions with other CSPs. The model was implemented in MATLAB and the test was performed on a 64-bit Windows 7 machine equipped with an Intel Core i7-3612QM CPU @ 2 :10 GHz Processor and 12 GB RAM.

First, we demonstrate the individual stability of the game through an empirical example. We start with an initial partition where five CSPs are acting alone, that is, Π_i={{1},{2},{3},{4},{5}}, and we run the algorithm of the federation formation game until reaching the final partition, which in our case was Π_f={{1,2,5},{3,4}}. The valuations of CSPs with respect to each other are given in Table B.3. Lets denote by F_1 and F_2 the two formed federations respectively. Its is noticed that none of the CSPs 1, 2, and 5 has incentive to leave its current federation F_1 and join F_2 since they all valuate F_1 more than F_2 according to Equation B.3. On the other hand, CSPs 3 and 4 valuate F_1 more than F_2, that is, $v_3(F_1 \cup \{3\}) > v_3(F2)$, and $v_4(F_1 \cup \{4\}) > v_4(F2)$, but both can not benefit from moving from F_2 and joining F_1 without negatively affecting the valuation of at least one of F_2's members, which in our case is CSP 2, since $v_2(F_1) > v_2(F_1 \cup \{3\})$ and $v_2(F_1) > v_2(F_1 \cup \{4\})$. Thus, the individual stability in the final partition is achieved.

Figure B.2 shows the average execution time of Algorithm 5 along with the standard deviation of the instances. The figure shows that the execution time could be affordable in a real-time environment where CSPs need to federate on the fly to respond to a service deployment request and automatically provision the resources. For instance, when $N = 8$, performing an exhaustive search for the optimal federation partition would mean to check all the possible federation partitions that could form, which is equal to the 8^{th} Bell number. Our model was checking only 53 different partitions on average before finding the final stable partition. Note that the average size of formed federations in the final partitions during the simulation was always between 2 and 3, which is a good indicator since the goal of the model is to generate more disjoint federations of smaller size, where CSPs feel at ease being among

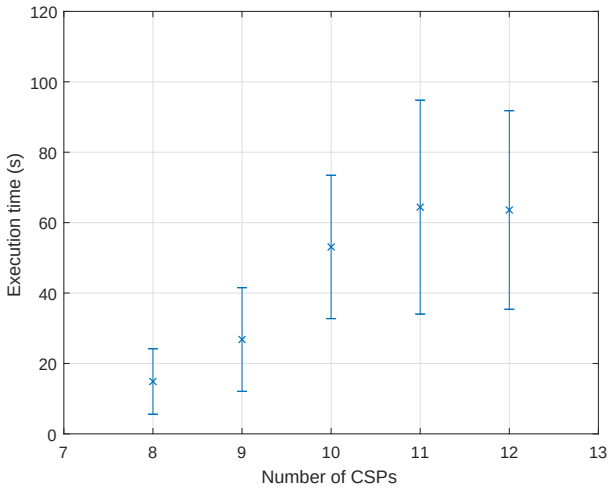


Figure B.2 The average execution time of the game (fifty runs).

the members, instead of forming the grand federation that could be undesirable for some of them.

B.5 Conclusion

In this paper, the Cloud federation formation problem was addressed from a new perspective. First, the security risk level within a Cloud infrastructure was quantitatively assessed according to several evaluation criteria, which could be standardized to provide a reference level for CSPs' security risk assessment. Then, a fractional hedonic coalitional game was used to model the federation formation process according to a security risk preference relation. Results showed that the model enables CSPs to form federations in which they enjoy the presence of each other, and refrain from joining undesirable federations where they either do not trust their members, or they feel that their security risk levels are unacceptable. We are currently in the process of collecting real information about CSPs' security risk levels to complete the evaluation of the performance of the proposed model based on a real data set. In the future, we are planning to integrate, along with security risk, other significant factors that could influence the Cloud federation formation process (e.g., costs, QoS), and study the trade-off that could exist between these factors.

APPENDIX C

ARTICLE 8 : ONLINE ALLOCATION OF CLOUD RESOURCES BASED ON SECURITY SATISFACTION

Talal Halabi, Martine Bellaiche, and Adel Abusitta

Accepted for presentation at *the 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (IEEE TrustCom-18)*, New York, USA, August 2018

Abstract

Businesses are becoming increasingly interested in exploiting the Cloud Computing technology. However, Cloud insecurity is still among the main factors that are blocking the full migration towards this paradigm. Increasing security investments will speed up the Cloud adoption process and improve the trustworthiness of the Cloud Service Providers (CSP). Moreover, the integration of the security element into the process of resource allocation will help increase the protection of the deployed services. However, this integration requires suitable modeling of customers' security requirements and CSPs' security offerings. To this end, we propose in this paper a broker-based model for the allocation of resources in the Cloud based on service security satisfaction. The resource allocation problem is modeled as a linear optimization problem and solved using an Evolutionary Computation approach, namely, the Genetic Algorithm (GA). The objective is to maximize the global security satisfaction of users' services by placing them on the data centers that adhere the most to their security requirements. Results show that the GA achieves an acceptable approximation of the optimal solution and is computationally efficient, which makes it suitable to function in online mode and cope with the scalability of the Cloud environment.

C.1 Introduction

The adoption of Cloud Computing is on the rise. Many organizations are deciding to shift their businesses to the Cloud to benefit from the scalability, resilience, and cost reduction characteristics. Recent statistics [1] show that the total worldwide Cloud IT infrastructure

revenue has almost tripled in the last four years, while the traditional IT infrastructure revenue continues to decline. However, the introduced security risk due to outsourcing customers' services and data to third parties poses an obstacle to the full migration to the Cloud. Cloud Computing involves many security threats inherited from its architectural model and technical properties like virtualization, multi-tenancy, and data distribution. These threats include data breaches, data loss, and denial of service [70]. Every year, the Cloud market giants witness serious security incidents and breaches causing outages and failures, which affects the production process and results in lost data and revenue [12].

Security is one of the principal driving factors of the Cloud market, especially for businesses that deal with sensitive information. Cloud Service Providers (CSPs) are expected to maintain the security of the Cloud service and data and protect their availability. According to the study in [11], the rate of organizations that completely trust the public Cloud infrastructures today to protect their data is only at 23%. This is due to the fact that CSPs are not investing enough resources and effort in deploying security solutions. To promote Cloud adoption, increasing the security and reliability levels of the Cloud infrastructures becomes indispensable, if not a priority. CSPs should try to offer advanced security features from which users could select to implement during application deployment according to their security requirements. Moreover, standardizing those features will help automating the service selection process and developing an appropriate pricing scheme with respect to the offered security levels, which will help CSPs in reducing the cost of security investment.

In this paper, we look at the Cloud resource allocation problem from a security perspective and beyond traditional factors such as performance, energy consumption, and load balancing. In this case, the resource allocation optimization problem would be based on evaluating the security satisfaction of users' requests and assigning their services to the Cloud infrastructures that respond the most to their security requirements to avoid future violations of Security Service Level Agreements (Security-SLA) and improve CSPs' reputations. The Security-SLA is the official contract that governs security management between the CSPs and customers and helps in guarantying their respective rights. It defines all the security elements that should be protected and the security policies that need to be implemented [150]. If adequate security measures were provided before the occurrence of an incident, violations of this agreement and what they entail in terms of financial and technical penalties could be reduced.

The resource allocation architecture that we propose is broker-based and aims at performing an online security-based allocation of users' requests on multiple Cloud infrastructures. These infrastructures could belong to different CSPs acting independently or within a federation, or could consist of data centers owned by the same CSP and offering different security confi-

gurations. The architecture can also be adapted to a Multi-Cloud setting, in which users can achieve optimal security satisfaction by taking advantage of each CSP's security capabilities through a mix and match service delivery model according to their requirements. Our contributions are :

— First, we develop a set of security features that could be offered in the Cloud in a standard fashion, and then model the problem of resource allocation in the Cloud based on security satisfaction.

— Second, we propose to solve the allocation problem using a heuristic approach based on the Genetic Algorithm (GA).

— Third, we implement the proposed algorithm and evaluate its performance. The algorithm achieved a good approximation of the optimal solution and demonstrated the ability to function in online mode.

The paper is organized as follows. Section C.2 discusses related work. Section C.3 describes the proposed architecture and security features and defines the security-based resource allocation problem. Section C.4 presents the proposed heuristic solution. In section C.5, simulations are performed and results are analyzed. Finally, section C.6 concludes the paper.

C.2 Related Work

We propose in this paper a model for online allocation of resources in Cloud Computing while considering users' security satisfaction. The model is principally based on the definition of qualitative security features inspired by our work on Cloud security evaluation in [129] and [130]. Our previous security-based service assignment model in federated Cloud environments [163] had the limitation of functioning in an offline mode only. The allocation of security resources in Cloud Computing has been studied in previous research, but in a very abstract fashion and not extensively. Liu and Lee [92] proposed a resource allocation algorithm for mobile Cloud Computing systems while providing a security guarantee. In their security-aware allocation model, security implementation is supplied by an extra number of Virtual Machines (VM) according to customers' security requirements. Liang et al. [93] proposed a Security Service Admission Model (SSAM) based on a Semi-Markov Decision Process to model the system reward for the CSP while allocating the security requests of the customers also in a mobile Cloud Computing setting. They divided the provided security services into two categories : normal security services which provide basic security mechanisms, and critical security services that provide more advanced security features.

