SECURE POLICIES FOR THE DISTRIBUTED VIRTUAL MACHINES IN MOBILE CLOUD COMPUTING

BOUBAKEUR ANNANE

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2020

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

ANNANE BOUBAKEUR

calon untuk Ijazah **PhD**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"SECURE POLICIES FOR THE DISTRIBUTED VIRTUAL MACHINES IN MOBILE CLOUD COMPUTING"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***13 January 2020.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
January 13, 2020.

Pengerusi Viva: **Assoc. Prof. Dr. Siti Sakira Kamaruddin** Tandatangan
(Chairman for VIVA) *(Signature)*

Pemeriksa Luar: **Prof. Dr. Abdul Hanan Abdullah** Tandatangan
(External Examiner) *(Signature)*

Pemeriksa Dalam: **Assoc. Prof. Dr. Nur Haryani Zakaria** Tandatangan
(Internal Examiner) *(Signature)*

Nama Penyelia/Penyelia-penyelia: **Assoc. Prof. Dr. Osman Ghazali** Tandatangan
(Name of Supervisor/Supervisors) *(Signature)*

Nama Penyelia/Penyelia-penyelia: **Assoc. Prof. Dr. Adel Alti** Tandatangan
(Name of Supervisor/Supervisors) *(Signature)*

Tarikh:
(Date) ***January 13, 2020***

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Pengkomputeran Awan Mudahalih (PAM) adalah gabungan pengkomputeran awan dan pengkomputeran mudah alih melalui teknologi tanpa wayar untuk mengatasi batasan sumber peranti mudah alih. Dalam PAM, virtualisasi memainkan peranan utama manakala sumber awan dikongsi di kalangan ramai pengguna untuk membantu mereka mencapai prestasi yang cekap dan mengeksploitasi kapasiti maksimum pelayan awan. Walau bagaimanapun, kekurangan aspek keselamatan menghalang manfaat teknik virtualisasi, di mana pengguna yang berniat jahat boleh melanggar dan merosakkan data sensitif dalam Mesin Maya (MM) yang diedarkan. Oleh itu, kajian ini bertujuan untuk memberi perlindungan terhadap MM yang diedarkan dan data sensitif pengguna mudah alih dari segi keselamatan dan privasi. Kajian ini mencadangkan pendekatan berdasarkan proksi awan yang dikenali sebagai Proxy-3S yang menggabungkan tiga dasar keselamatan untuk MM; kawalan akses pengguna, peruntukan yang selamat, dan komunikasi yang selamat. Proxy-3S memastikan MM diagihkan selamat di pelayan yang berlainan di awan. Ia meningkatkan pemberian kebenaran akses untuk tugas-tugas aplikasi yang diagihkan secara intensif. Tambahan lagi, algoritma yang membolehkan komunikasi yang selamat di kalangan MM yang diedarkan dan perlindungan data sensitif dalam MM di atas awan dicadangkan. Prototaip dilaksanakan pada simulator NetworkCloudSim untuk mengurus keselamatan MM dan kerahsiaan data secara automatik. Beberapa eksperimen telah dijalankan menggunakan aplikasi diedarkan penjagaan kesihatan dunia dari segi kecekapan, liputan dan masa pelaksanaan. Eksperimen menunjukkan bahawa pendekatan yang dicadangkan mencapai kecekapan dan nisbah liputan penyerang yang lebih rendah; sama dengan 0.35 dan 0.41 masing-masing dalam semua konfigurasi berskala berbanding dengan kerja yang sedia ada. Di samping itu, masa pelaksanaan pendekatan yang dicadangkan adalah memuaskan dari 441ms hingga 467ms konfigurasi awan kecil dan besar. Kajian ini bertujuan untuk menyediakan integriti dan kerahsiaan dalam bertukar maklumat sensitif di kalangan pelbagai pihak berkepentingan dalam aplikasi mudah alih yang diedarkan.

Kata kunci: Dasar keselamatan, Pengkomputeran awan mudah alih, Keselamatan virtualisasi, Serangan jarak jauh dan dalam kediaman, Mesin maya yang diedarkan selamat.

Abstract

Mobile Cloud Computing (MCC) is a combination of cloud computing and mobile computing through wireless technology in order to overcome mobile devices' resource limitations. In MCC, virtualization plays a key role whereas the cloud resources are shared among many users to help them achieve an efficient performance and exploiting the maximum capacity of the cloud's servers. However, the lack of security aspect impedes the benefits of virtualization techniques, whereby malicious users can violate and damage sensitive data in distributed Virtual Machines (VMs). Thus, this study aims to provide protection of distributed VMs and mobile user's sensitive data in terms of security and privacy. This study proposes an approach based on cloud proxy known as Proxy-3S that combines three security policies for VMs; user's access control, secure allocation, and secure communication. The Proxy-3S keeps the distributed VMs safe in different servers on the cloud. It enhances the grants access authorization for permitted distributed intensive applications' tasks. Furthermore, an algorithm that enables secure communication among distributed VMs and protection of sensitive data in VMs on the cloud is proposed. A prototype is implemented on a NetworkCloudSim simulator to manage VMs security and data confidentiality automatically. Several experiments were conducted using real-world healthcare distributed application in terms of efficiency, coverage and execution time. The experiments show that the proposed approach achieved lower attacker's efficiency and coverage ratios; equal to 0.35 and 0.41 respectively in all experimented configurations compared with existing works. In addition, the execution time of the proposed approach is satisfactory ranging from 441ms to 467ms of small and large cloud configurations. This study serves to provide integrity and confidentiality in exchanging sensitive information among multi-stakeholder in distributed mobile applications.

Keywords: Security policies, Mobile Cloud Computing, Virtualization security, Remote and co-residency attacks, Secure distributed virtual machines.

Declaration

Some of the works presented in this thesis have been published or submitted as listed below.

[1] Boubakeur Annane, Osman Ghazali, "Virtualization-Based Security Techniques on Mobile Cloud Computing: Research Gaps and Challenges" in Proceedings of International Conference on Future Internet Systems and Applications (ICFISA). 10 & 11 Dec 2018, Kuala lumpur. Malaysia.

[2] Boubakeur Annane, Osman Ghazali, and Adel Alti, "A New Secure Proxy-Based Distributed Virtual Machines Management In Mobile Cloud Computing " in Proceedings of 7th International Conference On Computing & Informatics (ICOCI). 28 March 2019, Bangkok, Thailand.

[3] Boubakeur Annane, Osman Ghazali, "Virtualization-Based Security Techniques on Mobile Cloud Computing: Research Gaps and Challenges" published in International Journal of Interactive Mobile Technologies (iJIM) – Vol. 13, No. 4, April 2019. Indexed Scopus: Q3.

[4] Boubakeur Annane, Osman Ghazali, and Adel Alti, "A New Secure Proxy-Based Distributed Virtual Machines Management In Mobile Cloud Computing" published in International Journal of Advanced Computer Research (IJACR), Vol 9(43). Indexed Scopus: Q4.

[5] Boubakeur Annane, Osman Ghazali, and Adel Alti, "Proxy-3S: A New Security Policies-based Proxy for Efficient Distributed Virtual Machines Management in Mobile Cloud" submitted to Journal of Transactions on Emerging Telecommunications Technologies (ETT). Clarivate Analytics, Indexed ISI and Scopus: Q2.

[6] Boubakeur Annane, Adel Alti, and Osman Ghazali, "SecNetworkCloudSim: An Extensible Simulation Tool for Secure Distributed Mobile Applications" accepted for publication in the International Journal of Communication Networks and Information Security (IJCNIS). Indexed Scopus: Q3.

.

Acknowledgements

In the name of ALLAH, Most Gracious, Most Merciful:
“Work; so Allah will see your work and (so will) His Messenger and the believers;”
____(The Holy Quran - AtTawbah 9:105)

My Deepest thanks and sincere gratitude goes to my supervisors Prof. Madya Dr. Osman Ghazali (School of Computing, Universiti Utara Malaysia) and Associate Prof. Dr. Adel Alti (Department of Computer Science, University Ferhat Abbas Setif-1, Algeria) for their tireless encouragement, wisdom and experience. Prof. Madya Dr. Osman Ghazali provided me with constant guidance and constructive criticism throughout all stages of my research. I will never forget your patience, input and suggestions. I must extend my thanks and gratitude to my co-supervisor Associate Prof. Dr. Adel Alti for his guidance and continuous support during my research and in all my university studies stages (bachelor, Master, and PhD). His wide knowledge of research, logical way of thinking and serious attitude toward research has given me great encouragement and inspiration to accomplish this research. He really showed and shared with me all his experience, research ideas (practical and theory) and motivated me in all my critical times to achieve the completion of my PhD journey. Thanks to both my supervisors, it was my pleasure to study and supervised under your excellency. Without your valuable support, my thesis would not have been possible.

I would like also to express a huge thank to the current and past members of InterNetWorks Research Lab whom I enjoyed working with. Especially my thanks and best regards to the Head of InterNetWorks Research Lab Professor Dr. Suhaidi Hassan and Dr. Yousef Ali Fazea Alnadesh.

My grateful thanks are also extended to the Dean of Awang Had Salleh Prof. Dr. Ku Ruhana Ku Mahamud and Deputy Dean Dr. Nur Haryani Binti Zakaria who support, helped me in my study, and reply all my inquiries.

Additionally, I would like to thank my friends in Setif and Malaysia for their sincere wishes, encouragement and prayers.

Not in the least, many thanks to my beloved Universiti Utara Malaysia for having trust in me to complete PhD journey.

Finally, my heartiest gratitude goes to my family, to my dear father Seghir Annane, to my dear mother Hayat whom always have faith in me and pray for my success, to my brothers Taki Eddine and Idriss, who are willing to extend a helping hand, to my beloved sisters Rima, Hiyam and Wided for their support and love.

Table of Contents

List of Tables

.

List of Figures

List of Abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
AWS	Amazon Web Service
CC	Cloud Computing
CSP	Cloud Service Provider
CPU	Centric Processing Unit
DH	Diffie Hellman key exchange
EC2	Amazon Elastic Compute Cloud
GUI	Graphical User Interface
IDE	Integrated Development Environment
IaaS	Infrastructure as a Service
IP	Internet Protocol
IT	Information Technology
MAC	Media access control
MCC	Mobile Cloud Computing
PaaS	Platform as a Service
PB	Public Key
PV	Private Key
Proxy-3S	Three policies Secure cloud Proxy
PS	Prescriptive Study
QoS	Quality of Service
RAM	Random Access Memory
RC	Research Clarification

S3	Amazon Simple Storage Service
SHA	Secure Hash Algorithms
SaaS	Software as a Service
SLA	Service Level Agreement
TCP	Transmission Control Protocol
VM	Virtual Machine
VMM	Virtual Machine Monitor

CHAPTER ONE

INTRODUCTION

1.1 Overview

Nowadays, mobile cloud computing is considered as an important technology that has grown fast among individual and community of users. It combines cloud computing paradigm with mobile devices through wireless technology in order to avoid the devices' restricts resources capacities and leveraging the cloud computing services offering [1, 2]. The mobile devices such as smartphone and tablets have several limitations in terms of resources capacities such as central processing unit (CPU), memory and storage space which inhibit the developers from providing powerful applications as well as hinder the users to enjoy the various mobile applications in their daily life [3, 4]. Integrating cloud computing services with mobile computing is an interesting solution to solve related issues.

Cloud computing is an attractive technology that is known to have increasing importance for users by delivering services over the Internet. It is defined as an Information Technology (IT) paradigm that allows the user to exploit cloud services in an on-demand way [5, 6]. Three main services are provided: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). IaaS is cloud computing architecture and infrastructure that provides users access to all computing resources in a virtualized environment. Such as servers, storage and networking. PaaS provides computing platform to clients such as programming languages, operating system, web server and others. Moreover, it provides platform development tools for user's developers such as Google App Engine and Microsoft Azure. SaaS offers on-demand pay-per-use of cloud application to users. Whereas, these applications are independent and it is unnecessary to install on personal computer. In addition, cloud computing has many benefits in terms of scalability, data availability, reliability, cost reduction for cloud user [5, 7]. Meanwhile, with the increasing daily use of mobile devices, cloud services are becoming more and more attractive for mobile users.

Mobile cloud computing allows users to move and upload their applications, services, and data on the shared cloud servers for exploiting large storage capacity and high-computing resources when running intensive applications and remote data storage that exhaust the battery life of the mobile devices. Figure 1.1 shows the architecture of the mobile cloud computing environment and the related services leveraged.

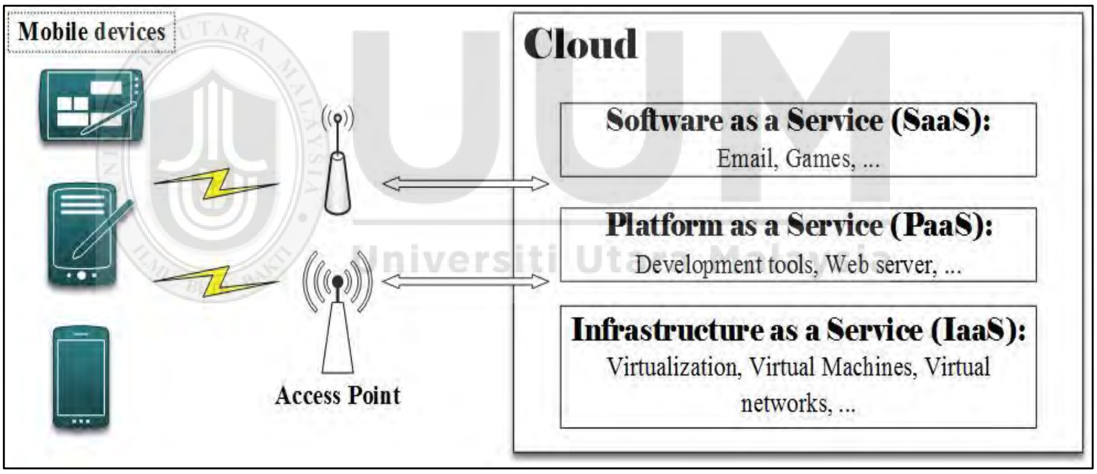

Mobile devices
Access Point
Cloud
Software as a Service (SaaS):
Email, Games, ...
Platform as a Service (PaaS):
Development tools, Web server, ...
Infrastructure as a Service (IaaS):
Virtualization, Virtual Machines, Virtual networks, ...

Figure 1.1. Overview of Mobile Cloud Computing Environment

Recently, the use of the mobile devices is not only retained for simple applications but also complex and crucial applications which deal with a sensitive data with various multimedia contents (e.g. texts, images, audios and videos), also banking application, health, transport, and others. Applications, services, and data are accessed, delivered and used over the Internet, instead of mobile devices, and are paid by cloud customer or provider [8]. The moving of clients' services and data to the cloud technology raise many security challenges especially data security and privacy protection that become major and serious concerns because data is located in different distributed places.

Data security and privacy are considered as the most barriers that hinder the widespread emergence of the cloud. Several and various commercial business and organizations still unwilling to move their data and resource computing management to the third-party Cloud Service Providers (CSP) [9]. Usually, the migration of the user's application services/data is occurred by the help of the virtualization layer by moving out the data from the mobile device to the virtual machines for processing on the remote cloud.

Various users' virtual machines are running on the same cloud host when they share the same cloud resources, which lead to more additional security risks like violating the data once they share the same Memory or CPU [10, 11]. Thus, an important question to be highlighted is whether the other cloud virtual machines' clients are trusted or not. Currently, data security and privacy issues have been studied by many researchers for ensuring the tenant of data confidentiality and integrity of the cloud providers. However, most of the solutions proposed are not practical due to the critical change in the cloud platform which eliminate side channels and removing such clocks as well as the hypervisor [12].

In this chapter, we present the problem statement of data/services security and privacy of the mobile user using the cloud computing services on the virtualized environment, also outlines the research questions and objectives. Then we give the scope of the research, including the contributions and significance of the study. Finally, the chapter ends with a summary and provides an idea about the next chapter.

1.2 Research Motivation

Recently, mobile computing is known as a fast-growing utilization of people's daily life. However, the main challenge that faced this rapid advancement is the constrained and limited mobile devices' resources. The mobile computing might be improved by integrating with cloud computing which leads to a new technology called Mobile Cloud Computing (MCC) [13, 14].

MCC is presented as services offered for the mobile computing environment in order to avail the cloud services characteristics such as broad network access, elasticity, on-demand self-service and resource pooling (e.g., cloud providers serve multiple tenants with their needs). Although the MCC has several advantages for both cloud service providers and mobile users, it also challenged by many critical issues such as security and privacy of the mobile user's data offloaded on the cloud [13, 15].

Security is considered a major challenge for MCC environment. The mobile cloud security issues are inherited from cloud computing, so they are the same issues but also more critical on MCC because of incapacity and limited devices' resources (e.g., lack of CPU capability) to process intensive malware application or complex algorithm to protect the sensitive data as like personal computers. There is a crucial need for a lightweight framework that guarantees security with minimum processing and communications overhead on mobile devices [16]. The tenants' worries are concentrated on the migration to the cloud, which might be faced with additional risks once they are sharing the same cloud resources with other tenants [13]. In MCC, the cloud service providers offer the sharing of their resources to mobile users through one of the popular technique called virtualization that increases the efficiency and effectiveness [12].

Thin virtual machine (or phone clone) is a virtual machine (VM) which is pre-installed once the mobile user migrates their services or data to the cloud for storage and processing. More precisely, the VMs are used to increase the utilization rate of cloud provider platform and to manage the maintenance of application computing more easily. However, researchers in [17] showed that virtualization had brought several security threats and issues. Many serious and various attacks have been illustrated in [18] that affect the virtualized systems such as Denial-of-Service (DoS) attacks when this kind of attack hits insipid information like workload statistics to know whether the system is vulnerable or not. We restrict our attention in this work to wide critical common attacks such as co-resident (co-location) attacks, distributed attacks, and the hypervisor attacks which affect the VMs or known as Thin VMs and violate its sensitive information. Moreover, talking to co-resident VMs (i.e.: VMs running on the same physical server) are logically isolated from each other, but attackers can build many side channels to avoid the isolation and retrieve sensitive data from legal VMS (legal mobile user). Researchers in [17] have shown that an attacker can reach 40% of efficiency, which means that an attacker when spread 10 VMs attacker, 4 of them can co-locate with target VMs (victim).

Many solutions have been proposed to solve these security issues. Firstly, eliminating the side-channel attacks against cloud system is one of solution which is proposed by [19, 20] to mitigate the risks of VMs side channel. Also, removing and adding some components for the hardware layer has been proposed by [21, 22, 23]. However, all the proposed methods have common drawbacks that require high deployment cost in terms of high execution time and high computation complexity for implementing and not immediately adopted under the continuous changing of cloud platforms [12]. This is the first reason that motivates us to find a low-cost solution-based software implementation and its immediate practical realization and not hardware.

Secondly, some works presented by [24, 25] consist of increasing the difficulties to establish co-resident attacks using the network-based measurement (e.g., the same VMs IP addresses are considered co-resident). In these attacks, the hypervisor is the target to get the IP address of VMs. Such solutions can be broken and not a simple IP address hiding can protect victim VMs (i.e., the attackers do not rely only on VMs' IP addresses) [12]. Moreover, in [26, 27] the authors have presented techniques consists of detecting the malicious VMs attacks (abnormalities features happen for CPU and RAM once the attackers want to retrieve sensitive information from VMs). Migrating the VMs from a host to another host on the cloud is a proposed solution by [28, 29], but the solution will lead to more power consumption as well as the quality of service degradation such as Service Level Agreement (SLA) between cloud's client and the cloud service provider. Using VM allocation policies is one of our interests which can cause difficulties for establishing co-resident attacks. Such policies have been only proposed in [12, 30].

Our main motivation in this study is solving the data security and privacy issue. More precisely, protecting the user sensitive data in terms of integrity and confidentiality on MCC environment as well as particularly better exploiting virtualization-based techniques. The specific motivations that encourage us to do research in this area are based on two main reasons.

First, most of the above described related works provide mechanisms for dealing with some common attacks and hardware-based mechanisms which is costly to be adopted due to high-computation time with algorithm complexity that needs of changing the current cloud platform. Secondly, some of the proposed mechanisms such as [12] worked on virtual machine allocation policy to defend against well-known attacks which are called co-residency attacks. This policy is one of our interests in this work due to the important role that can play once allocate the VMs. However, all the above-proposed works present some weaknesses in terms of robustness against attacks that target the VMs deploying on the different distributed host (physical server) and when VMs deploying on the same mobile devices or on distributed on different mobile devices.

The main aim of the work is to reach a high level of robustness to protect the data integrity and confidentiality in unsecured networks against different attacks (co-resident (co-location) attacks, distributed attacks, and the hypervisor attacks). Finally, we would like to solve the most security and data privacy side of the mobile user VMs ranging from the mobile device till offload to the cloud and until received by the authorized receiver. We will introduce an efficient proxy-based security technique that protects the distributed mobile application from being attacked once any transaction happens. We need to ensure whole data security and privacy protection of sensitive data of the mobile client application via a secure proxy.

1.3 Problem statement

Virtualization is a very promising mechanism in cloud computing that increases the efficiency of exploiting shared hardware resources such as memory, cache, and CPU of servers. The virtualization is defined in the Infrastructure as a Service (IaaS), where many data centers contain various servers which deploying Virtual Machines (VMs) that comprise huge data amount of users. An image of VMs of mobile devices also called phone clones are pre-created on these servers while offloading and performing mobile user’s intensive applications and tasks [13]. Similarly, unauthorized users can deploy their VMs and obtain data from the legitimate user by constructing many malicious side channels using the same sharing resources (CPU cache, memory bus).

The main goal of virtualization is to run different virtual machines of different mobile users at the same time or simultaneously. Thanks to the Virtual Machine Manager (VMM) so-called Hypervisor that ensures the management (e.g., creating, deleting, and migrating) of different VMs (phone-clones) and the isolation from each other. However, the hypervisor vulnerabilities can be exploited by an adversary to obtain access to users' virtual machines [18, 31].

Frequently, the VM of different mobile users executed on the same physical host are logically isolated from each other. However, malevolent users can escape the logical isolation while sharing the same resources (CPU, memory, and cache) and capture sensitive and private information like crypto keys from co-location virtual machines [17, 32]. Some proposed solutions [22, 23, 33] attempt to tackle this type of threat “VM to VM attacks” by ignoring the side channel constructed between co-location VMs which is not allowed by the cloud policies [23]. Moreover, the suggested frameworks demand major changes to be implemented in the existing cloud commercial platform. Consequently, the proposed methods are impractical due to high deployment cost (i.e.: high execution time and high computation complexity).

The previous studies show that the attackers need to co-locate their malicious virtual machines with the VMs target on the host cloud before they would be able to make their side channels to violate any useful information [12, 17, 25, 26]. Thus, using the VM allocation policy is one of the crucial factors that cloud providers can control and influence the possibility of co-location [12, 30]. So, researchers in [12] have attempted to solve the problem by finding a robust and secure virtual machine allocation policy that increases difficulties for attackers to co-locate malicious VMs with their targets and mitigates the possibility to perform co-location. However, the proposed works focused on the impact of how many VMs' attacker needs to be launched by malicious users to co-locate with the target legal VMs. This is one of main reasons that allowing start and deploy a limited number of users' VMs in the cloud's servers as well as reduce the co-resident attacks. This kind of solution may enhance the security protection of deployed VMs, but effectively will affect the quality of cloud service provider (i.e.: scalability) which decreases the Service Level Agreement (SLA) between a cloud provider and a user. Moreover, the proposed works also studied strategies for co-locating attackers' under different VM allocation policies. However, if the VM's attacker can success to co-locate with legal VM then VM attacker built malicious side-channel and get data from target VMs. Therefore, it is preferable to come up with an efficient approach that ensures the VMs protection even if the VMs co-location occurs in the cloud's servers.