Cloud profit maximization is often an essential factor when it comes to resource allocation. For instance, Goudarzi and Pedram [94] proposed a distributed solution to a SLA-based

resource allocation problem, which maximizes the total profit in the system while considering the following three dimensions in the optimization : the processing, data storage, and communication bandwidth. Also, Nezarat and Dastghaibyfard [95] proposed a game theoretical model to maximize profit in a Cloud environment. In their model, a combinatorial auction mechanism is used to select the winners among the competent users. Other factors such as energy consumption have also been a subject of attention when performing resource allocation in the Cloud. For example, Beloglazov et al. [152] proposed several energy-aware allocation algorithms which ensure efficient energy management within the Cloud's data centers while respecting the Quality of Service (QoS) constraints. Other research applied the GA to solve the energy-aware resource allocation problem. Quang-Hung et al. [153] proposed a GA for a power-aware allocation of VMs. Portaluri et al. [154] also proposed a power-efficient tasks allocation model based on the GA. Finally, Sharma and Guddeti [155] developed an improved GA to solve the multi-objective resource allocation problem in a green Cloud by creating enhanced initial solutions.

C.3 Problem Definition

In this paper, the problem consists on allocating users' service requests to multiple Cloud infrastructures via a third party (e.g., a Cloud broker) according to their security needs. To model the problem in the most general fashion, we assume that these infrastructures are owned by different CSPs. The goal of the broker at a time t is to perform an allocation that satisfies the most of users' security requirements. These requirements are expressed by a selection of specific security features that we develop based on our work in [129]. These features can be considered as qualitative binary metrics that reflect the security capabilities offered by a Cloud system for the different service delivery models : Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). While tailoring these features after a deep security evaluation in a Cloud environment, the principal objective was to protect the three security attributes defined by the CIA triad security model : Confidentiality (C), which concerns with protecting users' sensitive information and confidential data from unauthorized disclosure ; Integrity (I), which consists on protecting the accuracy and validity of data and computations ; and Availability (A), which describes the ability of the service and data to be accessible whenever required by the users. The proposed security features along with the security attributes which they aim at protecting are presented in Table C.1.

The broker-based Cloud resource allocation architecture that we propose based on security satisfaction evaluation is shown in Figure C.1. The broker receives a set of requests denoted

Table C.1 Some of the proposed Cloud security features that could be offered by CSPs.

Security features	Attributes
- Activation of Multi-Factor authentication	C
- Enforcement of policies on password strength	C
- Implementation of risk-based access entitlement	C
- Configuration of security groups	C
- Configuration of Access Control Lists on virtual ports	C
- Configuration of IDS/IPS	C, I, and A
- Deployment of DDoS mitigation solutions	A
- Enabling of client certificate for SSL/TLS	C and I
- Capability of open encryption methodologies	C and I
- Database deployment with SSL protected transactions	C and I
- Support of secure data deletion	C
- Implementation of data loss/leakage prevention techniques	C and A
- Use of data dispersion techniques	A
- Implementation of Secure VM and data migration	C and I
- Implementation of Hypervisor-level access control	C and I
- Enabling events monitoring and auditing by hypervisor	C and I
- Capability of encryption of virtual storage	C and I
- Deployment of data destruction procedures after migration	C
- Providing VM backup, restoration and clean-up capabilities	A
- Implementation of PDP techniques [121] for data validation	I
- Implementation of computing integrity checking techniques	I
- Deployment of applications with computing replication	I and A
- Providing data replication between Cloud nodes	A
- Providing Multi-failure disaster recovery capability	A
- Configuration of SIEM [126] incident reporting and alerting	A

by $\theta = \{\theta_i \mid 1 \leq i \leq I\}$ where I is the number of users and performs a real-time allocation to a set of N CSPs denoted by $\mathcal{N} = \{n \mid 1 \leq n \leq N\}$. The broker stores the information related to CSPs' security offers and uses it to evaluate the security satisfaction of incoming users' requests. When placing their requests, users could select the security features which they require for the deployment of their services from a drop-down list containing the whole

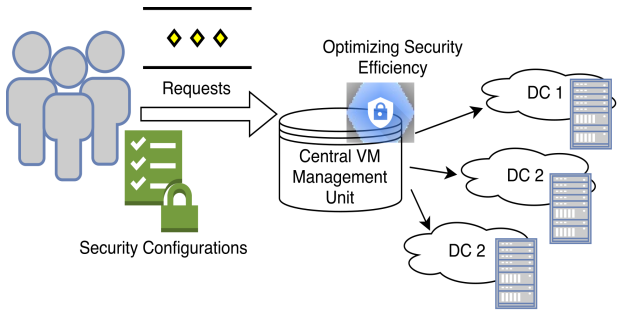


Figure C.1 The security-based resource allocation architecture.

set of L offered security features denoted by $\mathcal{SF} = \{SF_l \mid 1 \leq l \leq L\}$. This will help in automating the allocation process. Users would also assign specific weights to the selected security features to express their relative importance to the deployed Cloud service, since some features could be more significant to the users than others (e.g., sometimes service availability is more critical than data confidentiality). These subjective weights could be computed using the Analytical Hierarchy Process (AHP) technique [130]. When selecting to deploy the security features, users should consider evaluating the appropriate trade-off to deliberate between security level and service performance degradation, since implementing security solutions could sometimes interfere with the constraints on some QoS parameters such as throughput, response time, and scalability, which could be crucial for some application profiles (e.g., interactive workloads, urgent computing, financial simulations, etc). Finally, CSPs will also be able to price their provided services according to the offered security level.

The Cloud resources are usually allocated in the form of VM instances. We consider K predefined types of VMs that are offered by the CSP, each supplying a different combination of resources (i.e., number of CPU cores, memory amount, and storage capacity). The set of VM types is denoted by $\mathcal{VM} = \{k \mid 1 \leq k \leq K\}$ and the set of R resource types is denoted by $\mathcal{R} = \{r \mid 1 \leq r \leq R\}$. At a time t, each CSP $n \in \mathcal{N}$ has an available capacity of each resource type on its data centers, represented in the vector $\rho_n = (\rho_{n,1}, \ldots, \rho_{n,R})$. We believe that the developed security features could be more adapted to exist on the Cloud data center level and not on the physical server level. Therefore, the objective of the model is to evaluate the security satisfaction of users' requests and assign them to the appropriate data centers, which will eventually perform the internal distribution of resources and workload management according to other parameters such as physical servers' resource capacity, power efficiency, and load balancing. Users' requests involve three components : the vector containing the number of VMs that are required of each type $k \in \mathcal{VM}$ denoted by $V_i = (V_{i,1}, \ldots, V_{i,K})$, the selected security features represented by the binary vector SF_i^d that indicates which of the features in the set $\mathcal{SF}$ were selected by the user for deployment (d stands for demand), and the vector

of weights w_i that the user assigns to the selected security features. Hence, a user's request would be denoted by $\theta_i = (V_i, SF_i^d, w_i)$. The total usage $u_{i,r}$ of each resource $r \in \mathcal{R}$ by a request θ_i is calculated as follows :

$$u_{i,r} = \sum_{k \in \mathcal{VM}} V_{i,k} \alpha_{k,r} \quad \forall \theta_i \in \theta, \ \forall r \in \mathcal{R} \tag{C.1}$$

where $\alpha_{k,r}$ is the usage of resource r by the VM type k.
The evaluation of Security Satisfaction (SS) is based on computing the L-dimensional Euclidean distance between security offers and demands, where L is the number of security features. The concept of the Euclidean distance is used to provide an abstraction layer when computing the security satisfaction independently of the nature of the features (qualitative or quantitative). We denote by $SS_{i,n}$ the security satisfaction value provided by CSP $n \in \mathcal{N}$ to request $\theta_i \in \theta$. It aims at measuring how much CSP n adheres to the security features declared in request θ_i, and is computed as follows :

$$SS_{i,n} = \frac{1}{\sqrt{\sum\limits_{SF_l \in \mathcal{SF}, SF_{i,l}^d = 1} w_{i,l} (SF_{i,l}^d - SF_{n,l}^o)^2}} \tag{C.2}$$

where $w_{i,l}$ is the normalized weight assigned by request θ_i to the feature $SF_l \in \mathcal{SF}$ and $SF_{n,l}^o$ holds a binary value indicating if CSP n offers the feature SF_l (o stands for offering). We consider the security satisfaction as inversely proportional to the distance between what is requested by the user and what is offered by the CSP to indicate higher satisfaction when the distance is small.
To compensate for the possible lack of integrity of the CSP when providing the information related to its security offerings, a security reputation parameter is introduced and integrated within the model. The security reputation of a CSP $n \in \mathcal{N}$ is denoted by rep_n and is computed by averaging the ratings provided by its users until the time t of the current allocation. These ratings reflect how much the CSP was honest about satisfying users' security requirements in the past. The information related to CSPs' security reputation is stored on the broker side and updated regularly. The Collaborative Filtering (CF) approach, which is widely used in the development of recommender systems, could be employed in our case to exploit the Cloud users' ratings if an appropriate platform for security satisfaction evaluation is created. In the case where no ratings or only a small number of ratings are available for a particular CSP during the evaluation process, approaches based on the baseline models or deep learning techniques could be integrated to solve the incomplete information problem [161]. After estimating the value of security reputation of a CSP, it is normalized to the interval]0,1] and

multiplied by the value of $SS_{i,n}$ for each request $\theta_i \in \theta$ during the evaluation.
The optimal allocation at time t consists on maximizing the Global Security Satisfaction (GSS) of users' requests to reduce as much as possible the rate of future violations of the Security-SLA. The security-based Cloud resource allocation problem can be considered as a version of the VM Placement Problem (VMPP), which is usually described as a variant of the multi-dimensional Multiple-Knapsack Problem (MKP) [106] from the class of Bin Packing problems (BPP). We formulate the problem as follows :

$$Max\ GSS = \sum_{\theta_i \in \theta} \sum_{n \in \mathcal{N}} rep_n SS_{i,n} x_{i,n} \tag{C.3}$$

Subject to :

$$\sum_{\theta_i \in \theta} u_{i,r} x_{i,n} \leq \rho_{n,r} \quad \forall n \in \mathcal{N},\ \forall r \in \mathcal{R} \tag{C.4}$$

$$\sum_{n \in \mathcal{N}} x_{i,n} \leq 1 \quad \forall \theta_i \in \theta \tag{C.5}$$

$$x_{i,n} = \{0, 1\} \quad \forall \theta_i \in \theta,\ \forall n \in \mathcal{N} \tag{C.6}$$

where I corresponds to the number of items (requests), N to the number of knapsacks (CSPs' data centers), and R to the number of dimensions (resource types). We define $x_{i,n}$ as a binary variable that indicates if request θ_i is allocated to CSP $n \in \mathcal{N}$. Constraint C.4 ensures that the allocated resources of each type will not exceed the available capacity of that type on the CSP's infrastructure, and Constraint C.5 ensures that if a request θ_i is allocated, it will be allocated to only one CSP in $\mathcal{N}$. The multi-dimensional MKP has proven to be NP-complete [106], and no solution in polynomial time exists for this problem. Formal methods used to solve Integer Linear Programming (ILP) problems can only scale up to small problem instances. In our case, we need to solve the problem in real-time mode and be able to respond to a large number of users' requests. Therefore, we propose to use a heuristic approach to find an approximated solution to the problem.

C.4 Evolutionary Computation for Security-based Cloud Resource Allocation

In this section, we propose to solve the security-based Cloud resource allocation problem using the Genetic Algorithm (GA) from the family of Evolutionary Algorithms (EA) [15]. The GA is a population-based meta-heuristic search algorithm that mimics the natural behavior of evolution, and is commonly used in automatic programming and machine and robot learning.

request	1	2	3	4	...	θ_{I-1}	θ_I
CSP	3	2	4	3	...	N	2

Figure C.2 An example of the representation of the GA chromosome in our model.

It is suitable for the approximation of optimal solutions to complex optimization problems and could be considered computationally efficient. The GA usually starts by randomly generating an initial population of candidate solutions called chromosomes, and continuously applies the three genetic operators : selection, crossover, and mutation in order to improve the quality of the generated chromosomes with respect to the optimization problem's objective, until finally selecting the best solution to the problem. The process termination constraints usually involve the maximum number of iterations and early convergence, where the fittest chromosome can not be further improved.

In our model, a chromosome C is defined as an I-dimensional vector representing the assignment of users' requests to the CSPs' data centers. As shown in Figure C.2, the value at a position i in the vector indicates the CSP to which request θ_i will be allocated. For instance, in the shown example, request θ_1 will be allocated to CSP 3's data center and request θ_2 will be allocated to CSP 2's data center. The main operation in the GA is the evaluation of chromosomes, which is based on computing their fitness values. Based on the objective function of our problem represented in Equation C.3, we define the fitness function $f : C \rightarrow \mathcal{R}^+$ to evaluate the quality of each chromosome $C = (C_1, \ldots, C_I)$, as follows :

$$f(C) = \sum_{i=1}^{I} rep_{C_i} SS_{i,C_i} \tag{C.7}$$

The pseudo-code of applying the GA to the security-based Cloud resource allocation problem is presented in Algorithm 6. The algorithm takes the following as inputs : the set θ of users' requests, the amounts of available resource capacities on the CSPs' data centers, and CSPs' security reputation values, and outputs the best solution found by the GA. First, for each request θ_i, the total usage of resources is computed according to the required VM types in the vector V_i using Equation C.1, and the security satisfaction of the request is evaluated for all CSPs according to the selected features in the vector SF_i^d and their assigned weights in the vector w_i using Equation C.2. Then, the process of GA starts by randomly generating a chromosome population of size P and evaluating the fitness value of each chromosome in the population using Equation C.7. The GA then proceeds to create a new child generation

of chromosomes, also of size P, by following a selection process and applying crossover and mutation operations on the chromosomes of the parent generation. The roulette wheel selection method is usually adopted to select two chromosomes of the current population. In our model, we adopt the two-point crossover method that enables the children chromosomes to inherit the good parts of their parents. This is performed when the two selected chromosomes exchange their corresponding parts at a randomly generated position. For the mutation part that follows the crossover operation, we use the bit-flop mutation method where each child chromosome undergoes a modification at a particular randomly generated position in the vector. The crossover and mutation operations happen according to the crossover and mutation rates respectively, which determine their frequency of occurrence during the process. After a new generation of chromosomes is formed, the fitness of each one is evaluated using Equation C.7 and the chromosome with the best fitness value is selected as the current solution ($C_{current}$). If this chromosome is fitter than the one selected from the previous generation, then it becomes the best solution to the problem (C_{best}), which will be returned by the algorithm once one of the termination constraints is satisfied. Since our optimization problem is constrained, we adopt the constrained handling method proposed by [157], which is based on applying a tournament selection operator instead of the greedy selection mechanism. With this approach, when two solutions are compared, the following rules are enforced : a feasible solution is always preferred over an infeasible one, the fittest solution is preferred in the case of two feasible solutions, and the solution with a smaller constraint violation is preferred in the case of two infeasible solutions.

Algorithm 6 involves two parts : the evaluation of requests' security satisfaction, and the optimization using the GA. The time complexity of the first part is $O(I * N)$, where I is the number of requests and N is the number of involved CSPs. On the other hand, the time complexity of one iteration of the GA is determined by the complexity of the three operations : selection, crossover, and mutation. The time complexity of the roulette wheel selection method is $O(P)$, where P is the population size, and that of the crossover and mutation operations is $O(I * P)$. Since in a real life scenario, the number of CSPs in the allocation problem will be very small comparing to the population size in the GA, the total time complexity of Algorithm 6 is $O(I * P * Q)$, where Q is the number of maximum iterations in the GA.

C.5 Simulation and Results

In this section, we study the performance of the proposed algorithm. We model the security-based resource allocation problem using MATLAB, and make use of a Mixed-Integer Linear Programming solver to find the exact optimal solution to the problem. The solver first tries to

Algorithm 6 Pseudo-code of applying GA to the security-based resource allocation problem.

Input:
- The set of received users' requests θ
- The vectors of available resource capacities ρ_n, $n \in \mathcal{N}$
- The security reputation rep_n of each CSP $n \in \mathcal{N}$

Output:
- The best allocation solution C_{best}

for all requests $\theta_i \in \theta$ **do**
    Compute $u_{i,r}$, $\forall r \in \mathcal{R}$
    for all CSPs $n \in \mathcal{N}$ **do**
        Compute $SS_{i,n}$, the security satisfaction of θ_i by n
    end for
end for
$C_{best} \leftarrow$ *empty vector*
Generate the initial population of P chromosomes
Compute the fitness of each chromosome C
while termination constraints are not satisfied **do**
    while the size of the new generation $\neq P$ **do**
        Select two chromosomes using roulette wheel
        Apply two-point crossover on the chromosomes
        Apply bit-flop mutation on child chromosomes
    end while
    Evaluate the fitness of the new generation
    Apply the tournament selection operator to select the
    fittest chromosome as the current best solution $C_{current}$
    if $f(C_{current}) > f(C_{best})$ **then**
        $C_{best} \leftarrow C_{current}$
    end if
end while
return C_{best}

reduce the problem size, then uses heuristics to solve an initial relaxed problem and produce an initial feasible solution, and finally applies a Branch and Bound algorithm [158] to perform an exhaustive search for the optimal solution. Algorithm 6 was also implemented in MATLAB to find the approximated solution using the GA. The experiments are performed on a 64-bit Windows 7 machine equipped with an Intel Core i7-3612QM CPU @ 2 :10 GHz Processor and 12 GB RAM.

C.5.1 Simulation Setup

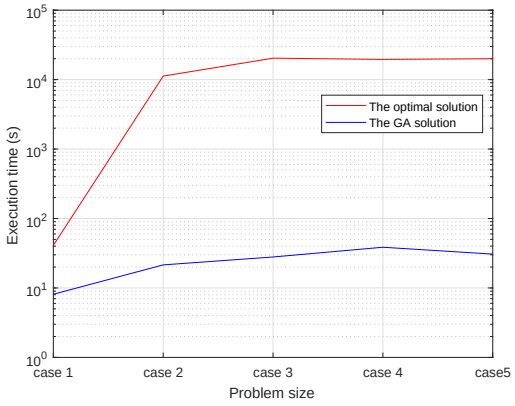
Since we were unable to find real information about Cloud users' requirements with respect to the proposed security features, we decided to generate our own data for the sake of this evaluation. The tests aim at evaluating the performance of the proposed optimization solution in terms of the allocation's optimality and speed. We simulate users' requests as objects, each having a number of VMs of different types, a binary vector of selected security features, and a vector of randomly generated weights that sum to one. We consider the three standard types of Cloud resources in our experiment : the number of CPU cores, the amount of memory (GB), and storage capacity (GB), and four different types of offered VMs like the ones offered by Amazon EC2. The four types are presented in Table C.2 along with their parameters $\alpha_{k,r}$ which correspond to the usage of resource type $r \in \mathcal{R}$ by the VM type $k \in \mathcal{VM}$. For each user, a random number between 0 and 10 is generated for each VM type.
Each CSP was assigned a binary vector of size 20 representing the deployment of security features presented in Table C.1, a vector of resource capacities equal to (1000 CPU, 3000 GB, 300 TB), and a security reputation value between 0.5 and 0.9. We simulate four different scenarios while varying the problem size in terms of the number of received requests I and the number of CSPs involved in the allocation N. The number of requests changes from 100 to 500 with a step of 100, and the number of CSPs changes from 5 to 25 with a step of 5. We set the population size P of the GA to 100 (the most used value in the literature for large size problems) and the number of generations to 1000.