The aim of this thesis is to tackle the hypervisor and VMs threats by combining the VM allocation policy with the hypervisor protection policy to guarantee both the phone clone integrity and hypervisor integrity. Indeed, we believe that such hybrid-policy would provide large protection against co-resident VM and VMM attacks.

The VMs communicate with each other to exchange private and sensitive information (e.g., distrusted application executed in the different host). We studied the limitations of the existing virtualization security co-location techniques proposed in the literature such as the work of researchers in [12]. We have identified that the main limitation is the absence of protecting sensitive information exchanged between mobile application's tasks deployed on different VMs on the cloud (i.e.: there is no mechanism that protects the data from being stolen while interacting between the VMs). For example, in Figure 1.2 the VM 1 in host 1 communicates with VM 1 in host 2 to exchange information which can lead the attacker to steal the private data exchanged between them. Hence, the only solution which was proposed by [34] is a hardware-based technique with a high-cost barrier. Thus, we aim to design a new solution referred as a proxy for secure distributed VMs on mobile cloud computing not only ensuring the privacy and confidentiality of sensitive data exchanged among multi-VMs data but also reducing the cost in terms of security management time and computation complexity (e.g. fast security management time, low computation complexity). Thus, we believe that our study is going to provide protection for the mobile users' information against different attacks (e.g.: hypervisor attacks, co-location attacks and distributed attacks). Moreover, in the evaluation, we are going to use benchmark data to compare our work and validate it according to two metrics for measuring the attacks: coverage and efficiency that mentioned by authors in [5].

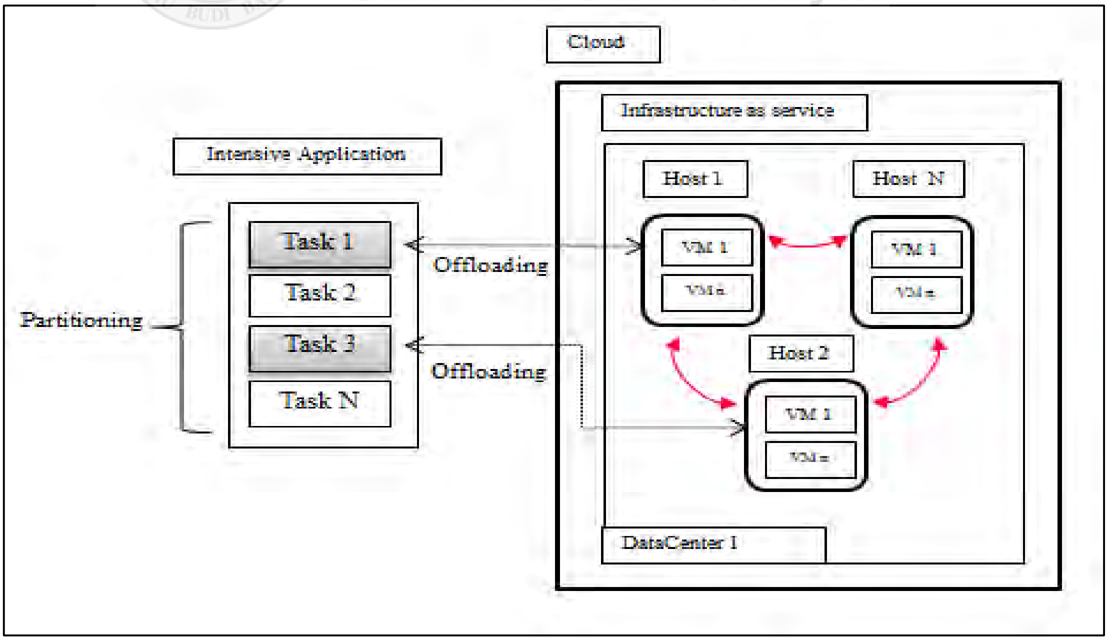


Figure 1.2. Communication of intensive application's tasks while deployed on Thin Virtual Machines

1.4 Research Questions

The main question of this research is how to mitigate the VM Co-resident and the hypervisor attacks, also the attacks on remote VMs located on a different host. Other secondary questions for this research are willing to be addressed:

1. How to define the identity of the remote client and how to manage the privacy and confidentiality of VMs allocation requests of the mobile client on the cloud?
2. What are the methods that can be exploited for ensuring both virtual machine integrity and hypervisor security?
3. How to protect the exchanged information between the VMs (distributed application) deployed in different host on the cloud side?

1.5 Research Objectives

The main objective of this research is to enhance the security of users' sensitive information in the mobile cloud computing environment exactly on the virtualization layer. A new secure approach is proposed named Proxy 3-S, which combines three main secure policies. In order to achieve this major goal, the following objectives are proposed:

1. To design a mobile user control access policy for preventing both unauthorized access to the cloud service provider and preventing the spread of malevolent users' VMs.
2. To design a secure VMs manager policy which protects the VMs allocation on the cloud hosts as well as protects the hypervisor from getting retrieved by unauthorized malicious VMs.

3. To design a VMs communication policy on the cloud using three hierarchical trust levels that guaranty the privacy and the confidentiality of sensitive information exchanged between VMs.

1.6 Research Scope

Cloud computing contains four main deployment modes: public could, private cloud, community cloud and hybrid cloud. The security in the private cloud is highly preserved compared to the public cloud [35, 36]. Our scope starts with focusing on the public cloud as an interesting point of our work. Then, we highlight the data security and privacy and we discuss as follows:

Security and privacy in MCC include many scopes of research that motivate researchers to work on them. It has many sectors inside such as data security, partitioning and offloading security, mobile cloud application security, mobile device security, data privacy, location privacy, identity privacy. For us, our scope of the presented work particularly focused on virtualization-based security and mobile distributed application security. Figure 1.3 shows the scope of our research work.

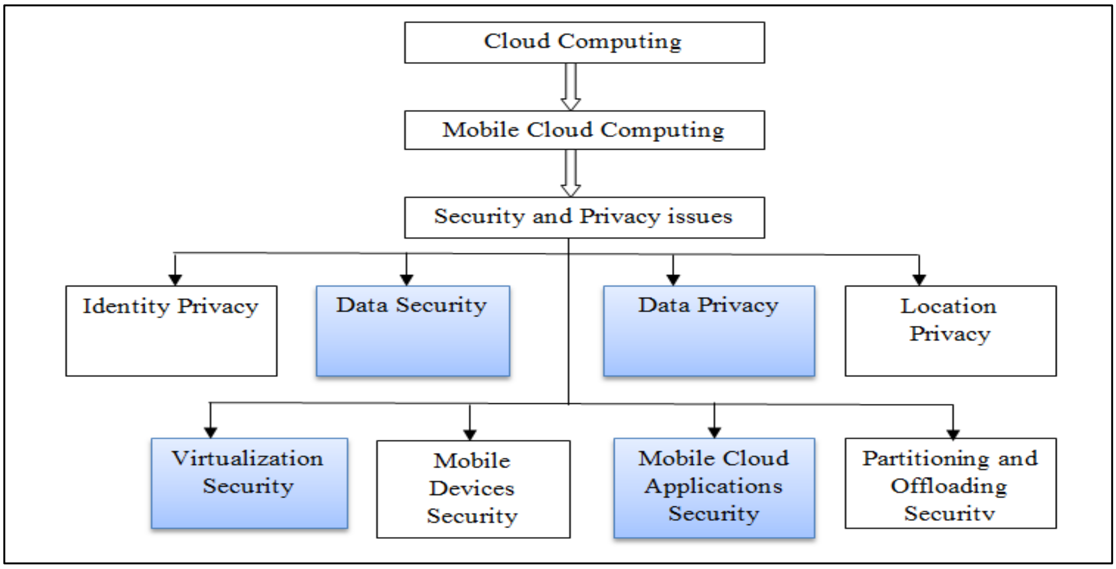


Figure 1.3. Scope of research – Blue Area

For ensuring the security of VMs data of distributed mobile application tasks which deployed on both the same and different host on the cloud is considered as a challenging issue. Several related works [37, 38, 39, 12, 40] have been proposed with many mechanisms for providing security and privacy of target virtual machines' information from the malicious attackers.

We have mentioned two main approaches: (1)-the approaches that concentrated on the attack against VMs Co-location on the same host of the cloud and (2)-the approaches that concentrated on the attacks against information transferred among cloud VMs. Despite this, these solutions have provided many benefits; they still need to protect information exchanged between different cloud VMs deployed on different hosts. Thus, the main scope of our work is to enhance existing approaches with strong security mechanisms of VMs shared distributed information deployed whether on the same or different hosts on the cloud. Furthermore, the level of our work is issued on the security of VMs data in infrastructure service, while both security on services (software/platform) of the cloud model will be planned for future directions works. Moreover, it must also be mentioned that the scope of this research is limited to VMs integrity while deployed on either the same or different cloud's hosts. In particular, we ensure the integrity and confidentiality of the distributed applications' data that process inside VMs. While the security of the cloud services model (software and platform) will be planned for future directions works.

1.7 Research Contribution

Our research serves to connect multi-stakeholder distributed mobile applications (i.e.: different departments in a hospital, distributed banking branches) securely and save the integrity and confidentiality of exchanged sensitive and important information (e.g., the financial information, health records).

The major contributions of this research are:

1- The proposed user control access policy aims for preventing the access of illegitimate mobile users that would leverage the cloud service provider and allocate their malicious virtual machines.

2- The proposed secure VMs manager policy aims to guaranty both the hypervisor security and thin VMs integrity by preventing the allocation of malicious VMs on the cloud host.

3- The secure VMs communication policy for ensuring the secure exchanges of sensitive data among VMs deployed on different cloud's hosts.

4- The extended security layers on NetworkCloudSim architecture aims to model the security of distributed tasks deployed on different hosts and simulate VMs based security intensive tasks scenarios.

1.8 Significance of the Study

The benefit of this research is to enhance the security level for sharing sensitive data on distributed mobile applications by describing the privacy level of exchanged data between different virtual machines on the cloud and helps us to adapt the well-known security VM-allocations policies to model a realistic system.

This research proposes multiple policies in a mobile cloud computing environment. The primary concern of these policies is to enhance the security of infrastructure service for a mobile user that leveraging from the cloud computing resources. This would be achieved by reducing the efficiency and coverage of the attacks against the virtual machine, which contains sensitive data of the mobile user. More precisely, virtual machines or phone clones can be distributed on many hosts in order to balance the workload between the servers on the different datacenter. The virtual machines are allocated, whether on the same or different hosts can be stolen by retrieving their data from another virtual machine attacker. These virtual machines are managed by a component called either hypervisor or virtual machine manager.

The detailed significance of our study is to securely allocate the mobile users' virtual machines by controlling and prohibiting the access of malicious users (i.e.: attackers) to the cloud services. Moreover, defending the hypervisor from the attacks which have as target the authorized virtual machines (i.e.: legitimate mobile user) and also the co-resident VMs hosted on the same host. Furthermore, ensuring the data user from being retrieved when it transfers from the mobile application to the virtual machine hosted on the cloud environment and also protecting the exchange of data between different mobile application tasks deployed on the cloud host. We believe such proposed policies would prevent mobile users from being attacked and lost their sensitive data or information.

1.9 Organization of the Thesis

This thesis has been organized into six chapters. The resume of each chapter is provided as follows:

Chapter One gives an overview of the research interest. It also includes the research motivation that persuades us to contribute to this type of research concept. The problem statement of the research is also introduced in this chapter as well as both main and secondary research questions. The chapter also states the research objectives and the scope that restricts and specifies the limit of our work. Concluding the chapter, the main research contribution and the significance of research have been mentioned as well.

Chapter Two provides a literature review of the area of the research. It covers the general literature about the important concepts of mobile cloud computing, security threats, offloading and virtualization technology issues. Moreover, the chapter illustrates the virtualization attacks roots that explain how the attackers can achieve their malicious goals. Furthermore, the chapter presents a section that reviews and compares the previous solutions presented to tackle the research problem and gives the strengths and weaknesses of each mechanism and policy. Moreover, the chapter discusses the cloud simulator tools and gives a details comparison between them in order to well understanding both the advantages and limitations of each tool. Finally, the chapter reviews the Diffie–Hellman algorithm and the various Hash functions as well as the main reason for using Hash-Diffie Hellman technique in this research.

Chapter Three addresses the research methodology used to conduct this work. The chapter starts by introducing the research phases and the conceptual model of the proposed approach. Furthermore, the selecting tool for conducting the simulation experiments and evaluation will be provided and illustrated in detail. The chapter defines the security performance metrics being using in the evaluation and validation of the proposed approach. A conclusion of this chapter is included that opens the door to chapter four.

Chapter Four presents the secure cloud proxy named Three policies Secure cloud Proxy (Proxy-3S) proposed in this research work. This chapter provides in detail three secure policies for controlling the VMs' user access to the cloud services and preventing the leakage of the sensitive data that is processed inside VMs on a mobile cloud environment. The chapter discusses the co-resident attacks and security metrics for measures the performances of the proposed secure proxy. Further, our improved security metrics are given in order to evaluate the distributed attacks (communicating VMs attack). The problem definition stating how to reduce the attackers' coverage and efficiency without reducing the number of users that intend to leverage the cloud services is provided, including the security actors modeling. In addition, a technical section for explaining the proxy usage of Hash-Diffie Hellman encryption and decryption process is presented. Finally, the chapter ends with details algorithms of the proposed three secure policies.

Chapter Five provides the proposed SecNetworkCloudSim: an extensible simulation tool based on NetworkCloudSim for securing the distributed mobile application over the cloud. The SecNetworkCloudSim is a secure, mobile and open-source simulation tool that preserves high confidentiality access to the shared data hosted on a mobile device and over distributed cloud's servers. Moreover, a section illustrating the diagram class and simulation execution workflow of the novel proposed tool is presented in this chapter.

Chapter Six presents the approach's implementation and evaluation using an intensive distributed healthcare mobile application. The chapter provides various comparisons between our approach results and the results of the related works in order to prove the effectiveness and performance of the proposed secure approach. In addition, analysis and evaluation of the SecNetworkCloudSim compared to NetworkCloudSim is briefly discussed in its dedicated section. The chapter ends by presenting a security comparison of the proposed mobile user access control mechanism and other related works as well as describing the general drawbacks and comparison of research works' security degree.

Chapter Seven concludes this thesis by stating the research summary along with research contributions, limitations, then suggestions and future directions.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The main objective of this chapter is to present a comprehensive literature review and background of the previous works regarding the research area of security in the mobile cloud computing environment.

This chapter is organized as follows. Section 2.2 presents the concept of mobile cloud computing and its services. It provides security and privacy challenges in mobile cloud computing. Further, it gives a definition of virtualization, its benefits and security issues regarding cloud computing. Section 2.3 provides a clear explanation of the utility of using the concept of proxy. Section 2.4 presents the virtualization attacks classification and details about the attacks on the virtualized system for well understanding the topic related security issues. Section 2.5 details the different approaches and techniques proposed to tackle the virtualization attacks and a critical review of the existing frameworks. Section 2.6 provides a comparative study and gives the strength and weaknesses of each solution. Further, this chapter presents a critical review of the cloud simulators in Section 2.7 and provides a comparison regarding the advantages and disadvantages of each tool. Section 2.8 reviews the Diffie Hellman algorithm and Hash functions proposed in the literature. Finally, Section 2.9 concludes the chapter.

2.2 Mobile Cloud Computing

Nowadays, mobile devices are considered essential facilities in our life. People are dramatically using mobile devices in their daily life. In 2014, the number of mobile users was growing faster from 5.6 billion up to 6.2 million in 2018 and the number of users' also still significantly increased, which means 84% of the population will use smartphones and tablets [41]. Many applications whether for entertainment or work running on mobile devices need power processing which exhausts the battery life. More precisely, intensive mobile applications like augmented reality, anti-virus, video editor and face detection need high-power CPU, memory, and high-storage capacity to efficiently perform application tasks or to manage the data. However, mobile devices are constrained by their limited resources and cannot be able to proceed with huge computing like desktop computers [42].

New technology has started to emerge. The main idea comes to integrate mobile technology into cloud computing to resolve the problems related to mobile devices which appear a new powerful technology called Mobile Cloud Computing (MCC) [43]. To give a clear definition of the mobile cloud, it should before understand the Cloud computing model.

Cloud computing is a paradigm that provides services like computing, software, and storage on-demand manner instead of the product [44]. This means that computing provided as utility or service for the end-user and the resources are always available for clients once they need (availability). International companies such as Amazon, Microsoft and Google have been invested in this technology by providing their own strong, reliable and cost-efficiency cloud platform for users, where services are made to appear anytime and everywhere with the pay as you use fashion which offers more benefits whether for small or big companies to consume the remote computing and resources data storage on-demand in the datacenters.

2.2.1 Mobile Cloud Computing Concept

Turning to mobile cloud computing, many definitions can be presented to illustrate this new paradigm. MCC is a combination of mobile computing and cloud computing [3] where mobile devices take the benefits from cloud resources using a set of techniques in order to leave out their constraints and getting the mobile devices more resistible in terms of power consumption such as extending the battery life. Other mentioned definitions have described that mobile cloud is the concept that refers to an infrastructure where the data computing and the data storage moved outside the mobile device which means on the cloud [15]. Figure 2.1 presents the architecture of mobile cloud computing.

There are three main services in cloud computing, the first one is Software as a Service (SaaS) which delivers the applications as a service for the client or the end-user over the internet [3, 45]. Such kinds of these applications: DropBox, Gmail, Microsoft Office 356, Rackspace, Salesforce and SAP Business ByDesign. The second service is Platform as a Service (PaaS) which gives the opportunity for developing applications in a platform using Application Programming Interfaces (API). Google App Engine, Microsoft Azure, and Amazon web services are the primary players known in this layer. The third main service in the cloud environment is Infrastructure as a Service (IaaS), this layer contains the hardware resources such as datacenters which provides storage and computation facilities using the virtualization for sharing the computing resources such as CPU, Memory of the cloud servers. Flexiscale [46], Amazon EC2 [47] and Amazon S3 [48] are examples of IaaS service providers.

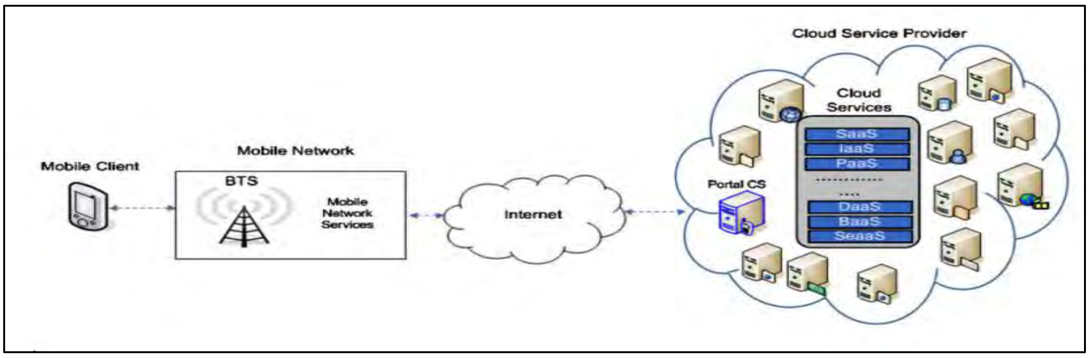
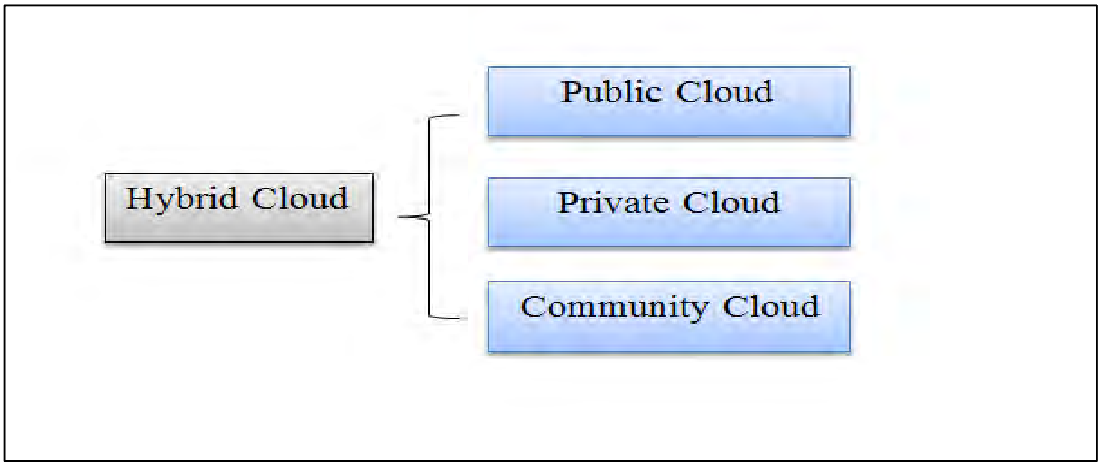


Figure 2.1. Architecture of Mobile Cloud Computing (Adopted From [16])

2.2.2 Cloud Computing Deployment Mode and Pertaining Security

Cloud computing is consisting of four main types of deployment modes named: public clouds, private clouds, hybrid clouds, and community clouds [44]. Each type has its own characteristics. Figure 2.2 shows the different cloud model deployment. For the public cloud, the cloud providers offer the resources (Network, Servers, Application) as a service for a general organization or individual for free or certain small chargeable amount. The disadvantage of the public cloud is that data of the tenants (costumers) are not under control, which leads to more security issues [4, 49].


Hybrid Cloud

Public Cloud

Private Cloud

Community Cloud

Figure 2.2. Cloud Computing Deployment Models

The second type is a private cloud. The private clouds are implemented only for being used by one enterprise or organization. The infrastructure of this type of cloud can be managed either by an external provider or the organization itself. Further, the security, reliability, and performance (Quality of Service) are higher than the cloud public deployment model type [4, 49, 11]. The third type is a hybrid cloud. The hybrid cloud is the combination of the two types of cloud models or three types, which aim to cover the limitation of each model. For instance, in infrastructure service, some of the parts are processed in cloud private such as sensitive information, where other parts are performed on the cloud public. These benefits provide more flexibility for the organizations and guarantee strong security and control over the tenant's data [4, 49]. The last cloud model is community cloud, wherein this model type, the infrastructure service is shared between various tenants and organizations which targeting a specific concern such as security requirement and compliance regards [11, 50].

2.2.3 Offloading

The execution of mobile applications is considered as computational intensive tasks that consumed large energy of mobile devices. Indeed, this kind of challenge has been defeated by the offloading technique [51]. The tasks and the computational intensive application are transferred to the cloud (remote server nodes) for processing and the results back to the mobile terminals afterward [52].

The use of remote servers leads to leverage the huge processing capacity also extends the battery life by saving the energy [53]. Figure 2.3 illustrates an example of the steps of the offloading process "partitioning, migration and execution" from the mobile terminals to the remote cloud.