C.5.2 Results Analysis

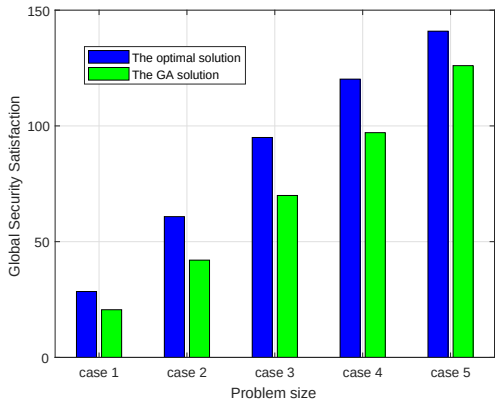
Figure C.3a shows the achieved Global Security Satisfaction (GSS) by both the optimal solution and the heuristic one. The GA was able to achieve at least 80 % of the optimal GSS value in most of the cases, which is an acceptable approximation. Better approximations could have been also achieved if the number of generations was set to be higher. The goal of the proposed model is to satisfy the most of the security requirements in users' requests

Table C.2 VM types offered by Amazon EC2.

	Small (k=1)	Medium (k=2)	Large (k=3)	ExtraLarge (k=4)
CPU cores (r=1)	1	2	4	8
Memory (GB) (r=2)	1.7	3.75	7.5	15
Storage (GB) (r=3)	160	410	850	1690



(a) The achieved Global Security Satisfaction.



(b) The execution time of the algorithm.

Figure C.3 Evaluation of the performance of the proposed solution in five different scenarios : case 1, $I = 100$, $N = 5$; case 2, $I = 200$, $N = 10$; case 3, $I = 300$, $N = 15$; case 4, $I = 400$, $N = 20$; case 5, $I = 500$, $N = 25$.

according to Equation C.2. According to the results, the GA-generated solution will to some extent enable the broker to avoid assigning the requests to unfulfilling CSPs and reduce Security-SLA violations. We believe that providing the proposed security features and taking them into consideration during the process of resource allocation will help increasing the protection of deployed services and boosting the Cloud adoption.

Figure C.3b shows the execution time required to find the optimal and approximated solutions on a logarithmic scale. The GA clearly demonstrates its ability to function in real-time, since it required at most 38 s to allocate users' requests to CSPs' data centers for the tested scenarios. Finding the optimal solution even for a small size problem instance would require high computational power, and would take up to 10^4s. The execution time of the GA was three orders of magnitude lower than that of the optimal algorithm.

C.6 Conclusion

In this paper, the problem of resource allocation in the Cloud was addressed from a security perspective. A set of security features that could be offered by CSPs to secure the deployed services was developed, and an allocation model based on requests' security satisfaction was proposed. The problem was solved using the genetic algorithm from the family of evolutionary computation, and results showed that the algorithm is able to find an acceptable approximation of the optimal solution and is computationally efficient. This will permit the online allocation of resources in the Cloud on the fly while considering the security factor. Other

factors such as performance and QoS parameters could also be integrated into the model to create a complete resource allocation framework. In the future, we plan to extend the study on the set of security features according to real data that we are collecting from several CSPs. We are also working on addressing the challenges to security satisfaction optimization using a model-free approach, where the system can automatically learn the optimal allocation based on its previous experience.

APPENDIX D

ARTICLE 9 : CLOUD SECURITY UP FOR AUCTION : A DSIC ONLINE MECHANISM FOR SECURE IAAS RESOURCE ALLOCATION

Talal Halabi, Martine Bellaiche, and Adel Abusitta

Accepted for presentation at *the 2nd Cyber Security in Networking Conference (CSNet 2018)*, Paris, France, October 2018

Abstract

The lack of security is still one of the main factors that are blocking the full migration towards the Cloud Computing paradigm. More businesses would be attracted by the adoption of the Cloud model if the Cloud Infrastructure Providers (CIP) start increasing their investment in security solutions and demonstrating more explicitly the potency of their infrastructures in protecting customers' data and services. However, implementing security solutions is usually costly, and does not necessarily generate higher revenues. One way to reduce the cost of security investments would be to rent the Cloud secure resources to customers in a competitive fashion. The CIP could place her added value of security up for auction, and customers would place their bids along with their resource provisioning requests. In this paper, we propose an online mechanism that performs the allocation of the CIP's resources to customers in a security-oriented auction-based fashion. The mechanism is Dominant-Strategy Incentive-Compatible (DSIC), i.e., it guaranties the truthfulness of the bidders. The mechanism aims at allocating the resources to customers who valuate their security the most, and shows an acceptable performance compared to the famous offline Vickrey-Clarke-Groves (VCG) truthful mechanism.

D.1 Introduction

The adoption of Cloud Computing technology continues to rise at an accelerated pace. Recent statistics [1] show that the total worldwide Cloud IT infrastructure revenue has almost tripled in the last four years. According to a new study [11], it is expected that in only 15 months, 80% of all IT budgets will be invested in Cloud applications and solutions. Many

organizations are deciding to move their businesses to the Cloud to benefit from its attractive features such as scalability, resilience, and cost reduction, and deliver an enhanced user experience to their customers. In a typical Cloud Infrastructure-as-a-Service (IaaS) setting, a Cloud Infrastructure Provider (CIP) offers different types of resources (e.g., storage capacity, processing power, etc) to customers in the form of Virtual Machine (VM) instances, on which they deploy their applications and flexibly manage their workload. However, outsourcing data and services to a third party adds a new level of risk due to loss of physical control and introduces many security threats such as data breaches, data loss, and denial of service [70], which makes security an essential driving factor of the Cloud market today. Organizations expect from the CIP to maintain the security and availability of their data and services. Every year, the Cloud market giants witness severe incidents and security breaches causing Cloud outages and failures, which affect the production process and result in lost data and revenue. For instance, in 2017, Microsoft Skype Europe users suffered from connectivity problems due to an apparent Distributed Denial of Service attack (DDoS) that affected the whole communications platform [12]. Similarly, in 2016, a number of Amazon Web Services (AWS) VM instances hosting critical workloads for big companies subsequently failed because of a power outage in the region of Sydney, Australia, resulting in a serious service disruption [13]. Customers blamed AWS, the world's largest CIP, for not being sufficiently prepared for such incidents.

According to the statistics in [1], the rate of organizations that completely trust public Cloud infrastructures today to protect their data is only at 23%. The organizations usually tend to increase their trust in the security of public clouds and their use of the Cloud services when the level of integration with security solutions is high. However, investing in Cybersecurity by deploying advanced security solutions and providing a full security integration across multiple Cloud environments normally introduces additional costs to the CIP's budget, and does not necessarily generate higher profit. These costs include the price of security infrastructure (e.g., firewalls, antivirus software, intrusion detection systems, etc), salaries of security architects, and the expenses of security training programs. Investing in security is not necessarily a priority for the CIPs, but has apparently become indispensable to boost the migration towards the Cloud business model. The return on security investment in an IT infrastructure is not usually measured in terms of the achieved revenue, but as a function of damage reduction, i.e, the decrease in loss expectancy due to security incidents and the rate of threat occurrence. According to the study in [14], the average total cost of a data breach is around 4 million $. This gives an idea about how crucial is to secure a Cloud infrastructure that hosts thousands of services and huge amounts of users' sensitive data.

A way for the CIPs to reduce the cost of security investments would be to offer their se-

curity added value up for auction, and make sure that the customers who are renting their resources are the ones who valuate their security the most. A customer would include a bid in her resource provisioning request, which expresses her valuation of security. The bid does not involve the actual price of her requested VM package, which is estimated according to the CIP's fixed price scheme. Hence, the higher her bid is, the stronger her appreciation of the integrated security element is reflected. On the other hand, it will be possible for a customer to manipulate the auction and negatively affect the outcomes for the other bidders, by declaring untrue amounts of VMs in her request or a bid that is different from her actual valuation of the request. To avoid these situations, the auction should satisfy the incentive-compatibility property, i.e., to give incentives to bidders to reveal their true requests and valuations. In this paper, we design a Dominant Strategy Incentive-Compatible (DSIC) online mechanism that allocates the CIP's secure resources to customers in real-time while guarantying their truthfulness, i.e., giving them incentives to reveal their actual resource provisioning requests and security valuations as a dominant strategy. The mechanism is based on a greedy allocation rule in which customers are prioritized according to their valuation of the security of these resources. Compared to the famous Vickrey–Clarke–Groves (VCG) [98] offline auction model, the proposed mechanism is computationally efficient and permits to achieve a good approximation of customers' optimal social welfare achieved by the allocation.
The remainder of the paper is organized as follows. Section D.2 discusses the work that is related to the addressed topic. Section D.3 describes the problem of allocation of Cloud IaaS secure resources, along with the mechanism design requirements. In Section D.4, the proposed online mechanism is described. In Section D.5, experimentation is performed and results are analyzed. Finally, section D.6 concludes the paper.

D.2 Related Work

This work tackles the problem of resource allocation in the Cloud from a security perspective. Along with Quality of Service (QoS) and power efficiency, profit maximization has been an essential factor when it comes to resource allocation in the Cloud. For instance, Goudarzi and Pedram [94] proposed a distributed solution to a SLA-based resource allocation problem, which maximizes the total profit in the system while considering the following three dimensions in the optimization : processing, data storage, and communication bandwidth. Also, Nezarat and Dastghaibyfard [95] proposed a game theoretical model to maximize profit in a Cloud environment. In their model, a combinatorial auction mechanism is used to select the winners among the competent users.
The design of truthful resource allocation mechanisms in the Cloud has been previously stu-

died by several researchers. For instance, Zaman and Grosu [164] proposed an online mechanism for VM provisioning and allocation in clouds. Their mechanism is incentive-compatible and aims at allocating VM instances whenever enough resources and matching bids are available. Zhang et al. [97] proposed an incentive-compatible online auction-based mechanism that allocates Cloud resources to users with heterogeneous demands. Their mechanism discourages the Cloud users from following a dishonest behavior when requesting resources, and its performance is comparable to that of the VCG mechanism. Nejad et al. [165] and Mashayekhy et al. [96] proposed an auction-based model for the problem of dynamic provisioning and allocation of VMs in the Cloud, and designed a truthful greedy mechanism that allocates the requests of the winning users and calculates their payments. Their mechanisms give incentives to users to be honest about their requests and valuations.
The main difference between these approaches and ours is that the latter aims at capturing customers' valuation of security, and not of the Cloud resources provided by the CIP. In our model, these resources will continue to be offered on the basis of a fixed-price scheme, but their security will be up for auction. If customers' rationality is assumed, i.e., their security valuations will be proportional to their need for security, the proposed mechanism will not only allow the CIP to reduce her cost of security investment, but also allocate her secure resources to the customers who need them the most. Eventually, the security risk level on her infrastructure and the probability of Cloud failures and data breaches will be reduced.

D.3 System Model

In this section, we present and model the problem of allocating the secure resources in a Cloud infrastructure in an auction-based context.

D.3.1 General Description

The Cloud IaaS delivery model consists in providing customers with a set of computational and storage resources in the form of VM instances which they will use to deploy their applications and perform their tasks. In a multi-tier architecture, these customers could be Cloud Service Providers (CSP) who rent these resources to provide different services (e.g., web hosting, scientific computing, financial simulations, email services, etc) to their users. Businesses normally tend to work with the IaaS model since it allows them to manage their workload without the need to manage the underlying infrastructure. Although customers could have the control over securing their VMs according to the security requirements of their applications, the process of securing the Cloud low level resources and supplying the customers with the right security capabilities is still in the hands of the CIP. The basic architecture of

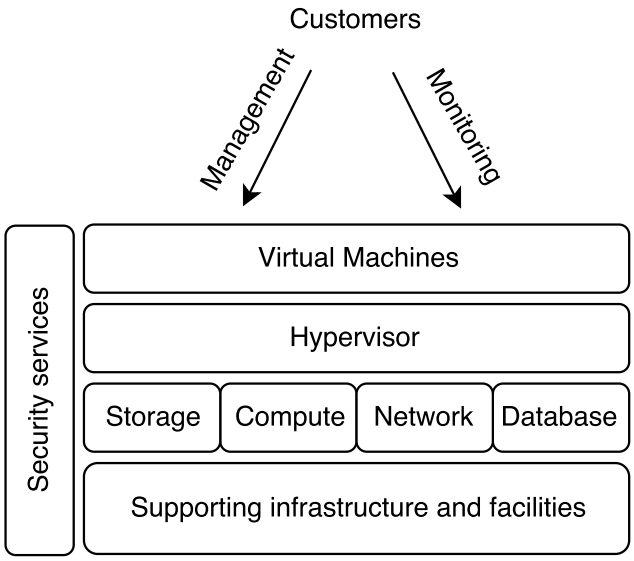
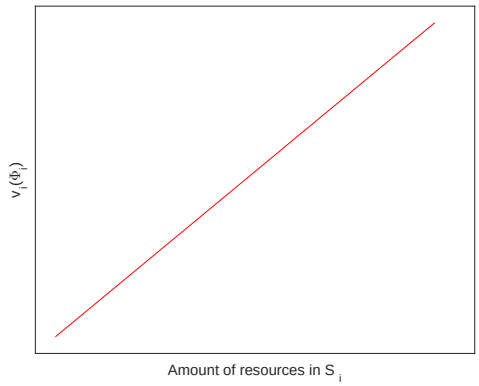


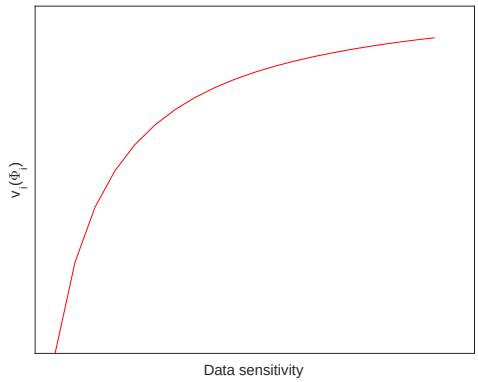
Figure D.1 The IaaS model basic architectural components.

the IaaS model is depicted in Figure D.1. Securing the infrastructure would imply to secure every component of this architecture. For instance, implementing monitoring solutions on the hypervisor level, providing encrypted VM live migration, scanning and filtering VM images before creation, and configuring full VM backup capabilities could be all part of a security investment plan on the virtualization layer. Similarly, to ensure secure data storage, the CIP might need to implement robust encryption key management solutions, support secure data deletion, deploy data leakage prevention techniques, and increase the data backup frequency. These security investment plans normally introduce additional costs to the CIP's budget, but are critical to protect her customers' workload and increase her trustworthiness. Customers would be ready to be part of this investment since they are the primary beneficent. In general, the CIP charges her customers on the basis of a pay per use service model according to the amount of resources they provision. By keeping this model in place, and placing her resources up for auction after integrating the security aspect, the CIP will become aware of her customers' valuation of security. She will then be able to allocate her secure resources to those customers who valuate their security the most, and probably reduce her cost of security investment. This strategy will encourage the CIP to continuously improve the security level of her infrastructure, and allow the customers who are mostly in need for this increased level of security to benefit from it.

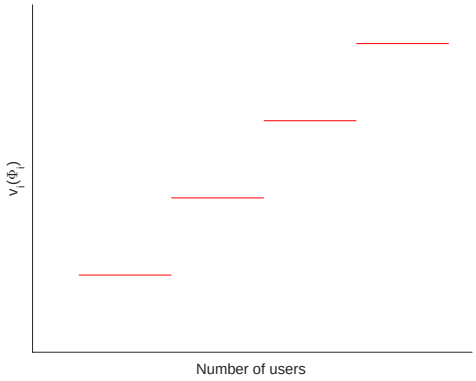
In our resource allocation model, we consider R types of resources provided by the CIP (e.g., number of CPU cores, memory amount, and storage capacity) and represent them in the



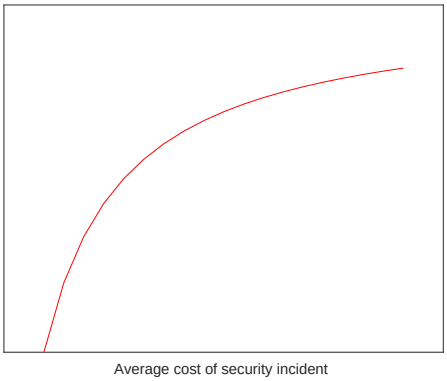
(a) Security valuation in terms of the amount of resources requested by the customer.



(b) Security valuation with respect to the level of sensitivity or confidentiality of customer's data.



(c) Security valuation as a function of the number of users of customer's application.



(d) Possible relationship between customer's security valuation and the estimated cost of a security incident.

Figure D.2 Examples of customers' security valuation functions.

set $\mathcal{R} = \{r,\ 1 \leq r \leq R\}$. We also consider K types of VMs that are offered by the CIP, each type provides a different combination of resources. The set of VM types is denoted by $\mathcal{VM} = \{k \mid 1 \leq k \leq K\}$. We consider a set $\mathcal{I} = \{i,\ 1 \leq i \leq I\}$ of customers, each requesting a specific VM package to manage their workload. We denote by $\Phi_i = (S_i, b_i)$ the request of customer $i \in \mathcal{I}$, where $S_i = (S_{i,1}, \ldots, S_{i,K})$ is a vector containing the required amounts of VM instances of each type $k \in \mathcal{VM}$, and b_i is the customer's bid, which is the monitory value that she is ready to pay to benefit from the added security element if her request is allocated to the infrastructure. The total usage $u_{i,r}$ of a resource $r \in \mathcal{R}$ by a request Φ_i is calculated as follows :

$$u_{i,r} = \sum_{k \in \mathcal{VM}} S_{i,k}\alpha_{k,r} \quad \forall i \in \mathcal{I},\ \forall r \in \mathcal{R} \tag{D.1}$$

where $\alpha_{k,r}$ is the usage of r by the VM type $k \in \mathcal{VM}$.
Customers in our model are assumed to be single-minded, meaning that, they are willing to pay their bids b_i if, and only if, they were able to get the allocation of their requests or a super set of them. The bid expresses the customer's valuation of security when her request is allocated for one normalized unit of time. Eventually, the payment that the customer will make will be computed using this value and according to the period of time for which her workload was executed, since this information is not always known prior to the allocation process. The bid b_i is generated by the customer's security valuation function defined as $v_i : \Phi_i \rightarrow \Re^+$, which reflects the benefit that customer i receives when her request Φ_i is allocated in the presence of security integration, with $v_i(\Phi_i) = b_i$. This benefit is usually related to the degree of damage that a security breach could cause to the customer's service or data, and multiple factors are normally involved in its estimation like the number of users of her application, the level of sensitivity or required confidentiality of their data, or the cost of service downtime.
In Figure D.2, four examples of customers' security valuation functions are shown. For instance, Figure D.2a illustrates how security valuation can be proportional to the amount of resources that the customer declares in her request, since provisioning more resources will most probably entail an increase in the application of security capabilities (e.g., provisioning more VMs would require to increase the number of storage discs available for backups). Figure D.2b highlights data sensitivity as the primary factor that helps the customer in evaluating her value of security. On the other hand, this valuation will eventually tend towards a limiting budget from the customer's perspective, which explains the concave increasing form of the function. Similarly in Figure D.2c, the customer estimates her appreciation of the security element according to the number of users of her deployed Cloud application. When this number increases, the customer becomes responsible for protecting higher amounts of sensitive data that belong to the users, and will consequently be motivated to expand her investment in advanced security measures. This situation is also reflected in Figure D.2d. Without loss of generality, the security valuation function is considered to be the customer's private information, and the allocation problem that we try to solve in this paper consists in effectively extracting this information from the customer during the auction.