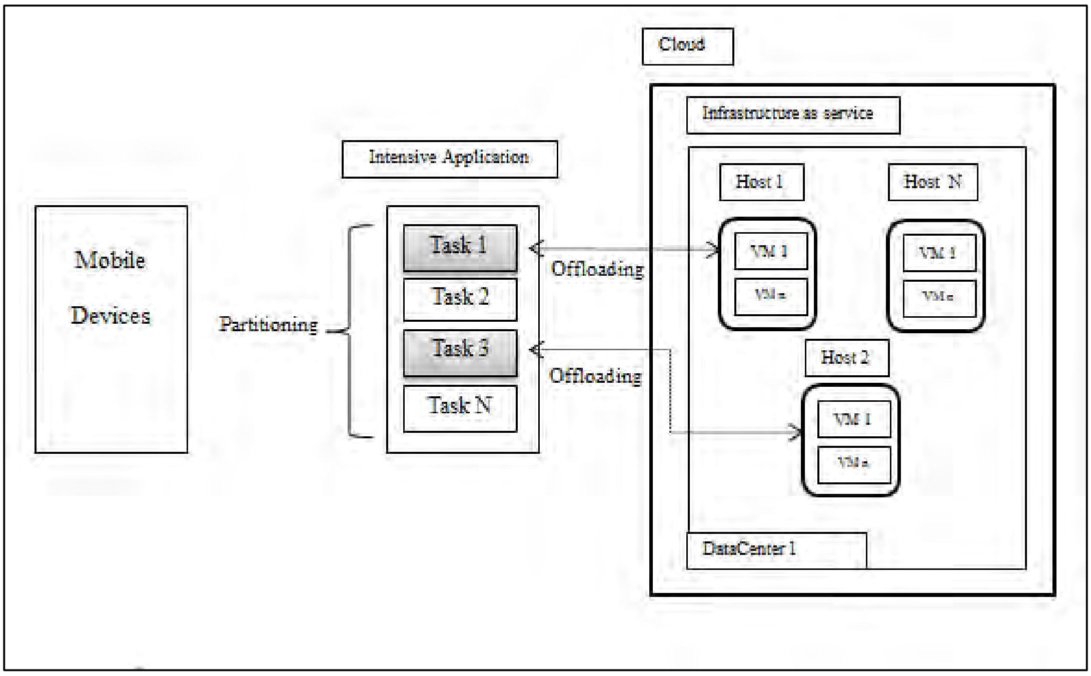


Figure 2.3. Partitioning and Offloading of Mobile Application to the Cloud

Before the mobile intensive application outsourced from the mobile device to the cloud, it was divided into many parts (or tasks) as presented in Figure 2.3. Some tasks are still executed on the mobile device due to the necessity of using local resources such as cameras, positioning and location system, and other sensors. Moreover, the tasks which have lower resources consumption can be performed in the mobile device. Otherwise, highly intensive resource tasks are migrated to high computing capacity (cloud) [54].

2.2.4 An Overview of Mobile Cloud Computing Challenges

Both mobile devices consumers and cloud service providers have taken advantage of the mobile cloud computing environment. However, the MCC stills face different challenges that hinder it and make it more difficult compared to Cloud Computing. In this section, we give a short brief about the challenges that have faced by the mobile cloud before we introduce our related works.

1- **Mobile devices resources limitations:** mobile devices still face various limitations: storage capacity, processing power, and battery power compared with a desktop computer. Even though there have been improvements in different aspects of mobile devices such as CPU, memory and battery life, they are incapable to run the power-intensive application in their local physical resources [55, 13].

2- **Heterogeneity:** in the environment of mobile cloud, various mobile application services are interacted and running on different processor architectures and operating systems, and communicating through various protocols and communication supports. This may affect the quality of service like application response time, communication quality, and service delivery [14, 13].

3- **Elasticity:** similar to cloud computing, elasticity and scalability are the main needed factors in MCC services. The cloud services provider needs to meet and satisfy all the mobile user requirements when they are over available resources. The interruption of services due to resource unavailability cause many problems between the end-user and cloud providers [13].

4- **Applications services issues**: the limited resources of mobile devices prevent the intensive task to be freely deployed and executed. However, the offloading technique needs to be applied for migrating the computationally intensive task from the device to the cloud environment [56, 57]. The most intensive task is running on the cloud server and a small part of the computational processing is executed in the mobile device. Consequently, the mobile user may face delay which affects negatively the quality of service [13, 58].

5- **Security and privacy challenges:** Compared to cloud computing, security and privacy issues are increasing in MCC environment [59]. Therefore, running intensive applications over vast distances against malware within mobile devices are very complicated due to the constrained resources. Thus, executing complex algorithms is inconvenient as like normal computer. For instance, various intensive applications will be communicating over vast distances, the need for secure communications is critical; otherwise, sensitive data and information would be put at risk. Also, communications and mobility should not be tracked; otherwise, it would violate privacy [14, 16].

2.2.5 Security and Privacy Requirements in MCC

United State National defense have defined the general security and privacy requirements for MCC, which are mentioned in the following [13].

1- **Confidentiality:** confidentiality is referring to keep the user's data secret and safe in the cloud and it considers as one main security and privacy requirement [60]. Accordingly, mobile users have risks once avail the cloud services. As the data is transmitted and received within a public network, also executed and stored in public cloud datacenters, there is a possibility of retrieving the data by unauthorized or malevolent users.

2- **Integrity:** the integrity is ensuring the data consistency and accuracy related to users in the cloud side once is stored on the service providers. Whereas, the alteration of sensitive data is prohibited by unauthorized users and it leads to various users' losses such us their business [61].

3- **Availability:** ensuring the availability for mobile users means that all cloud services must be always available for users at any time and everywhere according to mobile user's needs and their usage contexts [62]. Ensuring the availability includes prohibiting the different type of attacks which destabilize the availability of services.

4- **Access control and authentication:** authentication is the operation of identification of user correct identity [63, 64]. After the process of authentication is successful, it is necessary to identify the resources to which they have access and what type of execution can execute by the mobile user, such as viewing, editing, or deleting. These restricted operations called control access [65].

5- **Privacy:** Privacy is ensured directly or indirectly while the requirements stated above are checked. Confidentiality, integrity, and authentication are three needed objectives that preserve the privacy of the cloud service of mobile users [66, 67].

2.2.6 Security and Privacy Issues in Mobile Cloud Computing

Mobile cloud computing uses many techniques such as offloading, partitioning, virtualization, and mobile cloud-based application, outsourced storage in order to serve and process mobile users [68, 69]. However, these techniques which have various benefits for mobile devices, lead to several new security challenges that inherited from the cloud computing security's drawbacks that affect the mobile user on many sides [70, 71].

There are various defiances which have been discussed above. Security and privacy defiances are becoming more critical than other challenges due to many reasons such as remote distributed cloud processing and storage where the sensitive data resided on the cloud, the user data transmission of over network with the heterogeneous environment through various protocols, communication technologies, also the limited resources of the mobile devices [72]. Figure 2.4 depicts the main security and privacy challenges within the MCC. The next section will detail our issue work.

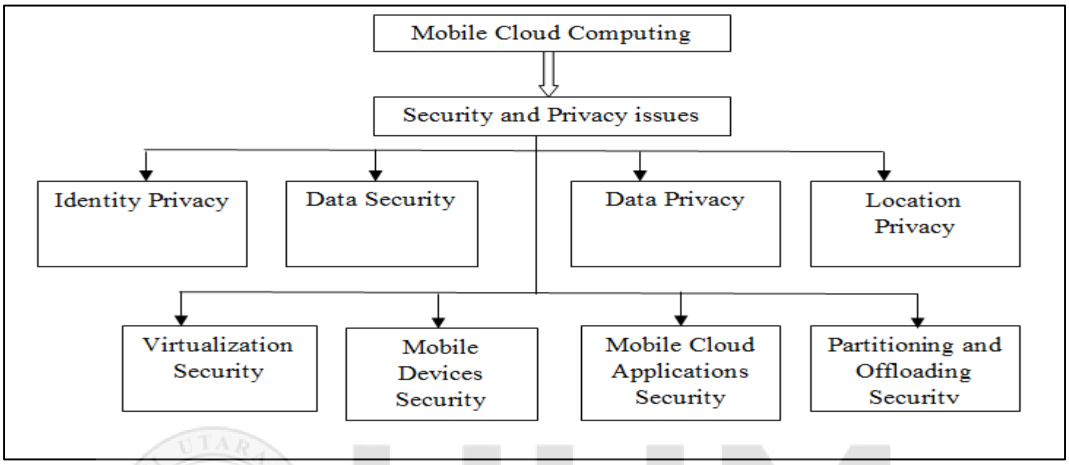


Figure 2.4. Security and Privacy Issues in Mobile Cloud Computing

2.2.7 Virtualization Security on MCC

In MCC, cloud services are provided for mobile users using virtualization technologies [73, 74]. IT research organization (InfoTech) considers that the distributed host on different datacenters leverages only 20% of the full capacity without virtualization. The virtualization process can increase hardware utilization (efficiency) between 60% and 80% [75]. The virtualization is defined as a middle layer between the software and hardware layers in the cloud servers that allows the cloud provider to efficiently exploit their services and computing resources [18]. These resources can be shared among multiple virtual machines in order to run them simultaneously (at the same time) and share also benefits from available servers' resources (e.g. CPU, network bandwidth, Memory, etc.) [76]. Figure 2.5 shows the virtualization layer in the cloud computing environment.

In the cloud end, once the mobile task is offloaded, an image of virtual machine of the mobile device (called also phone clone) is pre-installed for processing the mobile user's data and application which augment the efficiency of the cloud environment and decrease the maintenance overhead on the mobile devices [13, 77, 78]. Therefore, running the phone clones of mobile devices on the same server and isolate them is the main responsibility of the virtualization technology [79, 80].

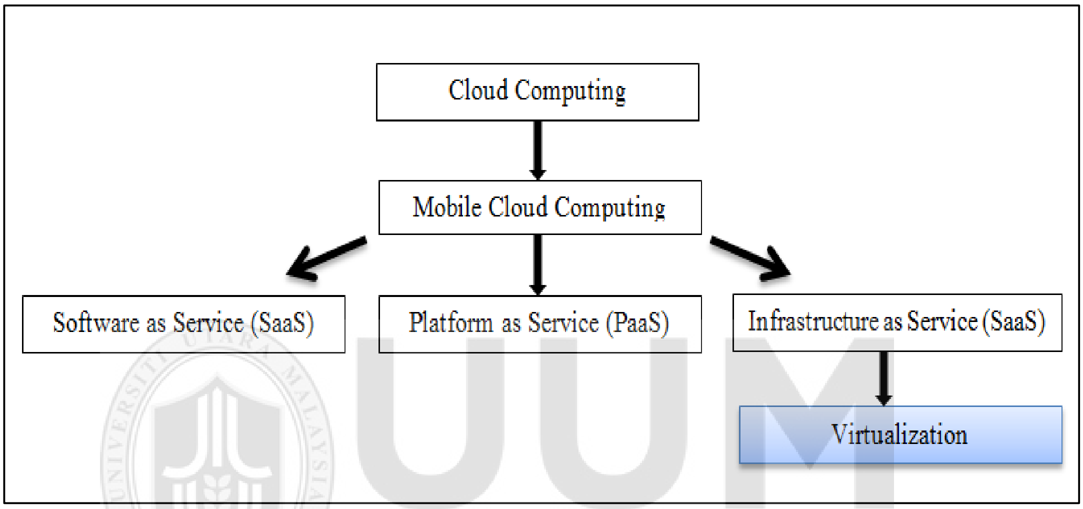

Cloud Computing
Mobile Cloud Computing
Software as Service (SaaS)
Platform as Service (PaaS)
Infrastructure as Service (SaaS)
Virtualization

Figure 2.5. Virtualization layer in the cloud environment

Cloud computing has commonly used virtualization and leveraged from virtual machines mechanisms. For the cloud client, virtual machines help to tear out the maintenance of computing resources from the client device itself and enabling scalability of resources (enough to accept any added functionality at any given time). For the cloud providers, virtual machine increases the effectiveness and the efficiency of the hardware's utilization rate [12, 81]. However these benefits, virtualization technique when applied on MCC, brings new security risks such as unauthorized access from malicious VMs, VMs to VMs attacks, the confidentiality of mobile users data, challenges within VM monitor and communication in a virtualized environment [18, 82]. Hence, ensuring security mechanism that prevents leakage of sensitive data and information from legitimate phone clones is not an easy task. Many researchers have undertaken to develop frameworks, policies, and approaches against this kind of challenge to ensure the security aspects for the mobile users. These methods are mainly focused on how to ignore the side channels attacks between VMs while the malicious VMs access the cloud servers [22, 23, 21, 33]. However, all the methods proposed need fundamental changes to the current commercial platform and they are not practical and not immediately deployed [83].

2.3 Concept of Proxy and Utility

In general, a proxy is considered as a software or computer system that works like an intermediary for requests from users would leverage the resources from other servers [84, 85]. The users connect to the proxy and requesting some services such as file, web page, connection, or other available resources from various servers [86]. Meanwhile, the proxy evaluates the requests in order to control and simplify their complexity as well as provide the anonymity and facilitating the access to the servers ‘contents [87, 88]. Figure 2.6 shows the communication between two entities through a proxy.

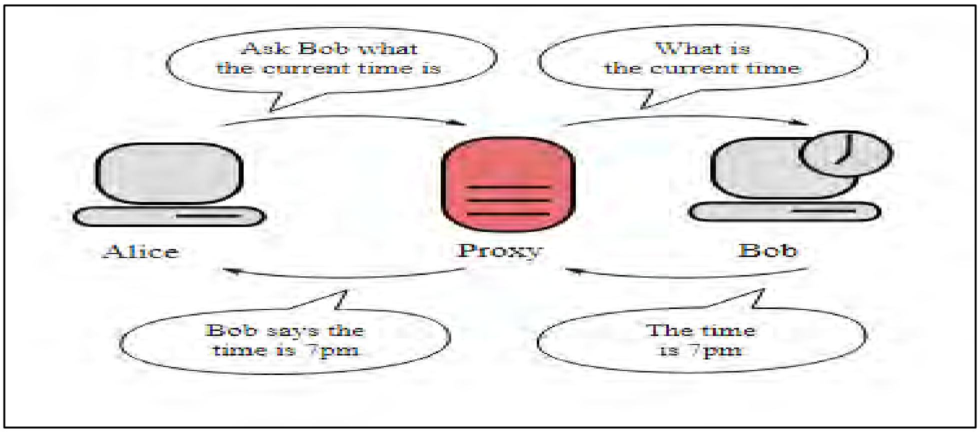


Figure 2.6. Communication between two entities through a proxy

In security aspects, the proxies are used for protecting against different operating system attacks and Web server attacks which enhance the security level for sharing and accessing the resources [89, 90, 91]. In Figure 2.6, an example of two computers communicating with each other through a third proxy which helps to guaranty privacy (Bob does not know whether the information is going to Alice or another one). So, the proxy takes the responsibility of providing the information to the legitimate party.

The reason for using a proxy in our work because of the need for an advanced approach based on proxy's architecture in order to monitor, control and secure the sensitive data of users' virtual machines on the mobile cloud computing environment. We can use another security approaches such as encryption/decryption [92, 93] and secure broker based trust [94, 95], but we crucially need a separate module which can manage the security aspects and filter the virtual machines as well as blocks the attackers who would intercept or retrieve the user information on the cloud' servers. Moreover, we keep the function of allocating the virtual machines to the hypervisor layer which consider as a complex task.

2.4 Virtualized System Attacks Classification

The virtualization has various benefits for the MCC environment. It also has many drawbacks that introduce new security concerns [96, 97]. Among these concerns are two crucial attacks (i.e.: resources sharing between thin VM and hypervisor attacks) that would be handled by the presented work. The taxonomy of the general threats in the virtualized system must be provided to clarify which threat model that we would tackle.

There are five main types of threats that can attack four layers of the virtualized environment. The first lower-layer dedicated to introduce the hardware attacks. In this kind of threat, the attackers try directly to access the memory for modifying or reading the virtual machine monitor space and violate the content of the legitimate VM [18, 98]. The second layer is the Hypervisor layer, this type of threat is mainly used to gain the full control of thin virtual machines by exploiting the vulnerabilities of the hypervisor [18, 99]. The third layer is the Virtualization layer, when the attackers exploit the virtualization benefits to retrieve sensitive data from VMs inside the same host cloud (co-resident attacks) [18, 100]. The fourth layer is Kernel layer, these attacks are destined to operating system to get the full control of the system [18, 101, 102]. The last layer is the Application layer, where the attackers inject malicious code into the tenant's application for executing it [13, 22, 103, 104].

The second and third layer is the interest of this thesis. The attacks on the virtualization layer which include co-resident threats have been studied by several researchers and try to tackle them by changing the architecture of the current cloud platform [20, 22, 23, 33, 105]. However, these solutions are not immediately practical. It has been shown that co-resident attacks can achieve a high efficiency percentage ranging by 40 % or higher [17], which means half of the attackers' VM numbers can co-locate with legitimate VMs. There are two types of co-resident attacks that have whether a specific goal or not [12]. Hence, the attackers who have not a specific target, their objective is to get disturbed cloud platform capacity and resources by providing unfair share among the virtual machines.

2.4.1 Attacks Roots in Virtualized System

In this section, we give the possible attacks in a virtualized environment. Our research work is constrained by two threats of co-resident attacks and hypervisor attacks, which are the essential parts of our research work. We consider these assumptions when the cloud provider is trusted, untrusted (malicious provider), trusted cloud provider with inside attacks or trusted provider but inquisitive. Figure 2.7 shows the possible roots attacks on the virtualized environment.

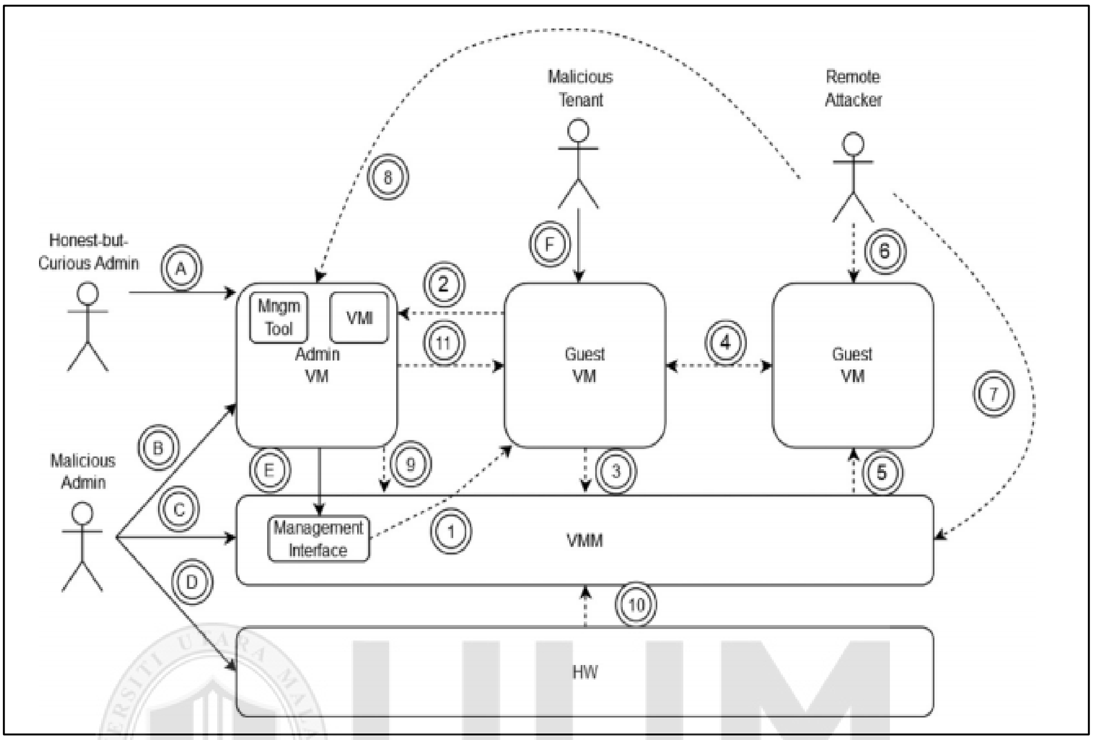


Figure 2.7. Attacks roots in a virtualized environment (adopted from [18])

Below are the main attacks on the virtualized environment:

1- **Honest but curious provider (Semi-trustworthy):** the curious providers may explore some data which not necessary to be shown for them. For example service providers of healthcare service, when the provider can collect the patient activity information, is not necessary to know the patient identity [5, 106]. So, in this kind of threats the provider can read only and access the Admin Virtual machine [107].

2- **Malicious provider:** Contrary to the previous threat, a malicious cloud provider can read and modify the administrator virtual machine status which reveals sensitive information and give access to the virtual machine monitor [18, 108, 109].

3- **Admin VM:** An administrator VM attacks may affect the virtual machine monitor over its management interface or the thin virtual machine directly. Therefore, such kind of attacks leads to lose the control on the virtualization system by interrupting the execution of the authorized VMs (affect availability) or modifying the VMs memory [18, 110].

4- **VMM:** The attacks from virtual machine monitor enable unauthorized access to the thin virtual machines status and produce several risks such as modifying the memory, processing the state of VMs, stop and running VMs, read sensitive data [111]. These attacks affect respectively the integrity, confidentiality and the availability of the victim VMs [18, 112].

5- **Tenants cloud (costumers):** migrating to the cloud solutions means the tenants are exposed to extra security risks due to the shared resources between them. So, a malicious tenant can attacks and get full access to the virtual machine [18, 113].

6- **Guest virtual machine:** a malicious guest virtual machine can get access to administrator virtual machines due to cross VM attacks or its vulnerabilities. The ports of Admin VM can be open only for Hypervisor's virtual private network. Moreover, a guest virtual machine can attack and gain access to the VMM by exploiting its vulnerabilities as well [18, 114, 115]. Another virtualization threat can happen is when a malicious virtual machine attacks another virtual machine to retrieve sensitive data (co-residency attacks) [116].

2.5 Related Security Techniques for Virtualization Challenges

This section will provide the main proposed work by researchers to tackle the issues related to the virtualized environment in either cloud or mobile cloud computing:

1. Security-Aware Provisioning and Migration Scheme (SWAP)

Researchers in [70] have proposed a security scheme called Security-aware Provisioning and Migration Scheme (SWAP) for provisioning and migrating thin virtual machine or phone clone for preventing the covert channel attacks. This kind of attack is a constructed link between VMs where the CPU cache and the memory bus are exploited to steal information from legitimate phone clones in a virtualized environment [17]. The proposed scheme includes two techniques. The first one is the provisioning of new phone clones where this technique works with mobile communication history to avoid users' phone clones to host with other strangers' thin virtual machines. The second technique is responsible for migrating the phone clones from one host to another when the threats of attacks increase.

The proposed solution has successfully kept little risks on phone clones compared with other proposed researchers algorithms. The performance gains of optimization allocation method that is similar when the cloud system has enough capacity to allocate phone clones in safety and avoiding risky allocation, but when the number of phone clones becomes larger, the proposed algorithm provides better migration mechanisms using optimal phone clones. The main shortcoming of the proposed approach is that researchers of this work are assumed that two phone clones in the same host do not attack each other when they have a link of communication between them. This work does not consider different scenarios (co-resident attacks, hypervisor attacks) which can happen and produce several security threats on phone clones.

SWAP Algorithms: the issue is related on how to reduce the risks of the covert channel when cloud provider does not have enough resources to isolate the foreigner's phones clones on the mobile cloud environment [70]. The authors have presented three algorithms in this proposed schema. The first and second algorithm provides the steps to allocate the phones clones on the specifics host by implemented various condition. The algorithm based on the communication history between the mobile users when the virtual machines represent node and communication between virtual machines represent the edges (communication graph). Thus, both algorithms allocate the phone clones based on communication history between mobile users. The phone clones that have a communication link would not attack each other and may allocate in the same host.

The third algorithm shows the migration steps of VMs users (phone clones). The migration means moving the phone clone from a host to another one in order to prevent the risks of retrieving the content of the phone clones from another adversary [70].

2. Secure Mobile Cloud Platform (SMOC)

Researchers in [34] have presented a platform called Secure Mobile Cloud Platform (SMOC) for securing the mobile cloud environment. The platform allows the user to run their applications whether on the cloud or on the mobile devices itself. In contrast, the proposed platform includes two concepts. The first one is sharing resource which means that a mobile application can freely change the running location for getting better user experience, not explicitly (obligatory) on the cloud. This design provides more flexibility compared to other proposed approaches and techniques. The second concept is ensuring security even though the operating system of the mobile device has been attacked. A thin virtual machine shares its information and files with the mobile device once the mobile application running in the cloud. Conversely, the mobile device shares its files and information (devices inputs/outputs) with VM-cloud for running the application.