D.3.2 Problem Formulation

In our model, we denote by $\widehat{\Phi_i} = (\hat{S}_i, \hat{b}_i)$ the declared request of customer $i \in \mathcal{I}$ and by $\Phi_i = (S_i, b_i)$ her true request. The problem consists in maximizing the social welfare [98] of the customers, which is a standard criterion that is used to evaluate the outcome of an auction mechanism. Maximizing the social welfare will allow the CIP to reduce her cost of

security investments by allocating her secure resources to the customers who valuate them the most. Customers are assumed to be rational in our model, i.e., their security valuations will depend on how much they are in need for the secure Cloud resources. Hence, maximizing the social welfare will also allow the customers who mostly are in need for the security added value to benefit from it to protect their services and data. Eventually, the security risk level on the infrastructure will drop and the probability of security incidents will be reduced. The social welfare S is defined as the sum of security valuations of the allocated customers. In this security-oriented context, the secure Cloud resource allocation problem that aims at maximizing customers' social welfare can be defined as a Multi-dimensional Knapsack Problem (MKP) [106], which is a basic version in the class of Bin Packing problems. We formulate the problem as a Constrained Integer Linear Programming (CILP) problem as follows :

$$\text{Max} \quad S = \sum_{i \in \mathcal{I}} v_i(\widehat{\Phi}_i) x_i \tag{D.2}$$

Subject to :

$$\sum_{i \in \mathcal{I}} u_{i,r} \; x_i \leq \rho_r \quad \forall r \in \mathcal{R} \tag{D.3}$$

$$x_i = \{0, 1\} \quad \forall i \in \mathcal{I} \tag{D.4}$$

where x_i is a binary decision variable defined $\forall i \in \mathcal{I}$ as follows : $x_i = 1$ if Φ_i is allocated, and 0 otherwise. Constraint (4) guarantees that the allocation of all types of Cloud resources respects their available capacities $\rho_r, \; r \in \mathcal{R}$, with $\rho = (\rho_1, \ldots, \rho_R)$ is the vector of capacities provided by the CIP for each resource. The problem is considered to be strongly NP-hard [106], and finding the optimal solution is computationally inefficient for large scale problems. The truthful VCG auction mechanism can only work in an offline scenario and can not be deployed to dynamically handle customers' requests in a real-time fashion, since it requires the existence of an optimal allocation rule. Therefore, we design in this paper an online truthful mechanism to solve the problem of secure resource allocation in the Cloud.

D.3.3 Design Requirements

The aim of this paper is to design a Dominant Strategy Incentive-Compatible (DSIC) mechanism that allocates the CIP's secure resources to customers in real time, that is, a mechanism that gives incentives to customers to declare their true requests for resource provisioning and their true security valuations. Mechanism design [107] is an interesting approach to solving online allocation problems involving dynamic multi-agent environments where players should

make truthful announcements for the sake of better system performance. Designing online mechanisms is becoming handy in multiple areas such as wireless networking, web servicing, and Cloud Computing. An online DSIC auction mechanism is usually defined by the combination of an allocation rule $\mathcal{A}(.)$ that determines which bidders receive their requested items, and a truth-inducing payment rule $\mathcal{P}(.)$ that determines based on the allocation results, the amount that each bidder must pay for her received item [98]. The allocation rule tries to maximize bidders' utility. The utility of a customer $i \in \mathcal{I}$ is defined as the difference between her valuation of her request and the payment P_i that she should make if the request is allocated :

$$utility_i = v_i(\widehat{\Phi}_i) - P_i \quad \forall i \in \mathcal{I} \tag{D.5}$$

Since bidders are assumed to be selfish players, their goal is to maximize their utilities, which could be done by manipulating their requests, that is, lying about the required amounts of resources or the placed bids. The objective of our mechanism is to prevent such manipulations by giving incentives to customers to reveal their true requests. Truthfulness in our case will ensure the right distribution of the secure Cloud resources to the customers who really need them. We denote by $\widehat{\Phi}$ the set of all declared requests and by $\widehat{\Phi}_{-i}$ the set of declarations excepted from the request of customer i. A mechanism is incentive-compatible if for every customer i, for every declaration of the other customers $\widehat{\Phi}_{-i}$, a true request Φ_i, and any other declared request $\widehat{\Phi}_i$ of customer i, we have $utility_i(\Phi_i, \widehat{\Phi}_{-i}) \geq utility_i(\widehat{\Phi}_i, \widehat{\Phi}_{-i})$. That means that declaring their true requests is a dominant strategy for the customers, which implies that they will maximize their utilities if they truthfully report their requests independently of the declarations of the other customers.

An incentive-compatible mechanism requires a monotone allocation rule, which means that if the rule allocates the request $\widehat{\Phi}_i$ to customer i as declared, then it will also allocate a more preferred request $\widehat{\Phi}'_i$ of customer i. We define the preference relationship $\succeq$ as follows : a request $\widehat{\Phi}'_i = (\widehat{S'}_i, \hat{b}'_i)$ is more preferred than or equally preferred as a request $\widehat{\Phi}_i$ (denoted as $\widehat{\Phi}'_i \succeq \widehat{\Phi}_i$), if the amount of resources required to provision the vector $\hat{S}'_i$ is less than or equal to the amount required to provision $\hat{S}_i$, and $\hat{b}'_i \geq \hat{b}_i$. In other words, if the customer requests the provisioning of less resources and declares a higher security valuation in the request. The mechanism also requires the design of a payment rule that evaluates the payment of customers independently of their requests. The payment rule should be based on a critical value, which is a unique value b_i^c defined for each bidder i, such that all the requests that are more preferred than or equally preferred as (S_i, b_i^c) are winning declarations, and all the requests that are less preferred than (S_i, b_i^c) are losing declarations.

D.4 The Design of an Online Mechanism

In this section, we first apply the VCG mechanism to the problem of secure resource allocation in the IaaS Cloud. This mechanism is usually implemented in offline settings, where the auctioneer have all the information about future requests before the time of the allocation. Then, we describe the online truthful mechanism that we propose to respond to the dynamic and real-time requirements of our problem.

D.4.1 VCG-based Allocation

The VCG auction model [98] aligns all bidders' incentives with the goal of maximizing the social welfare, which is achieved by telling the truth. With a VCG-based mechanism, the allocation rule is usually required to be implemented using an optimal allocation algorithm. In our case, we integrate the CILP problem presented in the previous section in the design of a VCG-based mechanism to find the optimal allocation. The mechanism also requires the implementation of a payment rule that computes the payments of customers based on the allocation results, which is defined as follows :

$$P_i = \sum_{j \in \mathcal{A}(\widehat{\Phi}_{-i})} v_j(\widehat{\Phi}_j) - \sum_{j \in \mathcal{A}(\widehat{\Phi}), j \neq i} v_j(\widehat{\Phi}_j) \quad \forall i \in \mathcal{I} \tag{D.6}$$

where the first term of the equation reflects the optimal social welfare that would have been achieved by all players if customer $i \in \mathcal{I}$ had not participated in the auction (allocation of $\widehat{\Phi}_{-i}$), and the second term represents the social welfare achieved by all players except for i, when i had participated in the auction (allocation of $\widehat{\Phi}$). In other words, the payment of a bidder is evaluated as being the harm that the bidder had caused to the other bidders by participating in the auction.

The VCG-based allocation mechanism is presented in Algorithm 7. It consists of performing the optimal allocation by solving the CILP problem (Equations D.2, D.3, and D.4) that maximizes the social welfare, and implementing the payment rule defined by the VCG model (Equation D.6). The mechanism takes two inputs : the set of customers' declared requests $\widehat{\Phi}$ and the vector of Cloud resource capacities ρ, and generates two outputs : the optimal allocation vector x, and the vector of payments P. First, the mechanism determines the optimal allocation of requests (line 1) using a dynamic programming optimization algorithm and generates the allocation vector x and the optimal social welfare S. Then, it computes the payment of each customer (lines 2-9). According to the VCG model, to determine the payment of an allocated customer $i \in \mathcal{I}$, we solve the allocation optimization problem without the participation of i (line 4) and compute the social welfare S^* achieved by the new allocation

vector x^*. The payment of the customer will then be obtained by computing the difference between the two values S^* and $S - \hat{b}_i$ (line 5). If this difference is null (the requests allocated by x and x^* are exactly the same), or customer's request was not allocated, the customer will have nothing to pay. In other words, the participation of the customer in the auction had no affect on the other players.

The VCG-based mechanism is incentive-compatible [98], and provides the CIP with the optimal resource allocation strategy. However, the defined CILP problem is strongly NP-hard and becomes computationally inefficient when the number of customers and the amounts of available resource capacities become large, which makes the mechanism suitable to be executed only in an offline setting. To be able to allocate the Cloud resources on the fly while guarantying the truthfulness of customers, we need to design an incentive-compatible allocation mechanism that is able to provide a good approximation of the optimal social welfare in polynomial time.

Algorithm 7 Offline VCG-based allocation.