Despite the benefits of this platform towards security concerns, there are many assumptions are not considered by the researchers such as untrusted hypervisor and untrusted public cloud provider. The proposed platform ensures the data security and privacy only when both assumptions (cloud provider and hypervisor) are assured. Thus, this platform ensures security towards the untrusted guest operating system which runs also unsecure applications.

3. Providing User Security Guarantees in Public Infrastructure Clouds

The authors of this work have proposed a framework for data and transaction security of infrastructure services [37]. The proposed framework contains various protocols for trust and the storage protection operation called Domain-Based Storage Protection (DBSP) and other protocols for trusting the virtual machines deployment called Trust-Lunching (TL). Trust VMs made a more suitable method that the virtual machines are running inside a trusted host using secure computing techniques.

The proposed work realized several security analyses against attacks and the obtained results improve the robustness and the efficiency of this framework. For more details, before the deployment of guest VMs, the protocol of trust VMs is performed. The second protocol is used cryptography techniques outside the IaaS domain for ensuring the data confidentiality stored in the cloud. Hence, authors have presented a list of malicious host attacks against IaaS environment to produce both secure protocols. The proposed framework can be integrated into the existing cloud platform due to numerous experiments and tests which have been realized on tenants' sensitive data (e.g.: public healthcare patients' data). However, this framework has been only considering specified attacks on IaaS platform which not guarantee for other security threats such as hypervisor attacks, co-resident attacks and so on.

4. An Approach to Defend Against Co-Resident Attacks in Cloud Computing

Security is most prevalent in cloud services. Authors in [12] proposed an approach to minimize the co-resident attacks in a cloud environment. This work improved VM allocation policy, which makes difficulties for attackers to achieve their goals by preventing them to co-locate the unauthorized VMs with a legitimate one. It made up a new allocation policy security technique that includes workload balance and power consumption. Before this work, authors have proposed a PSSF tool (Previous Selected server first) [83] focusing on security problems like co-residency.

The authors extended this tool with three policies: security, power consumption, and load balance to enhance the effectiveness and efficiency of the cloud platform environment. Moreover, they provide three security-based metrics (coverage, efficiency, and VMmin) in order to evaluate an allocation policy as safe or not. It applied several experiments on the simulation cloud platform CloudSim and the obtained results show remarkable robustness against co-resident attacks. However, the authors have only studied one type of attack: a co-resident attack occurred only inside one host and not distributed on different hosts. They also did not consider the migration attacks, which can make up large serious risks on VMs and increase the possibility of co-locating with VM victims.

5. Hardware-Assisted Secure Virtual Machines (H-SVM) under a Vulnerable Hypervisor

A new design for hardware based VM protection has proposed by [39]. The approach is called H-SVM, which is a Hardware-Assisted Secure Virtual Machine. The proposed mechanism protects the guest virtual machine for monitoring the malicious VM or hypervisor by isolating its memory virtualization. The authors have proposed a new flexible and efficient mechanism of memory protection by allowing restricted roles for the hypervisors and decoupling the memory isolation from memory allocation that is usually executed by the hypervisor. Therefore, the handler (processor) takes some roles of hypervisor such as scheduling VMs. The mechanism of changing the hardware architecture presents drawbacks regarding the deployment cost and also maybe not suitable for immediate deployment.

6. Security Isolation Approach for Virtual Machines Deployment

Authors in [117] have proposed an approach for securing the VMs deployment. This approach reinforces the isolation among the virtual machines and controls the availability of resources by using a security system mechanism called the Mandatory Access Control (MAC). The use of the MAC controls the access of one process to another. They used the hypervisor which is running on the server operating system to secure the isolation of guest VMs. Also, they implemented a secure channel for migrating the VMs whenever the risks threat becomes higher. Despite the benefits of the solution proposed, authors have assumed that the hypervisor is trusted and other studies proved that several crucial attacks may come from the untrusted hypervisor. Therefore, the authors do not give an evaluation that makes their solution more understandable.

7. Previously Selected Servers First (PSSF)

The Previously Selected Servers First Policy is proposed by [83] namely as PSSF. The aim of this policy is to defend against co-resident attacks. However, the proposed algorithm lack of different security enhancement such as securing the mobile device VMs residing on it, live migration, securing the communication data between the VMs while located on mobile devices and the Cloud. The algorithm also lacking from distributed application security algorithm deployed whether on mobile and cloud. Also, this algorithm targeting only the co-resident attacks and it does not target the other attacks like distributed VMs attacks communication, hypervisor attacks, and mobile device data attacks (Mobile Application). The hypervisor is responsible for isolating the VMs. Therefore, if the hypervisor is attacked successfully, the attacker can break down the isolation between VMs and the VMs will be retrieved by their data.

8. Co-Location-Resistant Cloud (CLR)

Authors in [30] have proposed a placement algorithm named Co-location resistant (CLR) which protects the VMs against two kinds of co-location attacks. The co-location attacks are divided into two sub-attacks in the public cloud. The first attack is complete *co-location* whereas the adversary aims to co-locate with all users VMs. Otherwise, on the fractional attack which is the second kind of attack. The adversary targets some of VMs and not all VMs. The objective of this approach is to deploy the legitimate VMs in such a manner that Adversary VMs cannot co-locate with user VMs.

The algorithm is targeting the optimization as well as the security aspects. However, it needs more improvement related to the isolation of the VMs once they executed in the cloud environment. Furthermore, the algorithm has not considered the risks when the VMs are communicating with each other and transferring the data which can affect the deployed task of application to be run correctly as well affect the integrity and confidentially of user data deployed on VMs.

9. VMs Co-residency Attack (VCDS)

Authors in [25] have presented a novel schema for detecting the VMs Co-residency attacks referred as VCDS. The covert side channel is a kind of attack when the isolation between the VMs is broken which leads to steal sensitive information from the users. The schema is aimed to detect the VMs co-resident by getting the location of the particular VM. To clarify more, the simple way to know whether two VMs are on the host is to rely on network metrics by performing TCP Traceroute steps to get the IP address of Hypervisor. If two hypervisor IP addresses are the same that means corresponding VMs are on the same host (Co-resident). The advantage of this solution is increasing the difficulties to establish co-location. However, the attacker can use another technique and ways to steal information from legitimate user VMs. Thus, not a simple hiding of the IP address of hypervisor can be efficient to solve the co-residency issues.

10. Improving Cloud Survivability through Dependency based Virtual Machine Placement

The authors have presented an approach of virtual machines placement on the cloud for defending against two types of attacks: co-resident attack and hypervisor attack [118]. The security of the VMs that sharing the same resources is depending not only on operating system and application they are running, but also the security of the virtual machines manager and VMs located on the same server. The approach employs Discrete Time Markov Chain (DTMC) to analyze any security threats on VMs. The approach deal with security risks of VMs by migrating them periodically from a host to another one in order to find out a placement algorithm for safely prevent VMs from being retrieved. However, this kind of solution leads to extra power consumption and may reduce the performance of the service offered by the cloud provider for the clients, which may break the Service Level Agreement.

11. Dynamic Secure Interconnection (DSI)

One novel mechanism was proposed by [119] which solved the security of users' data being processed by the shared and virtualized platform on the cloud environment named Dynamic Secure Interconnection (DSI). DSI isolates the cloud environment into several trust virtual zones where the users can securely deploy their tasks. A virtual zone hosted various VMs together on the same costumer's group in the cloud. As the VM is hosted using the same costumer's group, the VM is considered trusted. This mechanism helps to protect the VM's information security when VM migrating from one network to another. However, the migration solution causes extra bandwidth consumption overhead that degrades the service level agreement between a customer and a cloud service provider.

12. Cloudradar: A real-time side-channel attack detection system in clouds

The authors [120] have presented an approach called CloudRadar to detect and mitigate cache-side channels attacks in clouds in real time. The approach uses detection technique for co-resident VMs by verifying any abnormal behaviors while the VMs share caches. As it provides a reliable signature-based detection and continuous monitoring technique to identify the VMs that process cryptographic application. The proposed system achieved high accuracy detection with a lower performance overhead. However, it still limited as it does not cover other threats (e.g. VMs communications, hypervisor attacks and other side channels attacks), and often lacks of processing time integration on cloud applications.

13. MIGRATE: Towards a Lightweight Moving-Target Defense Against Cloud Side-Channels

In order to alleviate the co-location of VMs attacker on the same cloud server, [121] provide the users' VMs live migration for the side channels attacks minimization on virtualized environment. The proposed approach called MIGRATE, allows the lightweight migration of the tenants' VMs. It faced challenges that raising the high infrastructure's overheads by avoiding the full migration of the VMs. The mechanism evaluation showed a lower resource consumption compared to other VMs migration techniques. Nevertheless, VMs migration is not an efficient solution against side-channel attacks due to the VM's downtimes while moving from a host to another, which increase overhead challenges that may impact response time of users' requested services from cloud provider.

14. Cloud Aid: A Cloud Computing Tool for Mitigating Side-Channel Attacks

Authors in [122] proposed a technique to mitigate the security threats of side attacks inside the cloud environment. A malicious VM can co-locate with legitimate users' VM on the same host and generate side channel attacks inter-VMs for leaking and stealing the sharing of private information. The proposed technique is aiming to reduce the cache based side channels attack using a novel prevention tool called Cloud Aid. The tool has two-way protection. Firstly, it prevents the co-location of the tenants' VMs with the attacker VMs before side-channel attacks. Secondly, it hides (encrypts) the data of the users that become much harder for an attacker to recover the sensitive data. The proposed work provides a secure approach without any VM migration techniques and ensures fewer infrastructures overhead. However, this approach still needs more validation in the real experiments.

15. Secure and Efficient Enhanced Sharing of Data Over Cloud Using Attribute Based Encryption with Hash Functions

Authors in [67] provide a cryptographic technique based on hash functions and asymmetric key encryption to protect the integrity and confidentiality of user data stored in the cloud. Performance evaluations of the encryption /decryption time and key generation time are compared with other existing methods. The experiments showed that hash functions and asymmetric key encryption achieve better results than the existing Attribute-based Encryption (ABE) algorithms in terms of less encryption time, computing key time and decryption time. However, the proposed technique needs more real experiments to show the performance in preventing the attacks generated from virtualized environment.

16. Hardware-Assisted Secure Resource Accounting under a Vulnerable Hypervisor

Many research studies have shown that malicious tenants can circumvent the hypervisor and gain full control of the cloud platform. The hypervisor is responsible to allocate the resources for each user's VMs that intend to leverage the server cloud resources like CPU, memory and cache. Many efforts have been invested in the users' VMs protection against vulnerable hypervisor. The most prominent approach was proposed by [123] and utilizes hardware-based technique called Hardware Assisted Resource (HRA) for a secure cloud's resources under an unprotected Virtual Machine Monitor (VMM). Although, their mechanism makes it difficult for cloud users to check out the memory and CPU allocation size in which malevolent users perfectly steal resources. The unprotected hypervisor can arbitrarily allocate a huge amount of resources to malicious VMs, which affect the resources availability, and it will then decrease the Service Level Agreement (SLA) between the tenants and cloud providers.

2.6 Comparative study of related works

All of the above-described related works provide mechanisms for dealing with some common attacks and computation complexity of data security and privacy on the MCC. Table 2.1 illustrates the comparison process of different solutions regarding several attacks.

Table 2.1
Related works' comparison

Proposed Solution	Problem issues	Techniques Used	Simulation tools	Comparison techniques	Evaluated Parameters	Strengths	Drawbacks
SWAP [70]	- VMs co-location - side channel attacks	- Model-based users' communication relationships and potential risks detection when co-locating phone clones. - Minimizing potential risks based on clique-covering technique. - Migration strategy based on a decay function time varying feature of covert channel.	Testbed: Nodobo and Reality Mining Datasets	Benchmark	- VMs' integrity, - Cost and load balancing.	The proposed solution has successfully minimized the risk of phone clones' attacks compared with naïve provisioning and migration algorithm.	- Notable Absence of VMs co-location in the same host. Authors do not assume that two phone clones in the same host can attack each other when the communication link between each other get established
SMOC [34]	-VMs' sensitive data attack	Virtualization - based technique on security mobile	Testbed: Real mobile device and	A study base	VMs' sensitive data Integrity.	- The security of mobile device operating system is	- Researchers assume that both hypervisor and

Table 2.1 continued

	- Untrusted applications leveraging	device hypervisor (input/output proxy) to protect any sensitive user information.	Hypervisor named as KVM			ensured even when the latter is attacked. - The freely execution location of mobile application whether	cloud are always secure, so if the adversary succeeds in penetrating them, then the data of the user would be in risks - Dynamic migration of mobile application within VMs on the cloud.
DBSP [37]		- Protocol-based for trusted VM launch - Encryption-based technique to protect the stored data by a trusted third-party	Test-bed : Health record datasets	A study base	VMs' data confidentiality and operations security	Ensure the confidentiality of sensitive data and information of the user.	The framework handles specified attacks on IaaS platform that not guarantee for other security threats such as attacks on network communication, data geo-localization.

Table 2.1 continued

An approach to defend against co-resident attacks in cloud computing [12]	VM co-resident (co-location)	VMs allocation policy based on coverage and efficiency security metrics.	CloudSim and OpenStack Cloud platform	Benchmar: Three types of VMs allocation policies are compared.	- VM's integrity - Workload balancing - Power consumption	The approach includes several policies such as workload balancing policy and power consumption policy.	- Co-residency attacks: the authors have only studied one type of attack (a co-resident attack) which means the attacks that happen only inside one host not on distributed hosts. - Migration of VMs: they do not consider the migration attacks which can happen due to the vulnerabilities of migration algorithm
H-SVM [39]	Malicious VMs under vulnerable hypervisor	VMs allocation policy based on security metrics: coverage and efficiency.	Testbed: real implementat-ion	Benchmark	VMs' integrity	The policy protects the VMs allocated on the same host against co-located malicious VMs.	- The policy studied only co-resident attacks, but didn't address the attacks which performed on the interaction between distributed VMs

Table 2.1 continued

							- The policy does not secure the mobile user data while moving into the cloud. - The absence of data isolation while VMs communicate between each other (such tasks of distributed application) - Decrease the efficiency and coverage attacks only on the host.
MAC [117]	VMs' isolation	Linux kernel's Mandatory Access Control (MAC) – based mechanism for securing VMs.	Testbed: real implementation	Benchmark	VMs' integrity	- Securing the deployed VMs via an isolating technique. - Online VMs migration by introduce a secure channel among them.	A compromised or untrusted hypervisor will lead unguaranteed secure status of VMs.

Table 2.1 continued

PSSF [83]	VMs co-residency	-Virtual machine allocation policy based on security metrics: coverage and efficiency.	CloudSim	Benchmark : Three types of VMs allocation policies are compared.	VM's integrity	The policy protects the VMs allocated on the same host against co-located malicious VMs	- The policy studied only co-resident attacks, but didn't address the attacks which performed on the interaction between distributed VMs - The policy does not secure the mobile user data while moving into the cloud - The absence of data isolation while VMs communicate between each other (such as tasks of distributed application) - Decrease the efficiency and coverage attacks only on the host

Table 2.1 continued

CLR [30]	VMs co-residency	-Secure VMs placement algorithm - Cryptography-based Technique	Solution based theoretical approach - simulation not used	No comparison technique	VMs’ data integrity	Ensure the protection of VMs against fractional and complete co-location	- Strong isolation between user VMs running on the cloud - Secure communication between VMs once distributed application tasks communicate with each other resources to run the content of VMs.
VCDS [25]	VMs co-residency	Network measurement- based security Technique	Testbed: real cloud platform and KVM hypervisor	Benchmark	VMs’ security	Increasing the difficulties for malicious users to achieve co-location on the user VMs	- Simple hiding of the Hypervisor IP address is not sufficient and the adversary can use another way to steal information from VMs.
DTMC [118]	Co-resident attack and hypervisor attack	- Migration VMs-based Technique - Discrete Time Markov Chain	Not mentioned	A study base	VMs’ security	Protect user VMs whenever the possibility of co-location risks become	- A power Consumption continuously increasing while

Table 2.1 continued

						higher by performing periodic migration from host to host.	migrating the corresponding VMs - May reduce the service performance offered for client which can break down the service agreement. - Migration is not always practical if the other hosts do not have enough resources to run the content of VMs.
DSI [119]	Malicious VMs	- Isolation of cloud environment into several trust virtual zones - Classify the same costumers VMs in same group in order to be among VMs' trust collection.	Test-bed: real cloud platform and hypervisor	A study base	VMs' integrity	- This mechanism helps to protect the VM's information security when VM migrating from one network to another	- The migration solution causes extra bandwidth consumption overhead that degrades the Service Level Agreement (SLA) between a customer and a cloud service provider

Table 2.1 continued

		- Migration policy performed if threats detected on VMs.					
Cloudradar [120]	VMs co-location And cashe-side channel attack	- The approach uses detection technique for co-resident VMs by verifying any abnormal behaviors while the VMs share caches - Signature-based detection and continuous monitoring technique to identify the VMs that process cryptographic application	OpenStack platform	Benchmark : using other encryption methods	VMs' integrity	- The approach achieved high accuracy detection to mitigate cache-side channels attacks in clouds in real time with lower performance overhead	- The approach does not cover other threats (e.g. VMs communications, hypervisor attacks and other side channels attacks), - lack of processing time integration on cloud applications
MIGRATE [121]	VMs co-location	Users' VMs live migration to minimize the side channels attacks on virtualized environment	Test-bed: VMware and V-Sphere cloud	Benchmark	VMs' integrity	The mechanism evaluation showed a lower resource consumption compared to other VMs migration	Impact on response time for services that users require from the cloud provider because the VM is moving from host to

Table 2.1 continued

						techniques.	another host.
Cloud Aid [122]	Malicious VMs and cashe- side channel	- Prevention the co-location of the tenants' VMs with the attacker VMs before side-channel attacks happen - Encrypt the data of ~~the~~ users which become much harder for an attacker to recover the sensitive data	Testbed: OpenStack Newton	A study base	VMs' sensitive data integrity.	The proposed work provides a secure approach without any VM migration techniques and ensures fewer infrastructures overhead	The approach still needs more validation on the real experiments
ABE [67]	Data breach in cloud environment	Attribute based encryption method based on hash function and asymmetric key encryption	CloudSim	Benchmark : Compared to existing ABE algorithm	Data integrity and confidentiality.	- The proposed technique achieves better results than the existing Attribute-based Encryption (ABE) algorithms in term of less encryption time, computing key time and decryption time.	Need more realistic experiments in preventing attacks generated from virtualized environment.

Table 2.1 continued

HRA [123]	Unprotected hypervisor threats	Use of hardware-based technique called Hardware Assisted Resource (HRA) for secure the cloud's resources under unprotected Virtual Machine Monitor (VMM)	Testbed: Xen hypervisor	Benchmark	VMs and hypervisor integrity.	The mechanism makes difficulties for cloud users to check out the memory and CPU allocation size in which malevolent users perfectly can steal resources.	The unprotected hypervisor still can arbitrarily allocate a huge amount of resources to malicious VMs. which negatively affect the resources availability, and decrease the Service Level Agreement (SLA) between the tenants and cloud providers.

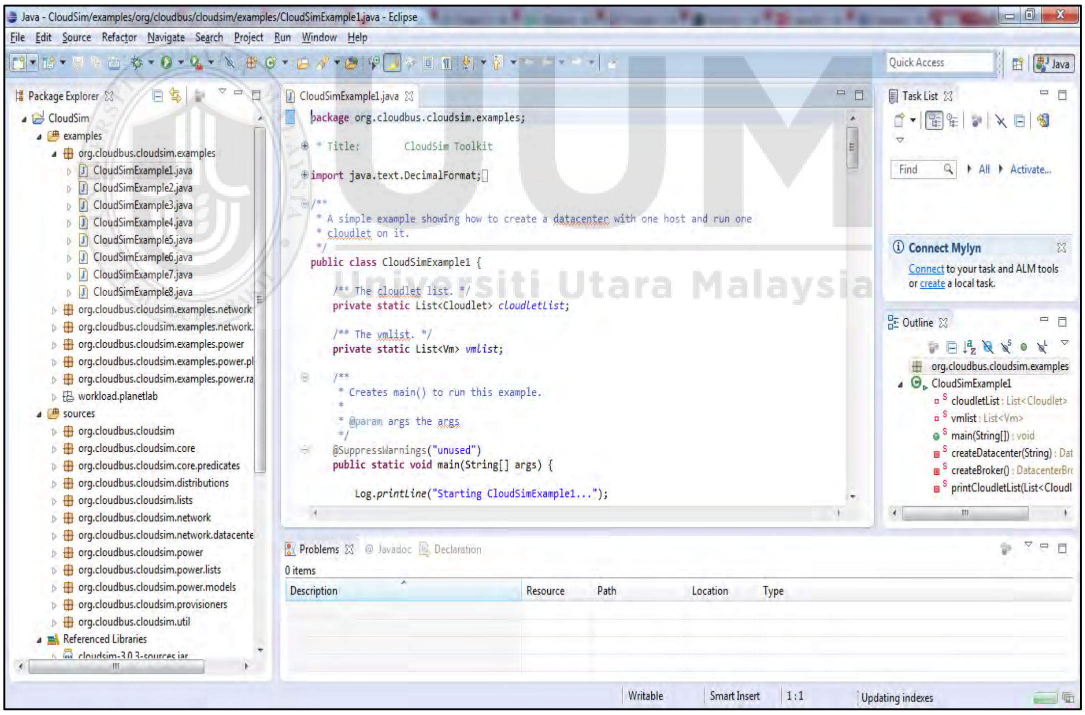
From the above table, it can be clearly seen that previous research works have many limitations regarding several reasons. The first reason, such proposed solutions lack a mechanism that ensures the communication between VMs located on different servers. To clarify, researchers in [83] have done a great idea whereby provided two metrics for measuring the attacks, but they do not consider the issues on data interaction between the VMs that attackers can violate while they communicate. The second reason, some solutions assume that hypervisor is always protected unless the attacker can get control of the hypervisor, which directly controls the whole VMs. Finally, other solutions provided security for phone clones on the cloud side, but not on mobile devices.

2.7 Related Cloud Tools and Comparative Study

There are various simulators for both cloud computing and MCC that attracted researchers to utilize their potential capabilities in different research issues such as load balancing, power consumption, offloading, security and privacy issues. Table 2.2 shows a comparison between the most popular cloud simulators selecting the right simulator tool is very necessary to evaluate the research project. These are the most popular simulators include CloudSim, GreenCloud, ICanCloud, GroudSIM, NetworkCloudSim, secCloudSim and so on. Thus, a critical study is carried out in this section in order to know the advantages and weaknesses of these simulators.

1. CloudSim

CloudSim is a simulator toolkit that provides modeling and simulation of the cloud environment and its resources [124]. Figure 2.8 shows the CloudSim package integrated with Eclipse Java IDE. CloudSim is able of simulating the variety cloud component such as data centers, hosts, users, virtual machines, service brokers and cloudlet, also different policies proposed by researchers like virtual machine allocation policies, outsourcing policies and scheduling. All the resources implemented as a Java class examples that easily requested an object to help researchers to deal with different requirements. The users can make different type of research work including the development of new policies, editing the existing one, implementing mechanism and approach and make several tests under different scenarios before trying on the real systems. The CloudSim simulator is widely used by researchers and organizations such as HP labs in USA. It demonstrates its success by conducting numerous case studies with it. However, the major disadvantage of this tool is the absence of the graphical panel.


Java - CloudSim/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explorer
CloudSim
examples
org.cloudbus.cloudsim.examples
CloudSimExample1.java
CloudSimExample2.java
CloudSimExample3.java
CloudSimExample4.java
CloudSimExample5.java
CloudSimExample6.java
CloudSimExample7.java
CloudSimExample8.java
org.cloudbus.cloudsim.examples.network
org.cloudbus.cloudsim.examples.network.
org.cloudbus.cloudsim.examples.power
org.cloudbus.cloudsim.examples.power.pl
org.cloudbus.cloudsim.examples.power.ra
workload.planetlab
sources
org.cloudbus.cloudsim
org.cloudbus.cloudsim.core
org.cloudbus.cloudsim.core.predicates
org.cloudbus.cloudsim.distributions
org.cloudbus.cloudsim.lists
org.cloudbus.cloudsim.network
org.cloudbus.cloudsim.network.datacente
org.cloudbus.cloudsim.power
org.cloudbus.cloudsim.power.lists
org.cloudbus.cloudsim.power.models
org.cloudbus.cloudsim.provisioners
org.cloudbus.cloudsim.util
Referenced Libraries
cloudsim-3.0.3-sources.jar
CloudSimExample1.java
Task List
Find
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
org.cloudbus.cloudsim.examples
CloudSimExample1
cloudletList : List<Cloudlet>
vmlist : List<Vm>
main(String[]) : void
createDatacenter(String) : Dat
createBroker() : DatacenterBr
printCloudletList(List<Cloudl
Problems
Javadoc
Declaration
Writable
Smart Insert
1 : 1
Updating indexes

```
package org.cloudbus.cloudsim.examples;

 * Title:        CloudSim Toolkit

import java.text.DecimalFormat;

/**
 * A simple example showing how to create a datacenter with one host and run one
 * cloudlet on it.
 */
public class CloudSimExample1 {

    /** The cloudlet list. */
    private static List<Cloudlet> cloudletList;

    /** The vmlist. */
    private static List<Vm> vmlist;

    /**
     * Creates main() to run this example.
     *
     * @param args the args
     */
    @SuppressWarnings("unused")
    public static void main(String[] args) {

        Log.printLine("Starting CloudSimExample1...");
```

0 items

Description	Resource	Path	Location	Type

Figure 2.8. CloudSim Package Integrated Eclipse

CloudSim Architecture: CloudSim has been designed to support the modeling and simulation in terms of providing management for cloud virtualization (VMs), storage, bandwidth, and memory. Figure 2.9 shows the CloudSim architecture with a different layer that contains multiple modules. The CloudSim previously has used a discrete event simulation engine, namely SimJava [125] which is illustrated on the lowest level of CloudSim architecture. This layer includes many functionalities that needed by the higher simulation level to create and process operations and events, manage and update the cloud entities (data centers, host, virtual machine, services, broker), communication between component [124]. Currently, the SimJava is removed due to the incapability of supporting such advanced operations. Next, GridSim layer presents functionalities of GridSim for modeling and simulating various Grid Infrastructure and its resources, also creating networks and the traffic profiles. All these functionalities have extended by the CloudSim to support extra modeling systems of cloud computing [126]. The SimJava and GridSim layers are encapsulated in the CloudSim core simulation engine.

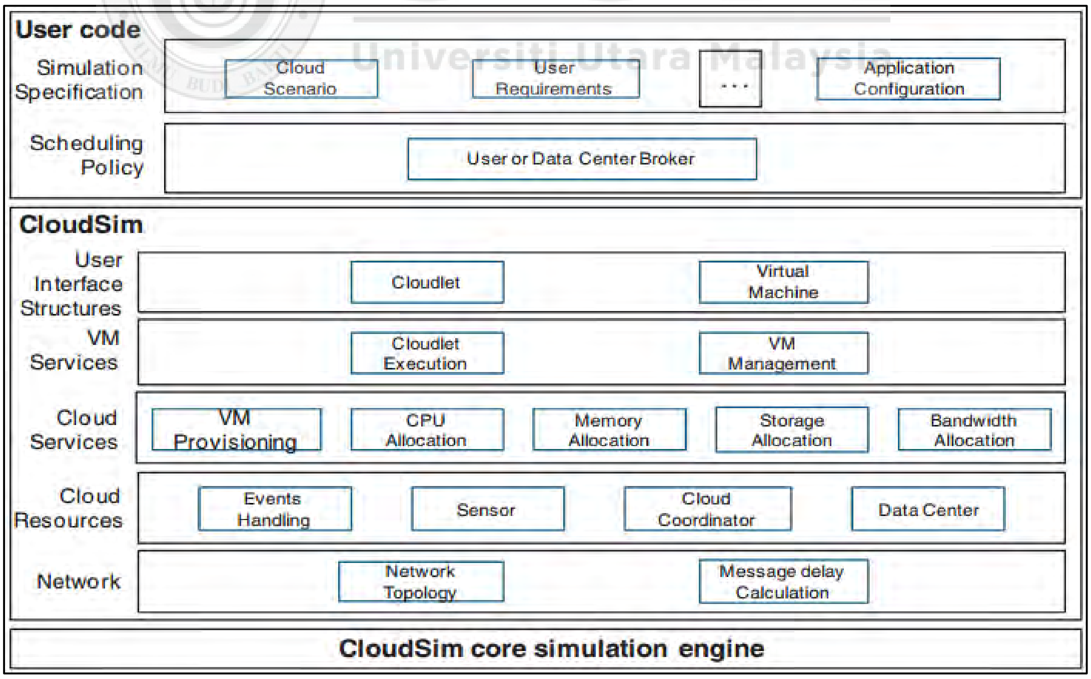


Figure 2.9. CloudSim architecture

The upper CloudSim layer consists of many units which contain several components such as virtual machines management services, cloud services, and resources. In this layer, thousands of cloud computing instantiation entities (VMs, host, application, and datacenters) can be managed and executed concurrently and transparently during the simulation step. Moreover, this layer enables to tackle other main issues related provisioning host to VMs depending on user requirements, dynamic monitoring and control the application execution. In our research, we need to programmatically extend the virtual machine allocation policies and other resources and policies to implement our research work approach. The top layer in this CloudSim architecture is the User Code. In this layer, the configuration for the host is established by determining the number of machines and their characteristics, also the configuration of applications by specifying the number of tasks and their requirements, number of user and application types, likewise other configurations related to scheduling policies.

2. Green Cloud

Datacenters are considered as the main components that provisioning the computing resources for tenants in cloud computing environment. Many research works reveal that communication between components on the datacenters and the computing units (Servers, VMs, and Switches) consume a high-energy cost. In order to come up with a new optimized energy aware schema from researchers. Green Cloud simulator [127] is destined for such issue of energy awareness of datacenters. The disadvantage of the Green Cloud is its TCP/IP implementing in the datacenter network, which requires high memory requirement and large simulation time. Green cloud scales well when the simulation overhead is low. Figure 2.10 shows the Green Cloud architecture.

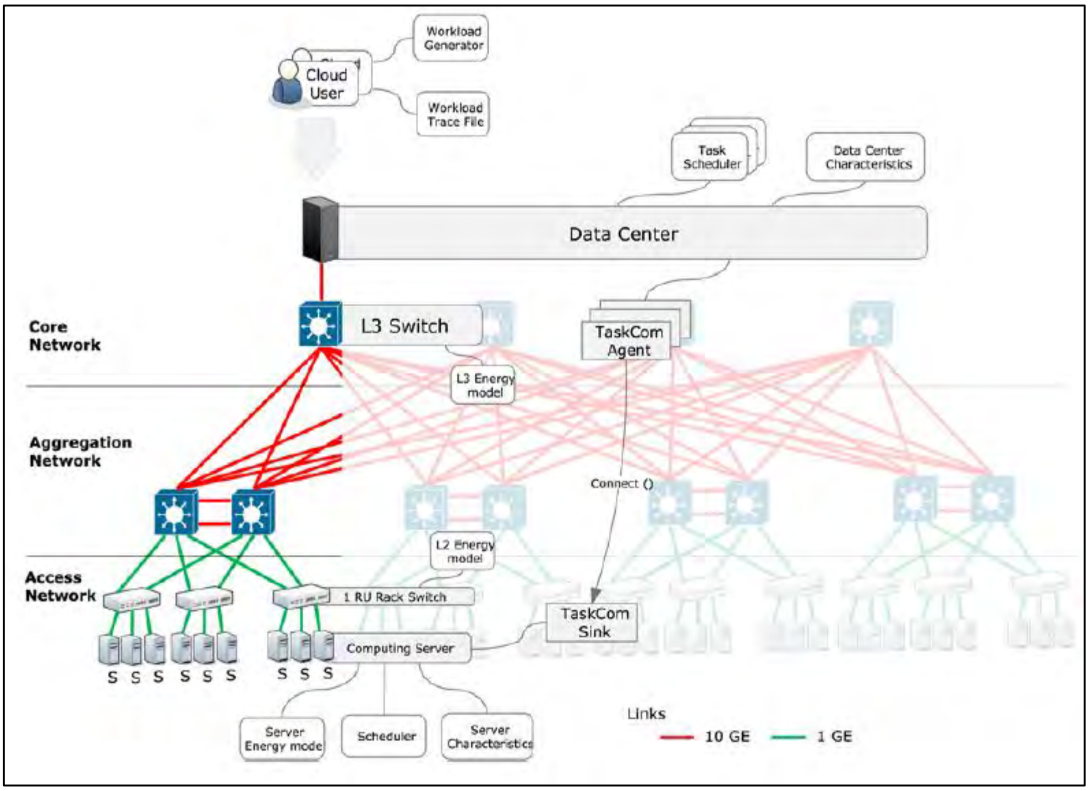


Figure 2.10. Green Cloud architecture

3. iCanCloud

Using simulation tools are considered as the best analysis approach to prevent the spending of cost (time, money) when studying and verifying different complex scenarios. This permit to develop and to evaluate the performance of any proposed approach in repeatable and control manner. iCanCloud is a simulation platform allows the simulation of various experiments of scientific researchers especially on the cloud brokering policies of virtual machines [128]. The simulator is concentrated on the Amazon cloud provider to do experiments in such platform. iCanCloud can simulate large experiments with either 32 bit or 64 bits systems because is written with C++ language compared to CloudSim that is created with Java language, which affects negatively the 32 bits systems design. The iCanCloud can use all the memory available on the hosts while running the experiments (i.e.: for 64 and 32 bits physical machines).

More advantage of iCanCloud is the graphic interface that helps researchers to create experiments and scenarios easily. The ability of iCanCloud to run parallel simulations, so one experiment can use different machines. CloudSim, MDCSim and GreenCloud do not support this point. Figure 2.11 illustares iCanCloud architecture.

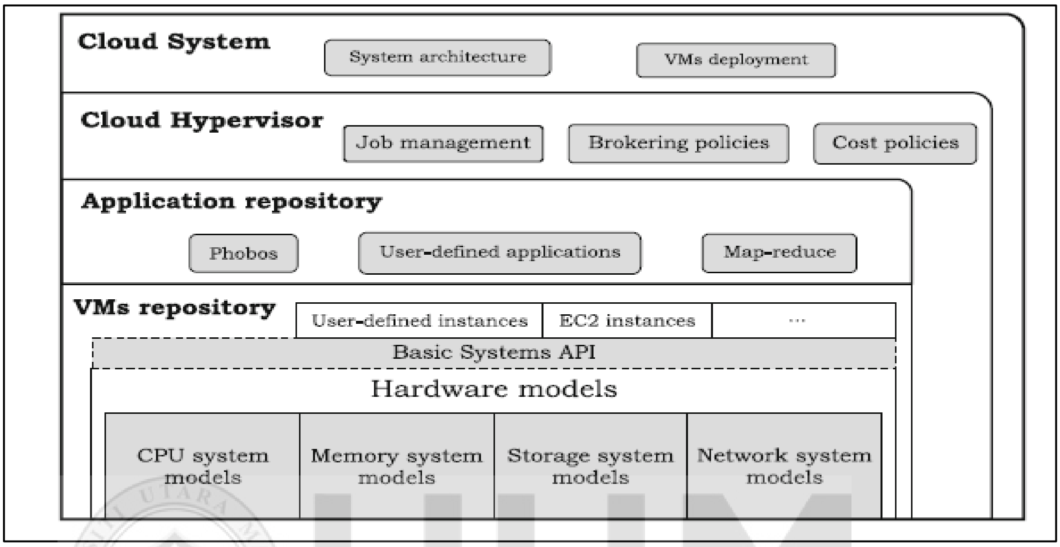


Figure 2.11. iCanCloud architecture

4. GroudSim

Due to the process based approach that executes each separate thread in host machine and the lack of the high scalability of the existing simulators such as CloudSim and GridSim. A GroudSim is proposed to support the large scientific applications either on the Grid or cloud systems. GroudSim is an event-based simulator, which only requires one simulation thread [129]. GroudSim is able to support complex simulation scenarios such as calculation of costs. The focus of GroudSim is the infrastructure as a service (IaaS). GroudSim consists of SimEngine for specifying the time-advance algorithm and the event lists. GroudSim is developed in Java environment. One of the main advantages of GroudSim is the ability to change the configuration when errors occur.

5. NetworkCloudSim

Previous simulators such as CloudSim consider the datacenters as a collection of virtual machines, which allow only the modeling of simple applications models. That affect the results by obtaining both inaccurate and non-realistic solutions. NetworkCloudSim is a simulator tool that permits to model realistic application such as message passing applications (MPI) that requires tasks communication and sharing data between each other [130]. This tool allows the modeling of the various network topologies on the Cloud Computing environment. NetwrorkCloudSim is the tool that we choose to simulate and evaluate our proposed approach. More details about NetworkCloudSim simulator are in chapter three.

6. secCloudSim

There is a strong need to develop security cloud simulators that provide opportunities to simulate the security experiments of different researchers' policies and approaches. Therefore, it is worth mentioning secCloudSim [131]. The latter, is an extended secure layer designed and implemented on the top of iCanCloud Simulator. secCloudSim is considered the only new secure cloud simulator among other simulators which are all not supporting the security, confidentiality and privacy aspect. The new simulator provides the users with the basis security characteristics of authorization and authentication in the cloud environment. However, secCloudSim has only focused on basic features such as authentication and authorization modules. Further, the secCloudSim is not able to model and simulate complex distributed applications because the backbone of it based on the iCanCloud simulator. Figure 2.12 shows the general architecture of secCloudSim.

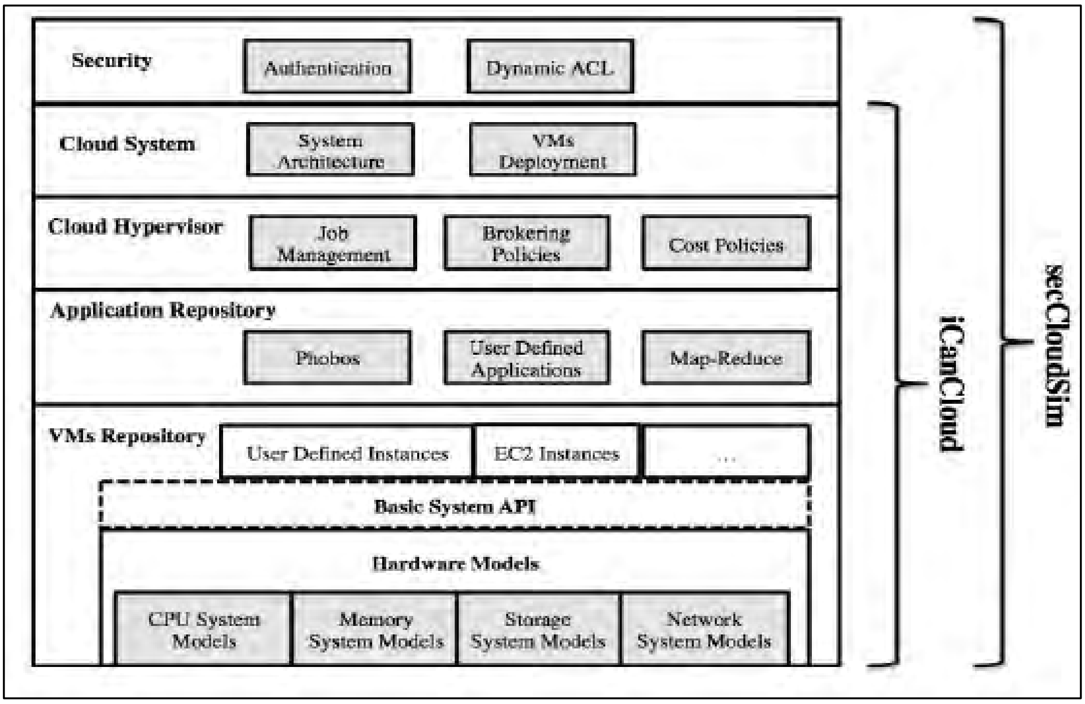


Figure 2.12. secCloudSim architecture

Table 2.2
Cloud simulators tools' comparison

Simulator	Availability and Programing language Used	Strengths	Limitations
CloudSim [124]	Open Source, Java	Ability to model and simulate the cloud environment and its resources provisioning policies (VM allocation policies, load balancing, Resource power consumption)	-Not support the modeling of parallel applications (communication Tasks) -Inaccurate results of complex applications evaluation - Not support security aspect
Green Cloud [127]	Open Source, C++ and OTcl	High capability of modeling and simulation of Energy awareness in cloud environment	High simulation scenarios overhead affect the results due to the TCP/IP model and simulation execution that measured with minutes (others simulators with seconds) -High memory requirement and large simulation time - Not support security aspect
iCanCloud [128]	Open Source, C++	-Simulation of cloud brokering policies -Support heterogeneous system -High memory leverage	-Using two different language C++ and OTcl to implement one single experiment -Not support security aspect

Table 2.2 continued

GroudSim [129]	Open Source, Java	-Support large experiments simulation (complex applications: time advance algorithm and the event lists) either in Greed or Cloud environment	The focus of GroudSim is the infrastructure as a service (IaaS) and no other services such as SaaS and PaaS - Not support security aspect
NetworkCloudSim [130]	Open Source, Java	Ability to model and simulate the realistic distributed applications with communicating tasks such as message passing applications (MPI), VMs networking -Modeling of the various network topologies on the Cloud Computing environment	- Not support security aspect
secCloudSim [131]	Open Source, C++	-Give the opportinuty for cloud users to work on designing secure cloud simulator as a new resarch direction. -Allow cloud users to simulate their secure polices with the basis security characteristics of authorization and authentication in the cloud.	- Not support the to model and simulate complex distributed applications because the backbone of it based on the iCanCloud simulator - secCloudSim does not implement other security features such as Encryption and decryption of users' data which preserve : the integrity and privacy of the virtual machines. As welll, secure VM allocation polices and other attacks like : co-resident and hypervsior attacks.

From the table above and except the case of NetworkCloudSim simulator. All the simulators are not able to model and simulate complex distributed applications with communicating tasks aspect on the cloud environment. Indeed, the results of the evaluation of any mechanism and policy will not be accurate as the researchers need. Furthermore, the Majority of cloud simulators do not focus on security issues especially the security of users' sensitive data in either mobile or cloud environment unless secCloudSim. For that reason, we motivated to create a secure cloud simulator in order to help researchers to evaluate their security approach and policies. Chapter 5 illustrates our added security layers on NetworkCloudSim which make it able to simulate security experiments or research secure scenarios that faced cloud community.

2.8 Diffie–Hellman and Hash on Encrypted Key Exchange

This section details Diffie-Hellman's algorithm and its main involved steps as well as the Hash algorithms. The important reasons for using them in this research work compared to existing techniques are also presented.

2.8.1 Diffie-Hellman Algorithm

Diffie-Hellman algorithm is a digital encryption method to securely exchanging cryptographic keys between two entities that willing to communicate sensitive data between them [61, 132, 133]. This method allows a secure channel between two entities that have no prior recognition of each other over an unsecured and public channel. Diffie-Hellman method involves the use of mathematical functions from both parties in order to get a common value which considers as a secret key that known only from those parties [134, 135].

Let's assume that there are two parties A and B intend to communicate over unsecured communication channel. We assume that A is VM1 and B is VM2 or A is User and B is Secure Cloud Proxy. There is another party C (Attacker) eavesdropping A and B in order to get the data shared between them.

When A encrypts the message that intends to send to B. It must also send the key to decrypt this message (Secret Decryption Key). The attacker can get the encrypted message as well as the key to decrypt this message while A sends to B. Diffie-Hellman provides the solution for this situation as follow:

First, A sends a public key to B, and B also sends a public Key to A. The Attacker C cannot see the public and private keys of both A and B. Second, A and B agree on two numbers: one prime number and generator number. Using the signature generator, the private key of A and B can be constructed. The attacker or any third party cannot find these two numbers.

In order to ensure security and privacy, first of all, the users A and B willing to exchange information, need to generate a pair of public and private keys. The public keys have to be exchanged before. Both public keys of A and B are generated using the following equation:

$$\text{Public key} = \text{Generator}^{\text{private key}} \bmod \text{Prime} \qquad (2.1)$$

After A and B exchanges their public keys. They can calculate the secret key by using the following equations:

$$\text{Secret key A} = \text{Public key B}^{\text{private key A}} \bmod \text{Prime} \qquad (2.2)$$

$$\text{Secret key B} = \text{Public key A}^{\text{private key B}} \bmod \text{Prime} \qquad (2.3)$$

They find the same result once the secret keys have been constructed, the same results can be founded (secret key A = secret key B). However, C cannot find the same result because it difficult to get the private number of A and B. More particularly if it is a big integer. Moreover, C will find a mathematical problem called Discrete Logarithm problem [61]. For example, it is easy to calculate: 3^{15} mod 17 = 6 but it is very difficult to calculate $3^{?}$ Mod 17 = 6. Especially when the private number is longer than 100. Therefore, the calculation will be computationally unsolvable. This secret key can use it with any encryption method.

A comparison between the Hash Diffie-Hellman technique and other encryption techniques will be more elaborated in chapter 6.

2.8.2 Hash algorithms

The hash function is one of the most powerful cryptographic tools that preserves data integrity transmitted over the unsecured public network [136, 137, 138]. There are many hashing algorithms such as MD5, SHA1, SHA224, SHA256, SHA384, SHA384, and SHA512. Otherwise, in encryption algorithms, there are AES, DES, RSA, Rot13, RC4, and XOR. Both hash and encryption algorithms encrypt the message to a ciphertext (encrypted message). However, in hashing, whatever the size of the message that intends to hash, the result given is a fixed size. For example in MD5 hash function, the resulting hash is 32 hexadecimal characters (fixed size) whatever the size of the message [139, 140].

In hashing, there is no need for a key to make the hashing process. The main objective of the hash function is guaranteeing the invisibility of the hash ciphertext, which can be seen only by the party which hash it (data integrity protection). If any party wants to know the original value of hash cipher text, it needs to give the right plaintext.

Using the example of cloud proxy, the user sends the right plaintext to the trusted cloud proxy, the proxy hashes the plaintext and checks the existing hash codes, if the value of the ciphertext matches any other ciphertext, the proxy then provides authorized access for the user in order to leverage its cloud services. If there is any change on hash ciphertext, that means that user data integrity has been modified or retrieved while it transmitted over the network from any attackers.

In this research, we adopted SHA256 function hash, which gives a ciphertext of 64 characters. In the encryption method, the message that being sent, is encrypted with a random key. In such method, the change of the plaintext gives unfixed size of ciphertext. Moreover, if the character changes, for example: “hello” and “Hello”, the ciphertext will be the same value. However, in hashing, if character changes, so the value of ciphertext will be changed as well.

2.9 Summary

In this chapter, we presented the concepts related to mobile cloud computing and security issues. First, it defined main concepts for MCC including MCC services, offloading technique, security and privacy issues, virtualization and the most popular attacks on virtualized systems. Then, current security cloud virtualization based mechanisms, virtualization hypervisor model; encrypted and VM allocation policy techniques were reviewed and strategically analyzed for the protection of VM information and significance. Furthermore, it introduced some of the recent related mechanisms to workload balancing and energy consumption in MCC. Based on our literature review analysis, this research studies the issues that related to the security and privacy of the virtual machines sensitive data and information in terms of VMs co-location attacks in the same host or on different hosts to achieve high security level and on both cloud and mobile parts for deploying and implementing their security techniques. In the next chapter, the research methodology for achieving the objectives of this research and evaluating the proposed approach will be presented.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

Currently, securing the sensitive data of distributed mobile applications that transit over cloud communication gateways is crucial. However, it has become easier for a third party to intercept and alter these data. The main objective of this research work is to come up with an approach that enhances the security of the user's sensitive data on the Mobile Cloud Computing environment (MCC) deployed on thin virtual machines from being retrieved and lost for another malicious user. The proposed approach contributes to protecting sensitive data shared among distributed mobile applications against different common attacks on virtualization layer such as co-resident attacks, hypervisor attacks and distributed attacks. Three well-known security policies are included in this approach that ensures the sensitive data safety via protected distributed mobile applications to be intercepted and altered while communicate which each other, which lead to achieving a high-security level on the mobile cloud computing environment.