Input:
- Customers' declared requests in the set Φ
- The vector of resource capacities ρ

Output:
- The allocation vector x
- The payment vector P

1: (x, S) = Solve the CILP problem for $\widehat{\Phi}$
2: **for all** $i \in \mathcal{I}$ **do**
3: **if** $x_i = 1$ **then**
4: (x^*, S^*) = Solve the CILP problem for $\widehat{\Phi}_{-i}$
5: $P_i = S^* - (S - \hat{b}_i)$
6: **else**
7: $P_i = 0$
8: **end if**
9: **end for**
10: $P \leftarrow (P_1, \ldots, P_I)$
11: **return** x, P

D.4.2 The Proposed Mechanism

We design in this section an incentive-compatible mechanism that allocates the secure Cloud resources in real-time according to customers' security valuations. In the case where large instances of the problem are handled (e.g., tens of thousands of items offered for sale), finding the optimal allocation becomes computationally exhausting. The goal here is to design a non-VCG mechanism that finds an acceptable level of approximation of the optimal solution using heuristics, in an incentive-compatible fashion. The mechanism should optimize the social welfare as much as possible, while guarantying the truthfulness of the bidders. In other words, it should give incentives to customers to always prefer to truthfully report their private valuations of the offered security settings to the CIP, rather than any malicious or dishonest declaration that could have a potential in increasing their utilities.

D.4.2.1 Mechanism Description

Since customers provision the requested VM instances in the form of R types of Cloud resources, any request $\hat{\Phi}_i$ with a vector $\widehat{S}_i$ of requested VMs can be mapped into an R-dimensional space of items, where the smallest item in this space contains one unit of each of the normalized R resources allocated for one unit of time. The goal here is to allocate these items according to the valuations of their security.

The online mechanism is presented in Algorithm 8, and is executed in the event where resource provisioning requests were placed at a time $t \in T$ (T is the infinite interval of time). Before the execution, the amount of remaining resources at time t within the Cloud is computed by considering all the previously allocated requests which execution is not yet completed. We denote by $\rho^t = (\rho_1^t, \ldots, \rho_R^t)$ the vector of available capacities of resources of all types at time t. The mechanism involves two stages : an allocation procedure and payment calculation. It receives two inputs : the vector of available resource capacities ρ^t at time t, and the set $\widehat{\Phi}^t = \{\widehat{\Phi}_i \mid i \in \mathcal{I}, \widehat{\Phi}_i \text{ is declared at time } t\}$ of requests, and generates three outputs : the allocation vector x, the payment vector P, and the updated vector of available resource capacities ρ^{t+1}. The mechanism starts the execution only if there were new requests that were placed at time t, and if resources are available within the Cloud (line 1). In this case, the mechanism will perform a heuristic allocation of requests using a greedy algorithm [98] and find the allocation vector x (lines 3-11). Next, the mechanism will compute the vector P of payments that should be made by the allocated customers (lines 14-34).

D.4.2.2 Allocation Procedure

Let $\hat{u}_{i,r}$ be the requested amount of resource $r \in \mathcal{R}$ by customer $i \in \mathcal{I}$ computed using Equation D.1 according to the vector $\hat{S}_i$, and let $\hat{u}_i$ be the vector containing the declared amounts of all resource types. In our mechanism, the allocation is performed using a greedy procedure and based on the Valuation of Security per Item (VSI) parameter that we define for a bidder $i \in \mathcal{I}$ as follows :

$$VSI_i = \frac{\hat{b}_i}{\prod\limits_{r \in \mathcal{R}} \hat{u}_{i,r}} \tag{D.7}$$

The VSI_i parameter reflects how much customer i valuates the security of each unit of resources allocated for one unit of time. In other words, it shows the customer's security valuation for the smallest item in the R-dimensional space of items.

The mechanism allocates the Cloud resources to customers in decreasing order of their declared VSI. It first sorts the requests $\widehat{\Phi}_i$ placed at time $t \in T$ in $\widehat{\Phi}^t$ in decreasing order of their VSI_i (line 5), then performs the allocation of the sorted requests while resources are still available in ρ^t, and updates the allocation vector x (lines 6-11). The mechanism adopts this concept since it aims at maximizing the social welfare of the players, which is the sum of the reported bids by the allocated customers. The mechanism helps the CIP in reducing the cost of security investments by allocating the secure resources to the customers who are ready to pay more for their security. Moreover, it aims at providing the secure resources to customers who really need them. The allocation procedure tries to approximate the optimal social welfare in polynomial time. Sorting customers' requests introduces a time complexity of $O(n\ log\ n)$ on average, where n is the number of requests, and verifying their allocation feasibility requires $O(n)$. Hence, the overall time complexity of the proposed allocation procedure is $O(n(log\ n + 1))$.

D.4.2.3 Payment Calculation

After the mechanism finds the near-optimal allocation of resources at time t, it computes the payment of each allocated customer, which is considered to be the minimum value a customer should declare in order to get the requested secure resources. To compute the payment of an allocated customer at time t, the mechanism implements the concept of Equation D.6 in real-time. First, it supposes that customer $i \in \mathcal{I}$ was not participating in the auction, and re-adds the resources that were allocated to i to the vector ρ^t by creating a new vector $\tilde{\rho}$ (lines 16-17). Then, it starts allocating the resources in $\tilde{\rho}$ to the customers that have lower VSI values than that of i in the set $\widehat{\Phi}^t \backslash \widehat{\Phi}_i$, and which were not allocated the first time, when customer i was included. The mechanism stops when the first customer j who has the VSI

Algorithm 8 The proposed online mechanism.

Input:
- The vector of available resource capacities ρ^t
- The set of requests $\widehat{\Phi}^t$

Output:
- The allocation vector x
- The payment vector P
- The updated vector of resource capacities ρ^{t+1}

1: **if** $\hat{\Phi}^t \neq \emptyset$ and $\rho^t \neq 0$ **then**
2: {Heuristic allocation of ρ^t to $\widehat{\Phi}^t$}
3: $x \leftarrow$ null vector of size I
4: Calculate VSI_i $\forall \widehat{\Phi}_i \in \widehat{\Phi}^t$
5: Sort $\widehat{\Phi}_i \in \widehat{\Phi}^t$ in decreasing order of VSI_i
6: **for all** $\widehat{\Phi}_i \in \widehat{\Phi}^t$ in decreasing order of VSI_i **do**
7: **if** $\rho_r^t - \hat{u}_{i,r} \geq 0 \ \ \forall r \in \mathcal{R}$ **then**
8: $\rho^t \leftarrow \rho^t - \hat{u}_i$
9: $x_i \leftarrow 1$
10: **end if**
11: **end for**
12: $\rho^{t+1} \leftarrow \rho^t$
13: {Payment}
14: **for all** $\widehat{\Phi}_i \in \widehat{\Phi}^t$ in decreasing order of VSI_i **do**
15: **if** $x_i = 1$ **then**
16: $\widetilde{\rho} \leftarrow \rho^t$
17: $\widetilde{\rho} = \widetilde{\rho} + \hat{u}_i$
18: {Allocation of $\widetilde{\rho}$ to $\widehat{\Phi}^t \backslash \widehat{\Phi}_i$}
19: $found \leftarrow FALSE$
20: **for all** $\widehat{\Phi}_j \in \widehat{\Phi}^t \backslash \widehat{\Phi}_i$ in decreasing order of VSI_j and with $VSI_j < VSI_i$ **do**
21: **if** $x_j = 0$ and $\widetilde{\rho}_r - \hat{u}_{j,r} \geq 0 \ \ \forall r \in \mathcal{R}$ **then**
22: $P_i \leftarrow VSI_j \cdot \prod_{r \in \mathcal{R}} \hat{u}_{i,r}$
23: $found \leftarrow TRUE$
24: break
25: **end if**
26: **end for**
27: **if** $found$ is $FALSE$ **then**
28: $P_i \leftarrow 0$
29: **end if**
30: **else**
31: $P_i \leftarrow 0$
32: **end if**
33: **end for**
34: $P \leftarrow (P_1, \ldots, P_N)$
35: **return** x, P, ρ^{t+1}
36: **else**
37: **return**
38: **end if**

value lower than VSI_i and who was not allocated when customer i was participating in the auction, gets allocated when customer i did not participate (lines 19-26). The mechanism evaluates the payment of customer i based on the security valuation VSI_j of customer j (line

22), which constitutes the highest VSI value among losing bidders. The intuition behind this is that bidder j is the bidder who lost exactly due to the participation of bidder i in the auction. This payment is the minimum value that the customer should report to get her request. If no such customer was found, or the request of customer i was not allocated by the mechanism, i will be required nothing to pay (lines 27-32). In other words, her participation in the auction did not affect the other participating customers, which would have achieved the same social welfare, if customer i had not participated.

D.4.2.4 Incentive-Compatibility

The proposed mechanism is incentive-compatible, since it satisfies the monotonicity and critical value properties. For instance, if a bidder $i \in \mathcal{I}$ is able to allocate her request $\widehat{\Phi}_i$, she will also be able to allocate a more preferred request $\widehat{\Phi}'_i \succeq \widehat{\Phi}_i$, since declaring a vector $\hat{S}'_i$ with less required resources than in $\hat{S}_i$, or a bid $\hat{b}'_i \geq \hat{b}_i$ will only help the bidder in increasing her VSI_i and moving up in the greedy order, making it easier to win. This implies that the bidder will not have incentives to declare a different request than the one she actually wants in order to guarantee her winning.
The mechanism implements payment calculation based on the minimum value that the bidder must declare in order to get the allocation of her request. Hence, if bidder i declares a lower value than this minimum value, she will lose the allocation ; otherwise she will win. This minimum value is the critical value of bidder i. Since our mechanism implements a monotone allocation procedure and a payment calculation based on the critical value, it is considered incentive-compatible [98].