This chapter presents the main steps that will be used to achieve the research objectives. An efficient and robust methodology is required to accomplish the research objectives presented in chapter one.

The remainder of this chapter is organized as follows: Section 3.2 details the research methodology which contains research phases, conceptual model and the development approach process. Section 3.3 details the proposed approach's policies. Section 3.4 presents the different techniques of performance evaluation and its evaluation environment. Furthermore, we define the security metrics to evaluate the performance of the secure cloud proxy and its related policies. Finally, Section 3.5 concludes the chapter.

3.2 Research Framework

In order to enhance the security level for sharing sensitive data in MCC and managing distributed VMs in a high abstraction level, we propose research approach consists of combination of three well-known security-based policies (i.e., co-resident attacks, hypervisor attacks, and distributed attacks) with a new security proxy-based approach for modeling the privacy level of sensitive data exchanged between different distributed mobile applications while the cloud is used for sharing computing resources and its ability to manage VMs efficiently. This approach aims to enhance a distributed mobile application to be more robust and to cover a maximum protection space against thin virtual machines' attacks that occurred inside and between different cloud hosts. Our approach is dedicated to securing VMs over fraudulent exchanges on sensitive data. Fraudulent exchanges can occur due to unauthorized VMs that aspire to share and to access the data through the cloud host or the communication gateway. These attacks are exactly performed while leveraging the virtualization techniques. The approach will also aim for reducing the cost in terms of security management time and computation complexity (i.e. proxy-based as a principal gate), as another benefit that makes the solution more practical and desired for the current cloud platforms. For achieving these objectives, a well and accurate understanding of the existing solutions is required to get a better enhancement and best outcome [141].

Our commitment to achieving the objectives set in Chapter One. The research follows the basic approach's stages as depicted in Figure 3.1. At the beginning of the research work, a taxonomy of security in MCC environment and mechanisms are examined which represents some solutions for solving such related problem in order to obtain a research plan.

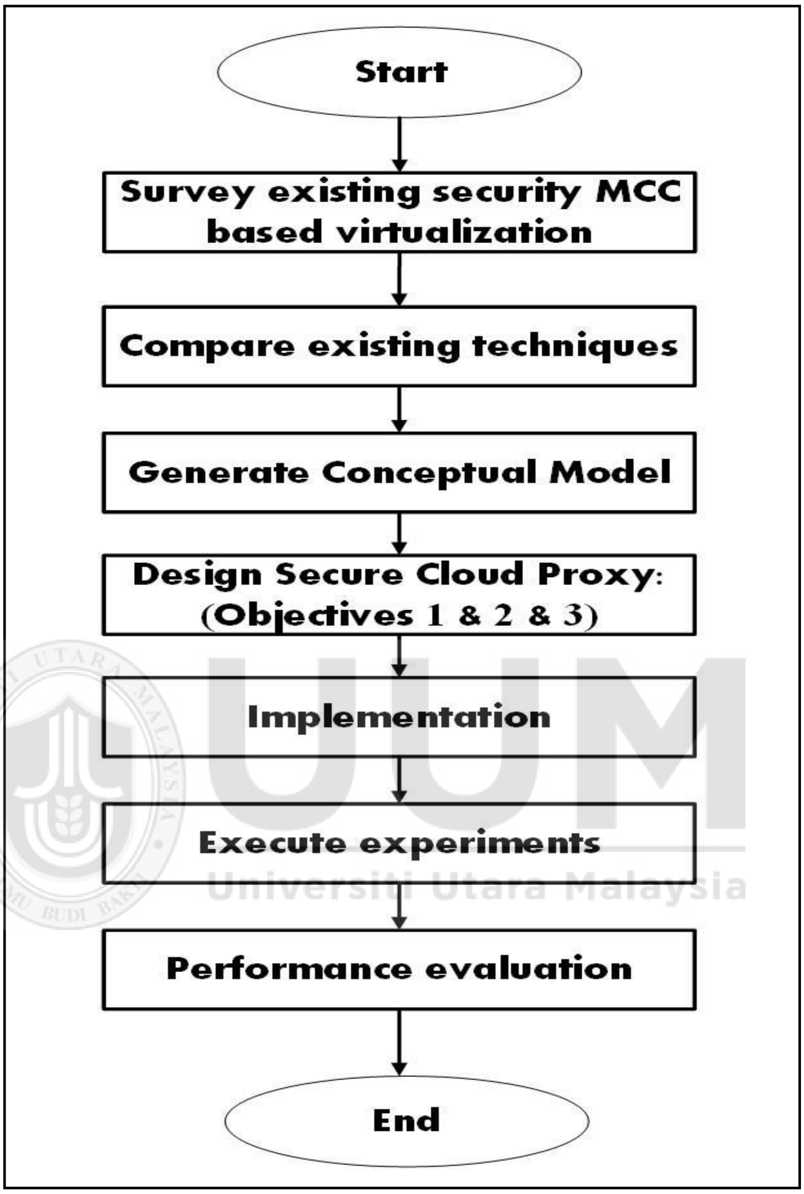

Start
Survey existing security MCC based virtualization
Compare existing techniques
Generate Conceptual Model
Design Secure Cloud Proxy: (Objectives 1 & 2 & 3)
Implementation
Execute experiments
Performance evaluation
End

Figure 3.1 Research Design

Secondly, we studied the literature review of the proposed solutions and their algorithms to solve the security problem on the virtualization layer in both CC and mobile cloud in order to identify the strength of each one. Then, a conceptual model is provided and generated after examining step one and two. For the simulation, NetworkCloudSim simulator will be used in order to achieve the main objective of this research work. Finally, much experimentation will be conducted as a final step to evaluate and validate the performance of the proposed approach.

3.2.1 Research Phases

This research work consists of multiple research steps in order to realize the main objectives mentioned in Chapter I:

1. A comprehensive literature review about the security in MCC in general and the existing solutions for virtualization security attacks.
2. Design a mobile user control access for preventing the unauthorized access of malevolent users to the cloud provider services.
3. Design secure VM allocation policies consist of securing the different type of users' VMs allocation and ensure the protection of the Hypervisor security by defending the VMs from any attack that intends to retrieve sensitive insider data.
4. Design a software VMs communication policy based on the proxy in order to protect the privacy and confidentiality of sensitive data exchanged among VMs.
5. Extend NetworkCloudSim simulator in order to enable simulating security scenarios of distributed intensive applications that use virtualization-based techniques for processing on cloud environment and getting real and accurate simulation results.
6. Evaluating the proposed approach in a mobile cloud environment in terms of the effectiveness of defending against the different stated attacks, also evaluating the approach in terms of cost: security

management time and computation complexity through simulations experiments.

3.2.2 Conceptual Model

In order to make the MCC more attractive by mobile users, it is compulsory to provide the best services that users can be confident once using them. One of the main aspects that hesitate the costumers to engage in this technology is security issues especially when they move out their sensitive information to the cloud infrastructure using the virtualization technique. Therefore, the increasing need for an effective and efficient approach that can cover the maximum threats and risks which can occur for the legitimate VMs becomes a major concern. This is very helpful to answer the required protection of the user sensitive-data from being stolen whether is hosted on the same and different hosts on the cloud.

The efficient and robust approach for protecting and maintaining the user sensitive-data over malicious attacks has led to a trade-off between costs (e.g., fast security management time, low computation complexity) and improved security level. In order to meet these requirements, this research aims to propose an effective approach based on two security policies (objective one and two) and one technique (objective three) that guarantee information privacy and confidentiality against different attacks that performed on VMs. We mainly focus on the suited mechanisms for distributed mobile applications that use virtualization techniques and share sensitive data to leverage cloud services and resources.

The protection of sensitive data in the distributed mobile application becomes a major issue, and it involves a wide range of several activities from various actors with distinct sites exchange a vast amount of data, which makes the security of such data very costly. Therefore, sensitive data may be the target of malicious attacks due to the lack of privacy and confidentiality checking. Unauthorized access to such sensitive data can cause many problems such as privacy violation. For achieving the main objective, two policies and one technique are included:

1- Design a client control access and secure VM allocation policies on the different hosts in the cloud.

2- Design a hybrid-policy consist of VM allocation policy and Hypervisor policy in order to ensure the data VMs from being retrieved in order to guarantee the thin VM integrity and hypervisor integrity.

3- To develop a proxy-based security technique on the cloud in order to protect the transferring of sensitive data between VMs.

A conceptual model for the research has been elaborated and described in Figure 3.2.

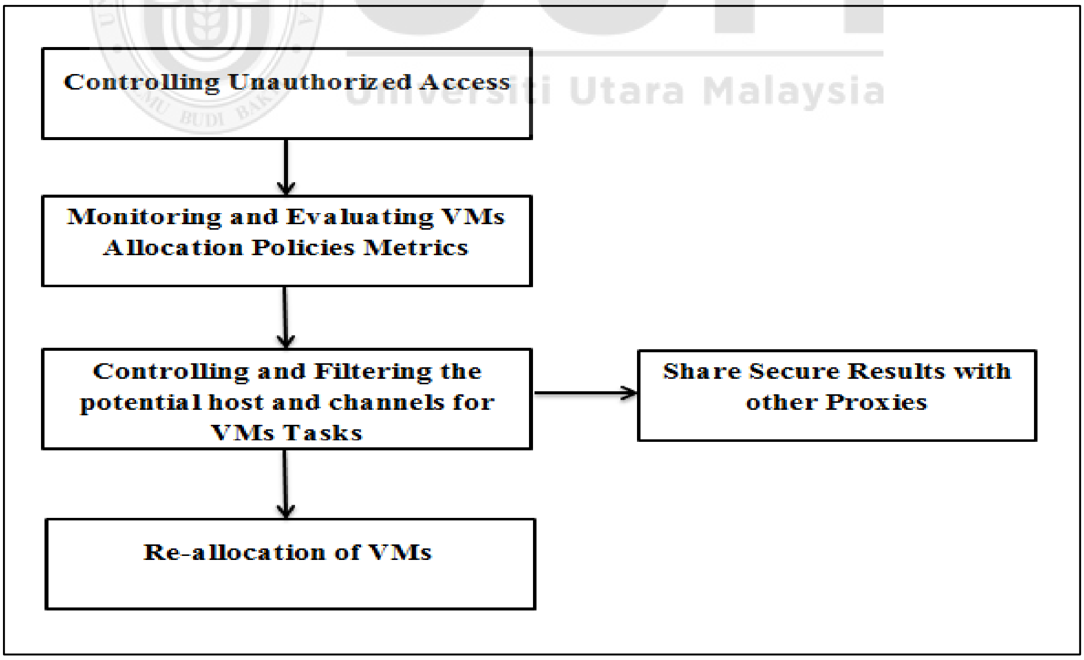

Controlling Unauthorized Access
Monitoring and Evaluating VMs Allocation Policies Metrics
Controlling and Filtering the potential host and channels for VMs Tasks
Share Secure Results with other Proxies
Re-allocation of VMs

Figure 3.2. Conceptual Model

The conceptual model of the research work contains five main steps. The first step is responsible for defining the identity of remote client site and check that the nonvulnerability and the eligibility of access are verified. Then, analyzing the mobile client requests and check if the security for VM allocation of distributed sensitive applications (e.g. critical banking intensive application and mobile health application) is authorized by the cloud provider. The second step is responsible for monitoring and evaluating the security policies of the proposed approach in terms of pre-known metrics (i.e. coverage and efficiency of attacks) related to the number of target VMs and input/outputs sensitive communication getaways. The third step is responsible for controlling and filtering the potential attacked host for VMs and restricted channels that transfer data from the mobile to the Cloud (also between VMs) which means reduce the number of externally available hosts and also reduce the time and the cost benefits from exploiting a computing resource. This step aims to enhance the security level for sharing security analysis results using the Share Secure Result component to the proxy. It is responsible for providing security directives to the policies proposed on the approach. Finally, the last step is called Re-allocation of VMs that update the VMs allocation based on the proxy directives. Figure 3.3 shows the proposed approach.

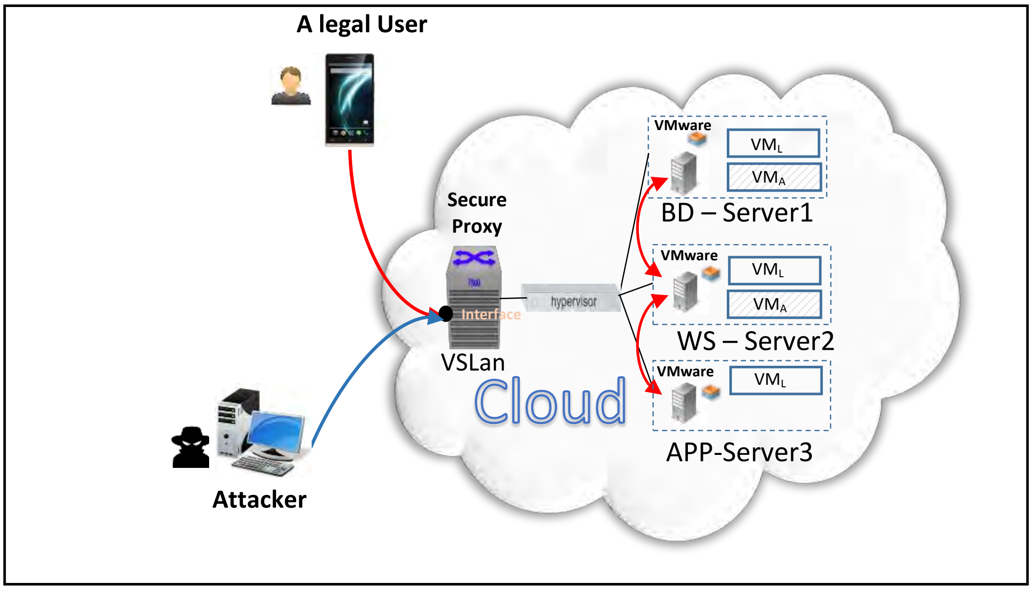


Figure 3.3. The Proposed Approach

The work of [83] is extended and adapted for distributed mobile applications in order to decrease the coverage and efficiency attack while reducing the attacked rate and the security management time. We integrate VMs secure interaction among various VMs deployed on different hosts along three security policies: input and output interfaces, hypervisor, and untrusted sensitive data using security-based proxy. Furthermore, benchmark data is used in order to validate our work.

3.2.3 Development of the Proposed Approach

This section aims to answer the research questions of the research work and ending with a mechanism or developed approach. Many processes in the design must be repeated to get a complete approach or mechanism. Figure 3.4 shows the development process of our proposed research work.

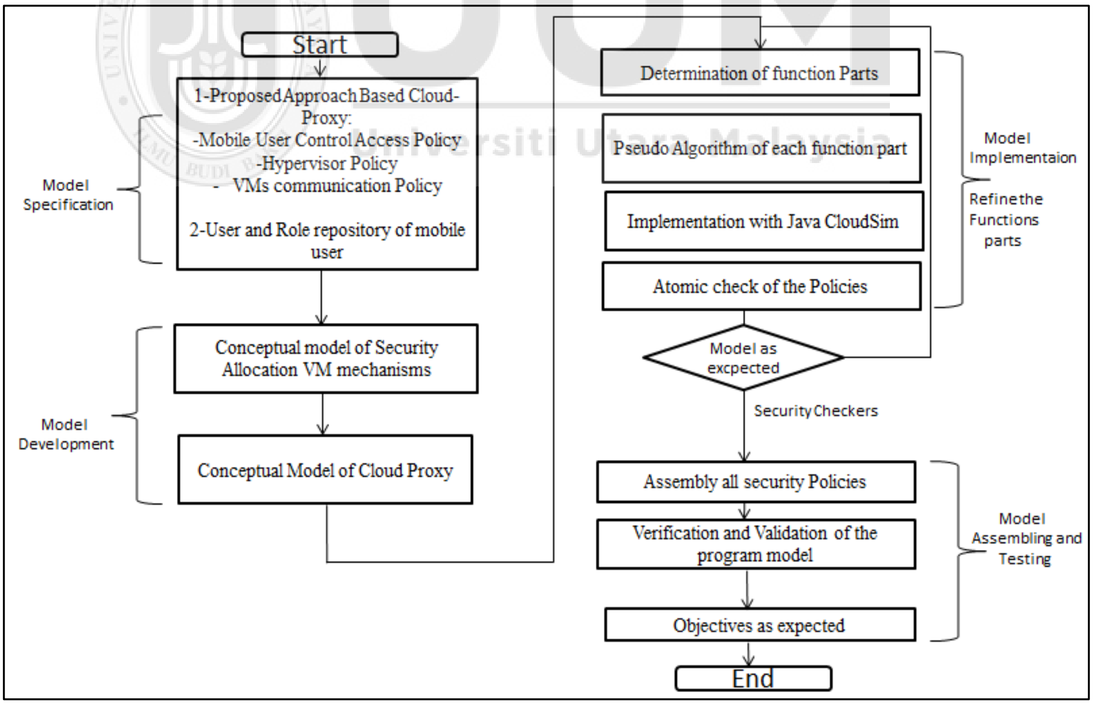

Start
1-Proposed Approach Based Cloud-Proxy:
-Mobile User Control Access Policy
-Hypervisor Policy
- VMs communication Policy
2-User and Role repository of mobile user
Model Specification
Conceptual model of Security Allocation VM mechanisms
Conceptual Model of Cloud Proxy
Model Development
Determination of function Parts
Pseudo Algorithm of each function part
Implementation with Java CloudSim
Atomic check of the Policies
Model Implementaion
Refine the Functions parts
Model as excpected
Security Checkers
Assembly all security Policies
Verification and Validation of the program model
Objectives as expected
Model Assembling and Testing
End

Figure 3.4. Development Approach Process

The development approach process is containing four main models, namely Model Specification, Model Development, Model Implementation, and Model Assembling and Testing. The first model is called Model Specification, aiming to determine the different policies integrate inside that the proposed security approach. The model development is considered as a complex step to achieve a realistic and final design because of many sub-processes that can be added and removed from it in the way of the research. The third model is named Model Implementation responsible for managing and refining the functions included in the research approach. Determining, implementing and checking the different parts of the policies are required for the research. The last model is called assembling and testing model. This model aims to assemble all parts on one complete unit, also testing the policies and validate them in order to achieve the main objective of this research work.

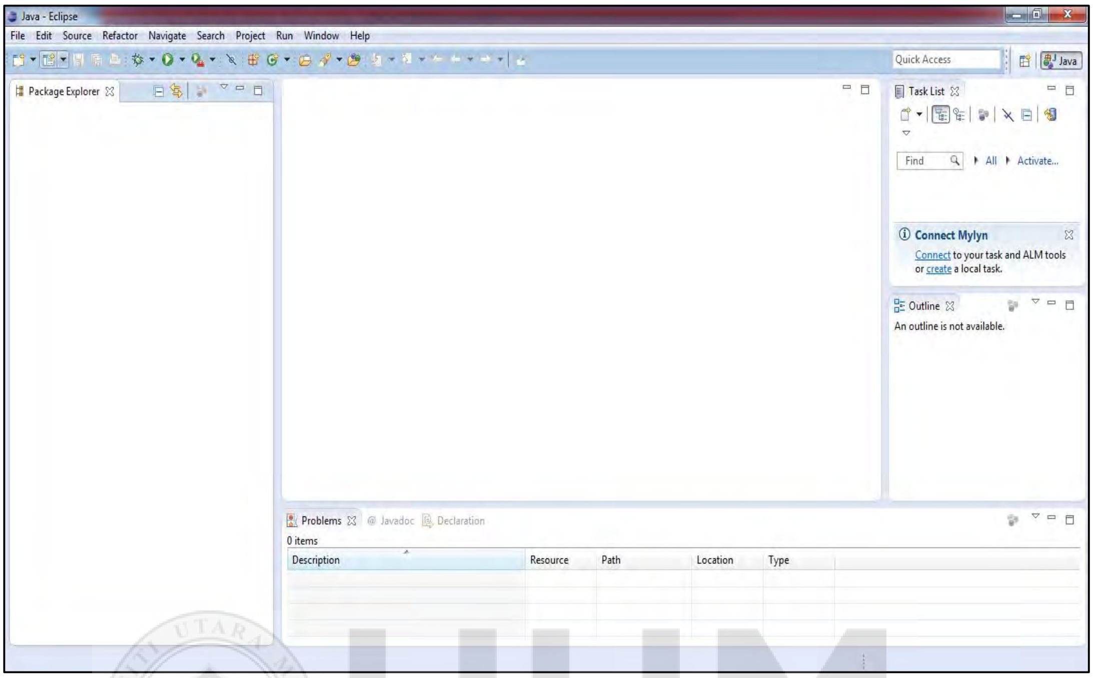
The main deliverables of the Perspective Study are:

- Chapter four and five (Objective one, two, three and four)
- Various research papers
- Design, implementation of the proposed approach
- Validation of the proposed approach

3.2.3.1 Implementation and Validation of the Development Process

The proposed approach consists of several parts. Each part provided a function to a specific target. These functions are implemented using Java programing language on the whole research work. Moreover, for ensuring that every single mechanism works correctly, code should be clean from any bugs and errors [142]. To realize that, Eclipse Java Development Tool (JDT) is the programming platform that will be used to achieve the objectives. Eclipse IDE supports the edition, compilation and generating executable files, also making the development process easier. Figure 3.5 shows the

Eclipse IDE with bugs and errors' panel for verifying that the mechanisms have written without any errors and warning.



Connect to your task and ALM tools or create a local task.

An outline is not available.

0 items

Figure 3.5. Java Eclipse IDE

For the validation model of this research work, the complete development process is presented below in three main consecutive steps (see Figure 3.7). We integrate the validation model in which researchers met their intended requirement in terms of policies included and the accuracy of test results [143]. All the parts and constraints are described in Figure 3.5 to satisfy elaborated requirements for main objective and sub-objectives.

3.3 Policies-Based Security Approach

The research approach proposed in this thesis contains three secure policies. Each policy is described below with related algorithms:

3.3.1 Mobile User Control Access Policy

Firstly, before mobile user access to service cloud provider for deploying their requested tasks, they authenticate by entering identifier and password. The secure proxy checks if the mobile user has access or no (control access requirement). If a user exists in a Black List then return access denied. Else, the proxy has a User Authentication Table includes all the users that have access to the cloud service.

In the case of the user provides correct identifier and password (e.g. exists on the User Authentication Table), so the proxy calls hypervisor to allocate VM in any available host (server side). Elsewhere, the proxy increases the probability of unauthorized access and the user re-enter the identifier and the password. When the probability of unauthorized access becomes greater than a given threshold, the proxy detects the malicious mobile user, updates the blacklist table and return access denied. After that, the mobile device that gets access will classify on User Trust Level Table as fully trust mobile device.

The secure proxy classifies in the green level, the mobile device that provides true identifier and password. Figure 3.6 shows the secure mobile user control access policy included in the proposed proxy.

```
Enter the user identifier and the user password
If secure proxy finds a user exists in a blacklist  then
   Deny the status of the user
else
while (the user identifier and the user password  are not existing)
   Increase the probability of unauthorized access
   If the probability of unauthorized access greater than probability threshold then
     Insert the user in a blacklist  and deny the status of the user
   else
     Enter the user identifier and the user password and check again
End while
Authorized the status of the user
The user’s virtual machine is allocated in the cloud host
Give the user full authorization access (i.e. green trust level)
```

Figure 3.6. Secure Mobile User Control Access Policy

3.3.2 Hypervisor Policy

The secure proxy classifies in the green level, the mobile device that provides the true secure keys. Otherwise, degrade mobile device to a lower level (orange or red) depends on the probability of unauthorized access session. As a result, the mobile devices that have level 3 (red statue) will be directly assigned to another table named as Black List Table and it will be denied access session. So, the proxy notifies the hypervisor of malicious detection and sends the order to the hypervisor to un-deploy all VMs and communication channel of the mobile device. Figure 3.7 shows the secure hypervisor policy included in the proposed proxy.

```
Repeat
    Ask the user identifier and their secret key of the current session
    If secure proxy finds the user identifier and their secret key are correct then
        A new virtual machine is allocated in the cloud host;
    else
        Increase the probability of unauthorized access session;
    If the probability of unauthorized access session greater than the first probability threshold then
        Set the user in a  medium authorization access category (i.e., orange trust level);
    If the probability of unauthorized access session greater than the second probability threshold then
        Set the user in a third authorization access level (i.e., red trust level);
    If the probability of unauthorized access session greater than the third probability threshold then
        Insert the user in a blacklist and deny the status of the user;
Until (the status  of the user is denied or out of sessions time)
 If the status is denied then
    Send notification to the hypervisor
    Stopped  all the user VM's machines
    Stopped  all communication channels of each VM's machines of the user
    Send notifications to other mobile users
 EndIf
```

Figure 3.7. Secure Hypervisor Policy

3.3.3 VMs communication Policy

When a given VM demand to establish a communication channel with another VM. The target VM asks the source VM to provide VM secret keys. In the same time when sending the data, the target VM continues to send its secret keys to the source VM. Otherwise, degrade VM to a lower level (orange or red) depends on the probability of unauthorized communication. As a result, the VM that has level 3 (red statue) will be directly assigned to another table named as Local Black List Table and make access communication denied. So, the VM notifies the hypervisor to malicious VM detection, and so on to Proxy and increase the probability of unauthorized access.

If a probability of unauthorized access exceeds the probability threshold, then the proxy sends the order to the hypervisor to undeploy all VMs and communication channel of the mobile device. Figure 3.8 presents the secure virtual machines communication policy included in the proposed proxy.

```
Repeat
    Ask the target virtual machine identifier and its communication key
    If secure proxy finds the virtual machine identifier and its communication key are correct then
         Establish a new communication channel among VMs and start providing data
   else
         Increase the probability of unauthorized communication;
    If the probability of unauthorized communication greater than the first probability threshold then
        Set the VM in a  medium authorization access category (i.e., orange trust level);
    If the probability of unauthorized communication greater than the second probability threshold
then
        Set the VM in a  low authorization access category (i.e., red trust level);
    If the probability of unauthorized communication greater than the third probability threshold
then
        Insert VM in a blacklist  and deny the access communication
Until (the status  of the user is denied or out of communication)
If access session denied then
     Send notification to the proxy module
     Stopped  all communication channels of the target VM
     Increase the probability of unauthorized access
     If probability of unauthorized access greater than probability threshold then
        Insert the VM target in a black list and deny the status of communication channel
    EndIf
EndIf
```

Figure 3.8. Secure VMs Communication Policy

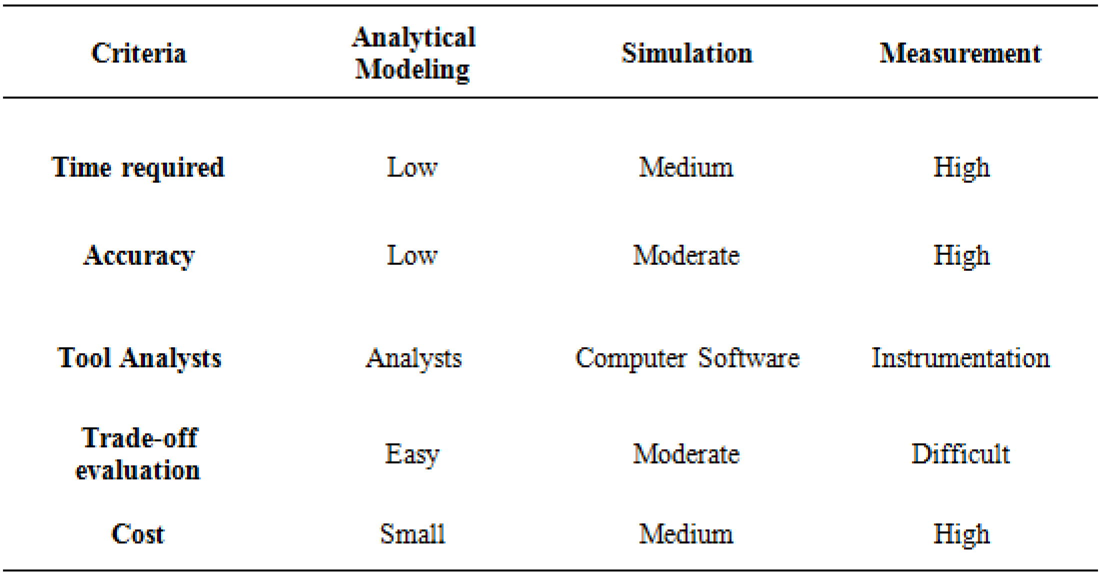
3.4 Performance Evaluation

The DS-II stage focuses on the evaluation part of the designed mechanisms and approach. For any research, the evaluation plays a crucial role to evaluate the work proposed in terms of performance and effectiveness. Three methods named as analytical modeling, simulation and measurements have proposed by [144] to evaluate the approach performance.

3.4.1 Different techniques of performance evaluation

It is very important to determine the evaluation performance technique for any research work [144]. Table 3.1 shows the comparison between methods of research, evaluation and describes the characteristic of each technique.

Table 3.1
Comparison of Performance Evaluation Techniques (Adopted from [144] *)*



Criteria	Analytical Modeling	Simulation	Measurement
Time required	Low	Medium	High
Accuracy	Low	Moderate	High
Tool Analysts	Analysts	Computer Software	Instrumentation
Trade-off evaluation	Easy	Moderate	Difficult
Cost	Small	Medium	High

3.4.1.1 Analytical model

An analytical model is mathematical modeling that uses many mathematic equations to analyze and evaluate the performance of the system [145]. The mathematical model is analyzed using programming software, which interprets the operations by availing the working relationships within the system. The outcomes of the software program can be presented by using graphical representation or symbolization. The users are able to regulate the system's conditions by changing the parameters of the software. This technique is preferably used in the case of study is unsafe or rare in real life, which leads to better understanding the primary result of the system before implementing the program. From Table 3.1, the Analytical Model technique has disadvantage related to low accuracy. However, it has many benefits regarding the low cost, easy in trade-off evaluation and less time required compared to simulation and measurement techniques.

3.4.1.2 Measurement

Measurement is a performance evaluation technique that uses an implementation or test-beds to conduct any research project such as network, cloud. This approach has a disadvantage in term of high cost due to the real equipment needed [139, 137]. Moreover, most of the measurements are habitually not repeatable and generally studying little scenarios. For instance, in cloud computing, many commercial IT organizations have launched real testbeds such as Amazon EC2/S3, PlanetLab, EmuLab, IBM/Google cluster to conduct a real cloud project. However, these test-beds are only supported such aspects of distributed computing and are not open source for general researchers [147].

3.4.1.3 Simulation

From Table 3.1, it is clear that simulation is a suitable method for the performance evaluation of the research work. Simulation is the method that commonly used for dealing with the various models of the real system using such software of simulation to emulate the programs' characteristic and operations. Furthermore, many reasons have been taken into account for preferring the simulation in terms of repeatable, scalable and controllable environment. The simulation allows the researchers to do repeatable experiments to get the highest accuracy level of the performance demanded [148]. In this research, the simulation method is selected to evaluate and measure the performance of the proposed approach.

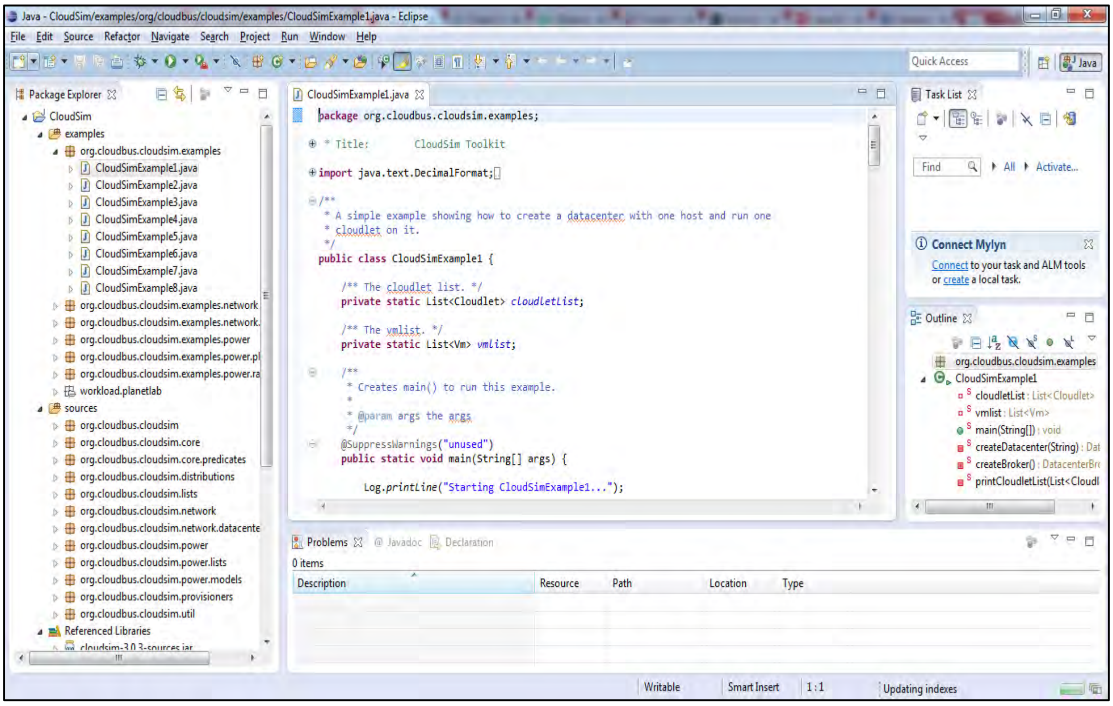
3.4.2 Evaluation Environment

In our work, we are going to evaluate the security and privacy of users' data while is interacted or deployed in virtual machines for processing in MCC environment. The specific elements of our study are the integrity and confidentiality of user sensitive data. The integrity means that unauthorized VM cannot modify the mobile user's data deployed on thin virtual machines. Moreover, confidentially means the encryption and decryption of users 'sensitive data. So, the work is specifically concentrated on the integrity of mobile users' data and the encryption/decryption of session keys.

3.4.2.1 NetworkCloudSim

Cloud computing is an advanced technology that provides services to the end-users in pay as you use manner. The application services have many requirements that need to manage such as configuration, complex provisioning policies, and deployment while they located on the cloud computing infrastructure. The performance, evaluating of such allocation and provisioning policies, power consumption model, workload model in a repetitive way is considered as very difficult task especially under different types of system and user configuration.

NetworkCloudSim is able to model distributed application with communication aspect while the datacenters process tasks. Mainly, the NetworkCloudSim considers all tasks as first-class entities called network cloudlets integrates both computation and communication. NetworkCloudSim is an extended layer of the CloudSim simulator which provides the network between datacenters but not inside them. In CloudSim, each VM connected to other VM without any details, but in NetworkCloudSim, a VM is connected to other VMs by switches: root, aggregate, and edge which give a real network model inside the datacenter. Figure 3.9 shows NetworkCloudSim Package Integrated with Eclipse.



Java - CloudSim/examples/org/cloudbus/cloudsim/examples/CloudSimExample1.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java
Package Explorer
CloudSim
examples
org.cloudbus.cloudsim.examples
CloudSimExample1.java
CloudSimExample2.java
CloudSimExample3.java
CloudSimExample4.java
CloudSimExample5.java
CloudSimExample6.java
CloudSimExample7.java
CloudSimExample8.java
org.cloudbus.cloudsim.examples.network
org.cloudbus.cloudsim.examples.network.
org.cloudbus.cloudsim.examples.power
org.cloudbus.cloudsim.examples.power.pl
org.cloudbus.cloudsim.examples.power.ra
workload.planetlab
sources
org.cloudbus.cloudsim
org.cloudbus.cloudsim.core
org.cloudbus.cloudsim.core.predicates
org.cloudbus.cloudsim.distributions
org.cloudbus.cloudsim.lists
org.cloudbus.cloudsim.network
org.cloudbus.cloudsim.network.datacente
org.cloudbus.cloudsim.power
org.cloudbus.cloudsim.power.lists
org.cloudbus.cloudsim.power.models
org.cloudbus.cloudsim.provisioners
org.cloudbus.cloudsim.util
Referenced Libraries
cloudsim-3.0.3-sources.jar
CloudSimExample1.java
Task List
Find
All
Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
org.cloudbus.cloudsim.examples
CloudSimExample1
cloudletList : List<Cloudlet>
vmlist : List<Vm>
main(String[]) : void
createDatacenter(String) : Dat
createBroker() : DatacenterBr
printCloudletList(List<Cloudl
Problems
Javadoc
Declaration
0 items
Description
Resource
Path
Location
Type
Writable
Smart Insert
1 : 1
Updating indexes

```
package org.cloudbus.cloudsim.examples;

 * Title:        CloudSim Toolkit

import java.text.DecimalFormat;

/**
 * A simple example showing how to create a datacenter with one host and run one
 * cloudlet on it.
 */
public class CloudSimExample1 {

    /** The cloudlet list. */
    private static List<Cloudlet> cloudletList;

    /** The vmlist. */
    private static List<Vm> vmlist;

    /**
     * Creates main() to run this example.
     *
     * @param args the args
     */
    @SuppressWarnings("unused")
    public static void main(String[] args) {

        Log.printLine("Starting CloudSimExample1...");
```

Figure 3.9. NetworkCloudSim Package Integrated Eclipse

3.4.2.2 NetworkCloudSim Architecture

We have chosen NetworkCloudSim because of its ability to codify the behavior of any complex cloud system with an ordered sequence of various events in time. This is what it calls a discrete event simulation. Moreover, it has many features that made us develop our simulation environment on top of it. In this section, we describe the functionality of various NetworkCloudSim layers. Figure 3.10 shows NetworkCloudSim Architecture.

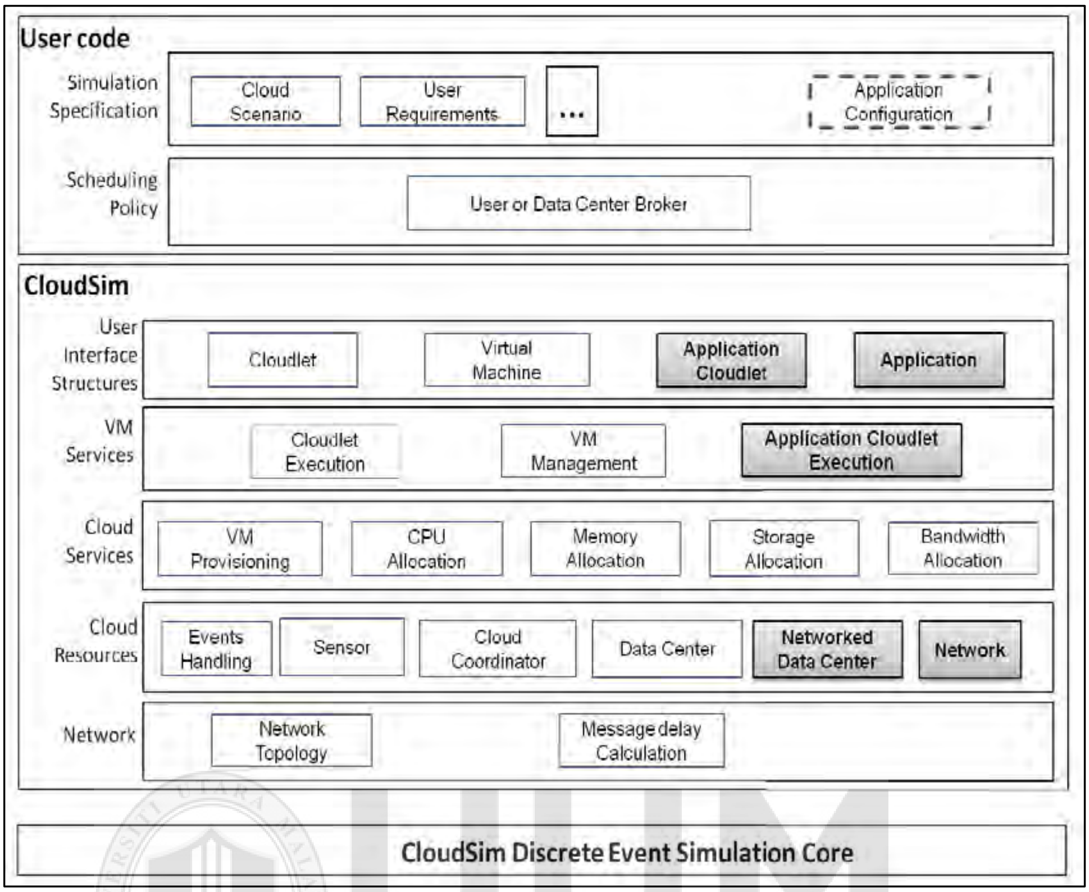


*Figure 3.10***.** NetworkCloudSim Architecture

The NetworkCloudSim is created based on CloudSim architecture. Some elements are added (dark boxes) to enable simulating applications with communicating tasks deployed on the cloud environment. The bottom layer of CloudSim treats the interaction and communication between the CloudSim elements and entities. The second layer is consisting of many sub-layers model the main components of the cloud computing such as datacenters, servers and other cloud resources that help to design infrastructure as cloud service. VM services and cloud services allow designing virtual machines provisioning and scheduling policies. The top layer permits cloud users to define and configure their simulation characteristic scenarios to validate their algorithms.

NetworkCloudSim has incorporated many components, which make the users model very complex many tiers applications that need more than one processor with internal network resources within datacenters in order to simulate realistic scenarios compared to modest modeling of CloudSim simulator. Those components are focusing on realizing two main issues: Application models and network models.

For the applications models, NeworkCloudlet class added to CloudSim to provide users with communicating tasks aspect to model their applications. Each NeworkCloudlet consists of Tasks processing in different stages of computation and communication. Another class added called AppCLoudlet that represents a several communicating NeworkCloudlet instance where each instance run in a specific virtual machine with communication and computing phases.

For the network models, NetworkCloudSim enables to model realistic network topologies inside the datacenters, not only between datacenters like the CloudSim. Many applications exchange data through the network may affected by data latency, which negatively reduces the quality of service delivered to the cloud users. Network CloudSim allows modeling data latency and bandwidth to analyze the performance of any applications. Switch, HostPacket and NetworkPacket are the main classes added by NetworkCloudSim.

3.4.3 Performance Metrics

Performance metrics is the aspect or context that researchers would investigate and target. For this research work, the performance metrics are ranging on coverage, efficiency, security management time. The mathematical equations and definitions of the Performance Metrics are elaborated in detail in chapter four.

Attack rate: In order to minimize the attacks rate and protecting the mobile user sensitive data from being retrieved, we need to reduce the attacks rate as much as we can: the attack rate has two pre-known sub-metrics namely: Attacker's efficiency and Attacker's coverage [12, 83, 142]. The equation for calculating these metrics is:

1- Efficiency: The attacker's efficiency is defined as the gain divided by the cost. To clarify more, the number of malicious VM that succeeds to co-locate with the target, divided by the total number of VMs launched by the attacker at time *t*.

$$Efficiency(Attacker, t\ ) = \frac{\left|Servers|SuccTarget\, VM(Attaker,t)|\right|}{|VM(Attaker,t)|} \qquad (3.1)$$

When to VMs attacker co-locate with the same target, it not counted two success co-location but only as one co-location. This Efficiency is only for preventing co-residence issue. However, it does not support the co-remote VMs (communication and interactions between VMs). The detailed of the efficiency of attack will be presented in chapter four.

2- Coverage: The attacker's coverage metric is considered another main criterion to evaluate the success of malicious attacker in allocating its VMs. The coverage is the number of malicious VMs that succeeds to co-locate with target, divided by the number of targets launched by legitimate user at time *t*.

$$Coverage(Attacker, t\ ) = \frac{|SuccTarget\, VM(Attacker,t)|}{|Target(Attaker,t)|} \qquad (3.2)$$

This Coverage is only for preventing the co-residence issue. However, it does not support the co-remote VMs (communication and interactions between VMs). The detailed of the coverage of attack will be presented in chapter four.

3- Normalized Execution time: defined as the time needed for checking the VMs that access the cloud hosts whether attackers or legitimates, and VMs interaction whether legal or malicious. Devided by is the maximum checking time that can be spent for security time verification.

$$T_{excution} = T_{security\_VM\_access} + T_{security\_VM\_Interaction} / Max\ (T_{security\_VM\_access} + T_{security\_VM\_Interaction}) \qquad (3.3)$$

Chapter 4 four will present the detailed of the whole metrics.

3.5 Summary

This chapter has presented a detailed research methodology that was used to accomplish the research objectives. This research aims to come up with a security approach in MCC environment to support sensitive data protection localized on the virtual machines. The chapter started with the overall research methodology where the main steps involved were briefly described. The fundamental phases used in achieving the research objectives were highlighted and discussed. We have provided the conceptual model of the proposed secure proxy and its policies after the examination of the research area. Here by criticizing the existing solutions, a sufficient understand of the current research issues was determined. Moreover, we proposed and detailed the development approach process of our proposed research work that uses three security policies: Mobile user control access policy, Hypervisor policy and VMs communication policy. The chapter then provided the performance evaluation section, which focuses on the evaluation step of the proposed approach. At last, the performance metrics used for proving the developed proxy based approach were briefly described in terms of Attacker's efficiency, Attacker's coverage and execution time.

CHAPTER FOUR

PROXY-3S: A NEW SECURITY POLICIES-BASED PROXY FOR EFFICIENT DISTRIBUTED VIRTUAL MACHINES IN MOBILE CLOUD

4.1 Introduction

This chapter presents a novel cloud proxy called Three policies Secure cloud Proxy (Proxy-3S) that combines three powerful security policies: VMs user's access control, VMs secure allocation and VMs secure communication. The approach aims to keep the distributed virtual machines safe in different servers in the cloud. It enhances the robustness and grants access authorization to permit intensive distributed applications tasks on the cloud or mobile devices while processing and communicating private information between each other. Furthermore, an algorithm that enables secure communication among distributed VMs and protection of sensitive data in VMs on the cloud is proposed.

The chapter is organized as follows. Section 4.1 gives the introduction of the chapter, which highlights the importance of the Proxy-3S proposed on securing the data on the mobile cloud environment. Section 4.2 provides an explanation on security metrics: efficiency, coverage, and execution time that are adopted in the proposed approach in order to study the performance of defending attacks and their execution time pertaining different VM allocation policies without taking in consideration our proposed solution (Normal situation of cloud Data processing). Section 4.3 describes the proposed approach with the general architecture and functional model of the cloud security system. The section provides details regarding the three secure polices of the approach.

Problem definition when an attacker intends to communicate or co-locate with VM users and improved security metrics are included. Section 4.4 gives detailed algorithms integrated into Proxy-3S based on secured exchanged hashed Diffie-Hellman Keys. Last Section 4.5 concludes the chapter by giving a summary of the complete chapter.

4.2 Co-Residency Attacks and Performance Measures

This section provides a detailed clarification of the co-resident attack that is performed inside servers on the cloud. The understanding of the co-resident problem and different goals of attackers either to target a clear set of VMs or have no specific target VMs to steal their sensitive data. The performance metrics to evaluate the attack are detailed precisely in order to be selected as the basic metrics in our mechanism. The comprehension of the whole problem leads us to adopt our solution for distributed VMs communication.