D.5 Experimentation and Results

In this section, we conduct a set of evaluation experiments and analyze the results. We implement the VCG-based allocation mechanism and the proposed online mechanism in MATLAB, and use a Mixed-Integer Linear Programming solver to find the optimal solution to the defined CILP problem. The solver first tries to reduce the problem size, then uses heuristics to solve an initial relaxed problem and produce an initial feasible solution, and finally applies a Branch and Bound algorithm [158] to perform an exhaustive search for the optimal solution. The experiments are performed on a 64-bit Windows 7 machine equipped with an Intel Core i7-3612QM CPU @ 2 :10 GHz Processor and 12 GB RAM.

Table D.1 VM types offered by Amazon EC2.

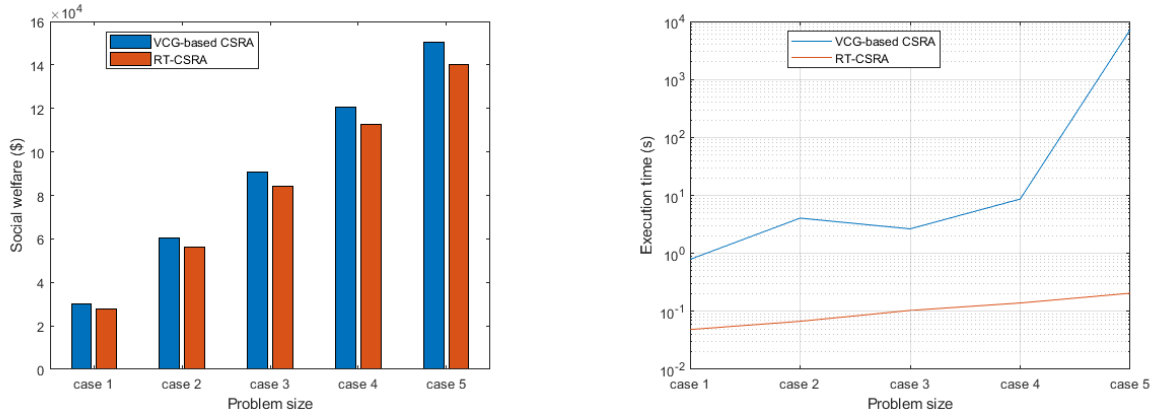
	Small (k=1)	Medium (k=2)	Large (k=3)	ExtraLarge (k=4)
CPU cores (r=1)	1	2	4	8
Memory (GB) (r=2)	1.7	3.75	7.5	15
Storage (GB) (r=3)	160	410	850	1690

D.5.1 Experimental Setup

The experiments aim at evaluating the performance of the designed mechanism in terms of the approximation of the social welfare and computational efficiency, as well as demonstrating the incentive-compatibility property. We consider the three standard types of Cloud resources in our experimentation, the number of CPU cores, the amount of memory (GB), and storage capacity (GB), and four different types of offered VM instances like the ones offered by Amazon EC2. The four types are presented in Table D.1 along with their parameters $\alpha_{k,r}$ which correspond to the usage of resource type $r \in \mathcal{R}$ by the VM type $k \in \mathcal{VM}$. For each request, the types of VMs are randomly selected and a random number between 0 and 10 is generated to simulate the number of requested VMs of each type. Finally, we randomly generate a value between 1 and 20$ to represent the declared security valuation (bid) in the request.

D.5.2 Results Analysis

To study the performance of the mechanism, we consider multiple scenarios where we vary the allocation problem size by increasing the number of requests and the amounts of available capacities of Cloud resources. Figure D.3 shows the results obtained for different combinations of I and $\rho_r,\ r \in \mathcal{R}$. In Figure D.3a, the social welfare achieved by the two mechanisms in each case is presented. The results shown are the average of fifty runs. We notice that the social welfare obtained by the proposed mechanism is very close to the optimal one achieved by the VCG mechanism. The greedy algorithm usually achieves a good approximation of the optimal solution (to at least 50%) for some specific problems, such as Knapsack problems.
In Figure D.3b, we show the execution time of the two mechanisms on a logarithmic scale since the values for some of the cases are very distanced. As shown, the VCG-based allocation is relatively slow and becomes computationally inefficient in the case of large scale problem instances since it aims at finding the optimal allocation rule, which makes it suitable to perform only in an offline allocation scenario. On the other hand, the greedy-based allocation mechanism is very fast and is able to find a near-optimal solution in polynomial time, which



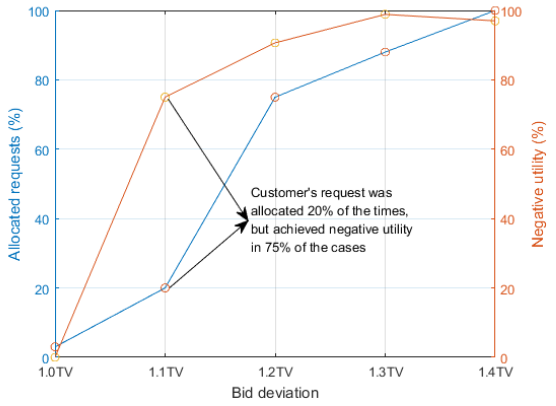
(a) The social welfare achieved by the two mechanisms.

(b) The execution time of the two mechanisms.

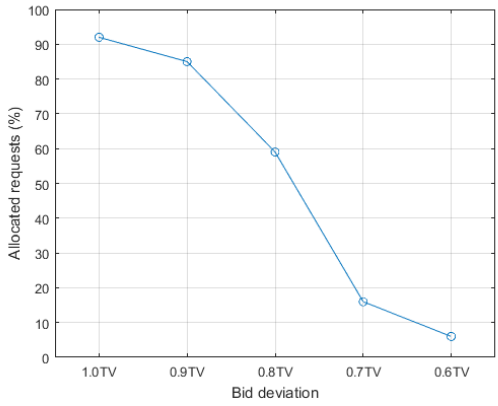
Figure D.3 Evaluation of the performance of the online mechanism with respect to the offline VCG mechanism in five different scenarios : case 1, $I = \rho_r = 5000,\ \forall r \in \mathcal{R}$; case 2, $I = \rho_r = 10000$; case 3, $I = \rho_r = 15000$; case 4, $I = \rho_r = 20000$; case 5, $I = \rho_r = 25000$.

makes it ideal to be implemented in a real-time mode.

To study the incentive-compatibility of the designed mechanism, we consider a total of a hundred cases where a customer $i \in \mathcal{I}$ participates in the auction while varying her reported bid. In Figure D.4a, we show the variation in the request's allocation rate along with the rate of negative utility when the bid value positively deviates from the true valuation of the request, that is, when the customer reports a higher value than her true security valuation in the request with the objective of increasing her chances of getting her request allocated. We denote by TV the true valuation of customer's request. We consider a case where $TV = 10\$$ and the vector of resources requested by the customer is equal to (2,3,4). We notice that when customer i increases her bid by reporting a value $\hat{b}_i$ higher than her true valuation, her chances of getting her request allocated will increase, which is perfectly normal since reporting a higher bid implies a higher value of computed VSI_i, which will move the customer up in the greedy algorithm and increase her chances of getting allocated by the mechanism. On the other hand, this strategy comes with a cost, which is increasing the possibility of achieving a negative utility. We use the achieved utility metric to demonstrate the incentive-compatibility of the mechanism. For instance, when the reported bid was equal to $1.1TV$, the customer achieved an allocation rate of 20%, but in 75% of the cases where the customer got her request allocated, her utility, which is equal to the difference between her true security valuation and her made payment, was negative. This goes back to the fact that customer's payment is computed based on the bids of the other customers, and in some cases, it could be higher than the actual valuation of the customer. This proves that the mechanism guarantees the



(a) Request allocation and negative utility rates when declared bids are higher than the true security valuation.



(b) Allocation rates of customer's request when declared bids are lower than her true security valuation.

Figure D.4 Study of incentive-compatibility of the proposed mechanism based on the bid deviation from the true security valuation.

truthfulness of the customers and gives them incentives to report their true requests, since deviating from the truth will not always achieve a positive utility, hence it is not an optimal strategy.

In Figure D.4b, we show the variation in the allocation rate of customer's request when the reported bid value negatively deviates from the true valuation. We start by defining a setting where the customer is able to obtain the allocation of her request in most of the hundred cases. The value of TV is set to 10$ and the vector of requested resources to (2,2,3). When decreasing the reported bid and performing the allocation, the allocation rate starts to drop, hence following this strategy by the customer will not always guarantee the allocation of her request. We can conclude that the designed mechanism satisfies the incentive-compatibility property, where telling the truth is a dominant strategy for the customers.

D.6 Conclusion

The lack of security and trustworthiness in Cloud Computing is still slowing down its adoption, especially by the organizations and businesses that deal with sensitive data. CIPs are required to increase their investments in security solutions to boost the migration towards the Cloud. However, investing in cybersecurity adds considerable costs to their budgets, and does not necessarily generate higher profit. We proposed in this paper an online mechanism that performs the allocation of CIPs' secure resources in an auction-based fashion. In our allocation model, customers would be required to show their appreciation of the security

added value by including bids in their resource provisioning requests. The mechanism implements an online allocation rule that places customers' requests according to their security valuation, and a truth-inducing payment rule that computes their required payments. This idea allows the CIP to reduce the cost of security investments, allocate their secure resources to the customers who are mostly in need for protection, and be encouraged to increase her investments in the future. Our mechanism showed acceptable performance compared to the offline VCG mechanism. It achieved acceptable approximation of the optimal social welfare and is implementable in real-time. In the future, we plan on identifying the security resources that could be placed for auction and integrate them into the model.