4.2.1 Co-Residency Attacks

There are two goals that an attacker can achieve through co-resident attacks. Firstly, the attacker has no clear set of target VMs. In this attack scenario, the attacker goal is to get an unfair share of the cloud resources [24, 143]. In Amazon EC2 platform, an attack case is revealed when some virtual machines consume more processing capacity and CPU time where it supposed to follow the hypervisor instructions and conditions which lead to reduce and steal the cloud service [151]. Secondly, the theft has main interest in the specific target of legitimate VMs. The attacker spread the malicious VMs with the targeted VMs in order to co-locate with them in the same server [145, 146]. After the co-location is established, the Attacker VMs can obtain sensitive data by constructing various side channels.

Many ways can be used to detect co-location VMs, such as network measurement. The attacker proceeds with TCP traceroute operation when it intends to obtain the IP address of a VM’s management Domain 0 (Dom 0). Two VM’s are considered as co-resident if their IP addresses Dom0 are the same. In the current cloud platform, this co-residence technique detection has been blocked.

In order to achieve co-location, the attacker either spread as many VMs as possible on the cloud (brute-force strategy) or take the opportunity when multiple VMs launched at the same time (these VMs are allocated to the same host). An attacker can co-locate with target VMs on the same server by performing Prime-Probe cache timing measurement. In Prime-Probe cache timing, the technique is used for constructing coarse-grained side cache channels where the attacker waits for predefined time to fill and read from same memory cache sets that are used by other users. If the users use much cache (many activities), the attacker’s data will be excluded from the cache.

Fine-grained side channels can be constructed to breach the confidentiality of tenants on cloud platforms. User sensitive information, computation tasks, and data must be isolated from either the malicious cloud provider or users.

4.2.2 Normalized Performance Measures

Generally, a defending mechanism should ensure the protection of sensitive information from any alteration and attackers intrusions. The proposed secure approach should be easily adopted in a cloud computing platform for preventing any fundamental changes that bring cost issues in term of complexity and security checking time. Defending security threats by sacrificing others metrics like load balancing and power consumption can affect the usability and performance of the cloud platform provider as well as may present new security problems.

VM allocation policy is one technique that is the focus of this thesis. In the best the researcher of knowledge, only two works have considered secure VM allocation as the best solution to secure private information [30,12]. The VMs allocation policy mainly makes it hard for attackers to achieve co-resident. The research adopts the metrics of [12] for evaluating the security aspect of the proposed approach.

Obviously, it is better to explain the security metrics (efficiency, coverage) of [12] and then introduce our theory contribution regarding these metrics for adoption in VMs co-remote problem. The security metrics are defined as efficiency, coverage and execution time which are discussed in the next sub-section.

1- Efficiency

Spreading the minimum malicious VMs for achieving co-location is considered as the best scenario for cloud VMs attacker. The efficiency is defined as the gain divided by the cost. To clarify more, the number of malicious VM that succeeds to co-locate with the target, then it is divided by the total number of VMs launched by the attacker.

$$Efficiency\ (VM(A,t)) = \frac{|Servers|SuccTarget\ VM(A,t)||}{|VM(A,t)|} \qquad (4.1)$$

Where:

- A: refers to the attacker.
- $VM(A,t)$: refers to VMs of A
- $Servers|SuccTarget\ VM(A,t)|$: refers to the total of servers of VMs attackers which succeed to co-locate with the target, *t* is the exact time.
- $VM(A,t)$: is the total VMs launched by an attacker in time t.

When two VM attackers co-locate with the same target, it is not counted as two success co-location but only as one co-location. This Efficiency is only for preventing the co-residence issue. However, it does not support the co-remote VMs (communication and interactions between VMs).

2- Coverage

The coverage metric is considered another main criterion to evaluate the success of the malicious attacker in allocating its VMs. The coverage is the number of malicious VMs that succeeds to co-locate with the target, and then it is divided by the number of targets launched by the legitimate user.

$$Coverage(VM(A,t)) = \frac{|SuccTarget\ VM(A,t)|}{|Target(A,t)|} \qquad (4.2)$$

Where:

- *A:* refers to the attacker.
- $VM(A,t)$: refers to VMs of A.
- *SuccTarget* $VM(A,t)$*:* refers to the sum of VMs machine attackers which succeed to co-locate with the target, *t* is the exact time.
- $Target(A,t)$: is the sum of all VMs target in time *t.*

Figure 4.1 illustrates an example of computation of attacks efficiency and coverage of [12].

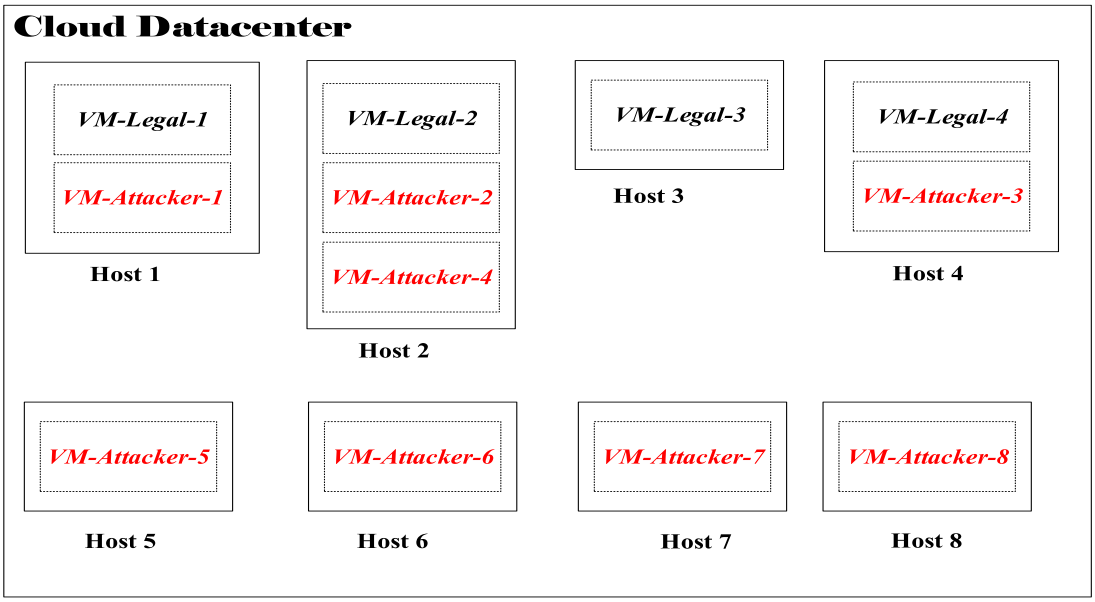


Figure 4.1. Computation example of attacks efficiency and coverage.

The efficiency here is 3/8 and the coverage is 3/4. The efficiency and coverage consider as probability values that vary between 0 and 1 continuously. Whereas, security execution time is considered as a continuous variable that measured with seconds or milliseconds and the value grows from 0 to a large time rate.

3- Normalized Execution Time

It is the total time taken by the security mechanism to finalize its processes. The time is from the time of user submission tasks until the time of the result delivered to the right users. The security execution time plays the main important role to deliver the cloud service to the client. If the value is positive, then it meets the expectation of cloud users. However, it affects performance and fails to meet user satisfaction. The processes of the mechanism are the time to check the VM and time to check the interaction between VMs.

The normalized execution time is computed as follow:

$T_{excution} = T_{security\_VM\_access} + T_{security\_VM\_Interaction} / Max\ (T_{security\_VM\_access} + T_{security\_VM\_Interaction})$ (4.3)

Where:

$T_{security\_VM\_access}$: refers to the average time needed to verify the security key correctness provided by a VM and the access time of VM.

$T_{security\_VM\_Interaction}$: refers to the average time needed to check the interaction legitimacy between VMs.

$Max\ (T_{security\_VM\_access} + T_{security\_VM\_Interaction})$: is the maximum checking time that can be spent for security time verification.

4.3 The Proposed Approach

The new cloud Three policies Security Proxy (Proxy-3S) approach helps to enhance the security of distributed VMs and enables the enforcement of corporate security policies for protecting the mobile user's sensitive information on the cloud. The novelty of the proposed approach provides: (1) a valuable balance between VM control access and good performance, scalability, robustness, and efficiency (2) a secure verification process and hierarchical protection of the sensitive data of distributed VMs from any malicious attackers in the cloud. The approach is based on three security policies that are collaborating to guarantee the user's sensitive data integrity and confidentiality from the input point (sender mobile device) to the output (receiver mobile device or cloud). The access to private VM's data is not public, thus we use Proxy-3S in order to enforce authorization access to the VMs on the cloud and give a sight of management of users' VMs. In order to reduce the cost overhead of the cloud resources, the user's data is processed inside those VMs. In addition, the result of data processing eventually returns to mobile devices.

4.3.1 General Architecture and Functional Model

The approach composes of two sub-sections, namely general architecture and functional model.

4.3.1.1 General Architecture

In this section, we propose the approach design to ensure the security of distributed VMs integrated into cloud architecture. Figure 4.2 depicts a general overview of security architecture. The mobile client connects to cloud services through secure Proxy-3S. The architecture consists of three components:

a. **Client**: this is the mobile application that is currently deployed on Smartphones, tablets or PCs.

b. **The Secure Cloud Proxy-3S**: the Proxy-3S is the core of our system that plays as the mandatory entity between the mobile application site and the cloud server. It also secures and manages the VMs that are authorized to handle sensitive data on the mobile cloud environment. When a mobile user request services from the cloud environment, it offloads the intensive processing tasks to the cloud and wait for the results. Since the secure proxy deploys the application. It runs the three security policies to identify the possible malicious attacks between the mobile user and the cloud or between collaborated VMs on the cloud. The Proxy-3S integrates mobile user control policy, virtual machine management protection policy and secure virtual machine communication policy into a single offering or unit.

- **Mobile User Control Access Policy Component**: this is the first component aiming to control the access of the mobile users' devices into the cloud. It only authorizes the permitted users to leverage cloud services. It consists of a list of authorized users with their encrypted password, the users' status that informs the proxy of the trust level of the users that facilitate access control and make decision authorization whether the user is allowed to access the cloud service or no (denied).
- **Virtual Machine Manager Protection Policy Component**: A cloud hypervisor is a software that creates, deletes and updates when needed the VMs on specific servers that they assign to the guest mobile user. In order to secure the hypervisor, the huge VMs need to manage silently or securely to avoid the deployment of malicious VMs. The policy based on proxy detection and order prevents the illegal VM to co-locate with legal VMs.
- **Secure Virtual Machines Communication Policy Component:** The VMs form a network VM while communicate and transfer the data with each other across multiple virtual channels on the datacenter. Secure communication is provided using symmetric and asymmetric cryptographic methods. The policy contains the VM communication information that helps ensure security compliance and secure communication to and from VMs. The management information includes the VMs originated by the mobile user, its identifier, the communication trust level and the current session with the encrypted password. This policy aims to secure the sensitive user's data confidentiality and privacy that exchanged among VMs.

c. **The Cloud:** it defines a network of VMs connected between each other by sending and receiving packets. It stores sensitive user's data and executes intensive tasks.

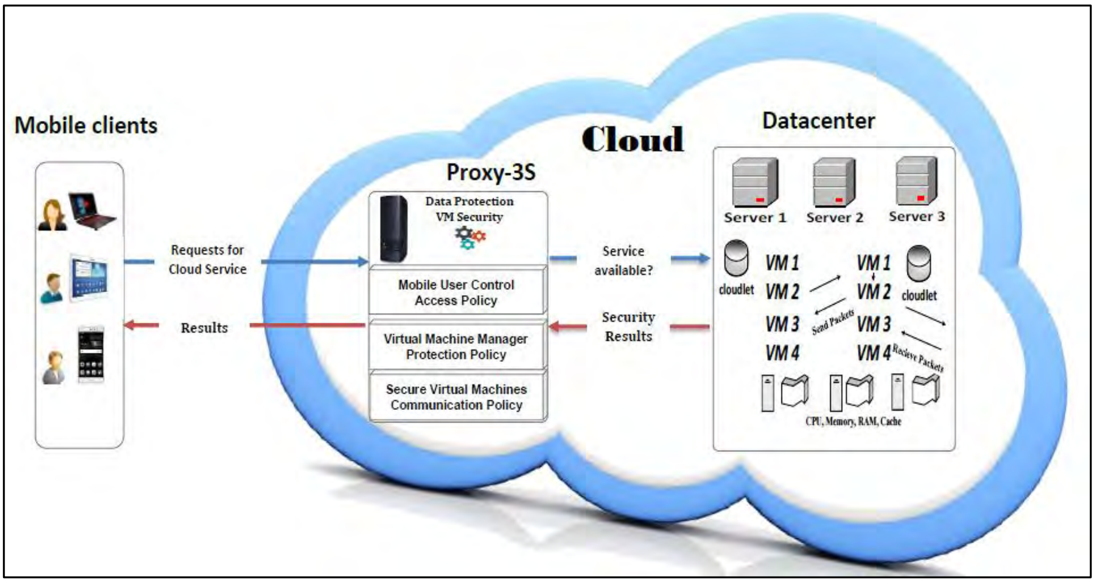


Figure 4.2. Overview of security architecture

In the next section, we will give the detail of the functional model of the system where mobile tenants intend to access the cloud service in order to process their intensive tasks on cloud servers.

4.3.1.2 Functional Model of the Security Cloud System

The objective of the proposed security system is to ensure that the user's mobile applications and their sensitive data are protected from unauthorized access and malicious VMs in the cloud. An overview of the functional model of the security system provided in the following Figure 4.3:

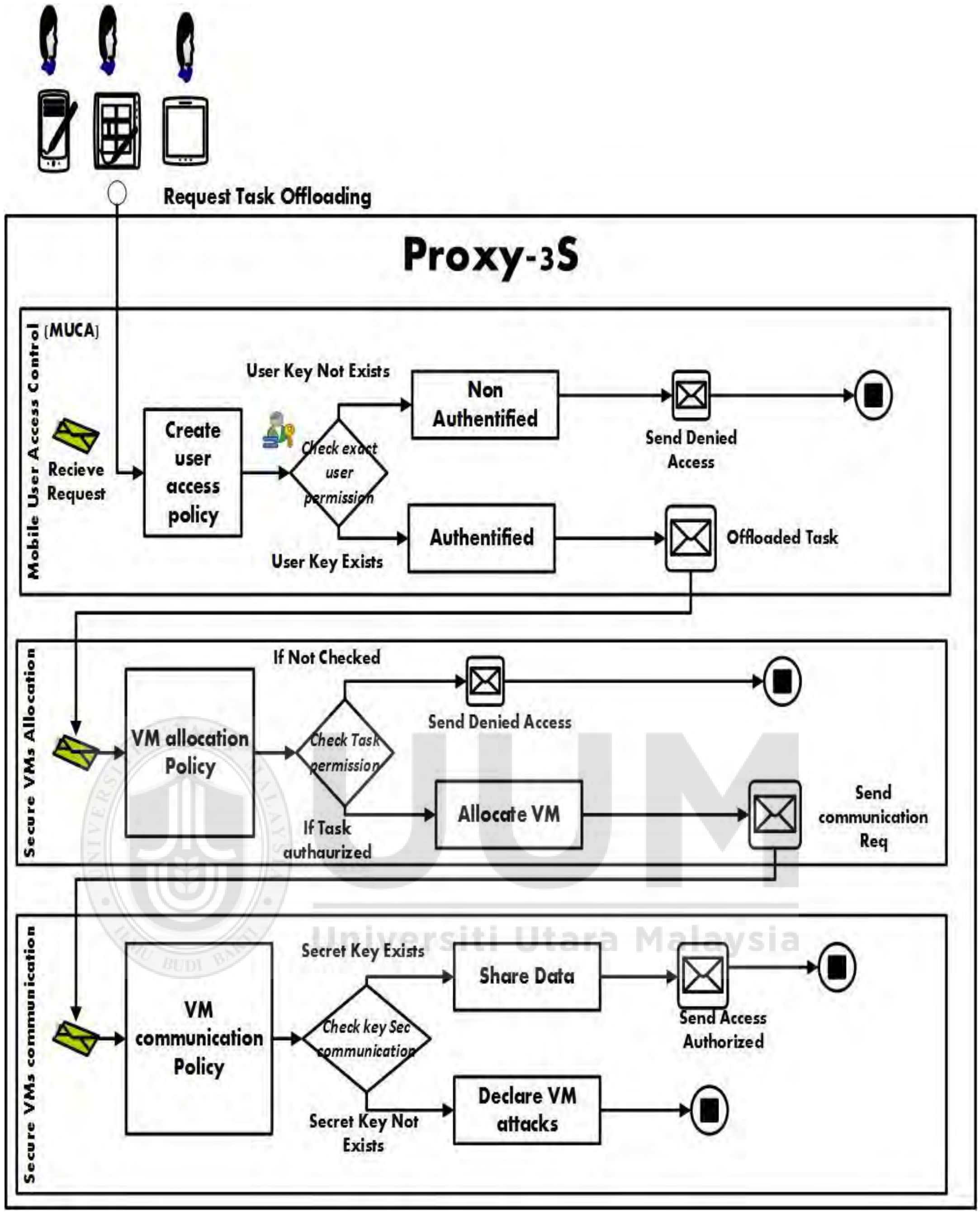


Figure 4.3. The functional model of security three policies Proxy-based approach

The proxy manages distributed mobile applications which run on different server operating systems either in mobile device or data center. It consists of various tasks that consider either heavy or light tasks. The mobile device can process the light tasks, but the intensive task must execute on cloud servers. The proxy ensures the security of managed intensive computational tasks and their sensitive data in the cloud.

The process ends when the system delivers results to the mobile user and intensive-tasks are finished. We detail the functional process as follows:

- Firstly, users request to deploy their intensive tasks on the cloud. They send their identifiers and passwords through a mobile device to the Proxy.
- Secondly, the proxy receives the user identifier and password of users. Then it calls the Mobile User Control Access Component (MUCA) in order to check the authentication of the user where the proxy must recognize the user at the first, otherwise the access is denied. If the user's credentials are correct, then the proxy sends an order for the hypervisor to allocate VMs for the user depending on various offloaded intensive tasks. Each VM is associated with a secret key, which allows it to control the VMs behavior by its VMs manager protection module where the VM can be legal or malicious. If the VM is legal, the module keeps it inside the server for processing the user task.
- Thirdly, the VM takes to delivering their obtained intermediate results to its other VMs is anchored in the cloud security proxy system and the associated Secure VMs Communication Component. The VMs deliver their obtained intermediate results to the secure proxy system. After that, the proxy shares secure results with other Secure VMs Communication Component. By using these components, we have checked the credibility of the VM that would communicate. If the VM is legal, the proxy allows the communication between the sender and receiver VM. Therefore, the tasks can share sensitive data between them.
- Finally, when intensive tasks are completed, the proxy sends the result to the mobile user and removes VMs that finish their tasks process from the host cloud.

4.4 Virtual Machines Communication Security in Cloud Computing

In this section, we present some improvements regarding the efficiency and coverage of external attacks compared to the work of [12, 83]**.** Then we describe the problem definition and an overview of our proposed approach.

4.4.1 Improved Efficiency and Coverage Metrics

Distributed mobile applications have many tasks deployed on cloud computing servers, and the tasks are performed in various VMs. When tasks communicate with each other, they exchange various forms of private information. In the cloud, data security and privacy of the distributed application have several threats that impact the application. An attacker can deploy malicious VMs in either the same or different server. Authors in [83] studied experimental deployments of VMs in different scenarios using different allocation strategies. The study develops co-residency security metrics called efficiency and coverage in order to increase difficulties to VM attackers to co-locate with legal VMs. However, these metrics have not been used in analyzing and evaluating VMs communication, whereby VMs are deployed on different hosts and communicate to exchange sensitive information. Therefore, we include these metrics in VMs communication attacks study. In the proposed approach, we define the remote co-location attack when having a successful VM's attacker that communicates with at least one of the target legal deployed on the different hosts. In work [12, 83], two metrics are proposed for detecting the attack, namely efficiency and coverage. We used these two metrics and improved in our approach as follows. (Table 4.1 details used notations):

Definition 1 (Efficiency): is the ratio of the number of malicious VM that are successfully co-located with the target, divided by to the total number of VMs attacker launched. Due to the exchanges between VMs in a distributed mobile application, while deployed in different hosts on the cloud, the efficiency metric of [83] is used and improved. Now, the efficiency is the ratio of the number of success attacker's co-location subtracted from the total number of newly detected attackers, divided by to the total number of VMs attacker and VMs attacker interaction. The efficiency metric for the remote co-located attacker "A" is defined as follows:

$$Efficiency(\left|Remote\_colocated(A,t)\right|) = \frac{|SuccTarget\ VM(A,t)| - NewDetected\_Attacker}{|VM\ (A,t)| + |VM\_interaction(A,t)|} \tag{4.4}$$

Where newly detected attackers are the total number of VMs attacker recognized as legal before the proxy detection (i.e. interaction between VM attacker and another VM attacker is not considered remote co-location or co-location).

Definition 2 (Coverage): is the number of VMs attackers that are successfully co-located with legal VMs divided by the target VMs (legal). Due to the interaction between VMs in distributed mobile applications that communicate between each other while deployed in different hosts on the cloud, the coverage metric of [83] is used and improved. Now, the coverage considers the security property relative to the data exchanges between two VMs (attacker and legal). It is defined as the ratio of the number of successful attacker's co-location subtracted from the total number of new detected attacker, divided to the total number of VMs legal and VMs legal interaction. The coverage metric for the remote co-located attacker "A" is defined as follows:

$$Coverage(\left|Remote\_colocated(A,t)\right|) = \frac{|SuccTarget\ VM(A,t)| - NewDetected\_Attacker}{|VM\ (L,t)| + |VM\_interaction(L,t)|} \tag{4.5}$$

Table 4.1.
Detailed notations regarding the security metrics

Notation	Description
D	The Datacenter
N	The total number of servers in Datacenters
A	Refers to attacker
L	Refers to legal VMs
Servers ({a set of VMs})	Servers that host the set of VMs

VM (L, t)	The set of VMs started by L at time t
VM (A, t)	Refer to VMs of A started during one attack at time t
Target(A)	The target set of VMs that A intends either to co-locate or to communicate with in time t, $Target(A) = \sum t\ VM(L, t)$, $\|Target\ (A)\| = T$
SuccTarget VM (A, t)	Refer to the sum of VMs machine attackers, which succeed to co-locate with the target, t is the exact time.
NewDetected_Attacker	The VM attacker that behaving like VM legal before the detection from the proxy. This attacker was co-located with VM attacker.
VM_interaction (A,t)	Refers to the total VMs attacker interactions channels with target VMs launched by an attacker in time t, both successful malicious communication and unsuccessful malicious communication.s
Succ_malicious_communication (A, t)	Refers to the sum of channels of VMs attackers which success to interact with the target VM, t is the exact time.
Unsucc_malicious_communication (A, t)	Refers to the sum of channels of VMs attackers which unsuccessful to interact with the target VM, t is the exact time.
VM_interaction (L, t)	Refers to the total VMs legal interactions channels with another VMs launched by a legal VM in time t, both successful legal communication and unsuccessful legal communication .
Succ_Legal_communication (L, t)	Refers to the sum of channels of VMs legal which success to communicate with the target VM, t is the exact time
Unsucc_Legal_communication (L, t)	Refers to the sum of channels of VMs legal which un-success to interact with the target VM, t is the exact time. Unsuccessful means that VM legal communicate with VM attacker that behaving as legal.

Figure 4.4 shows five legal VMs deployed on different hosts. The Attacker “A” starts seven VMs, four of which co-locate with three legal VMs and two remote VMs co-locate with two legal VMs. The proposed efficiency and coverage metrics that include the communication aspect among a set of legal and attackers VMs dispersed across various hosts are applied and evaluated.

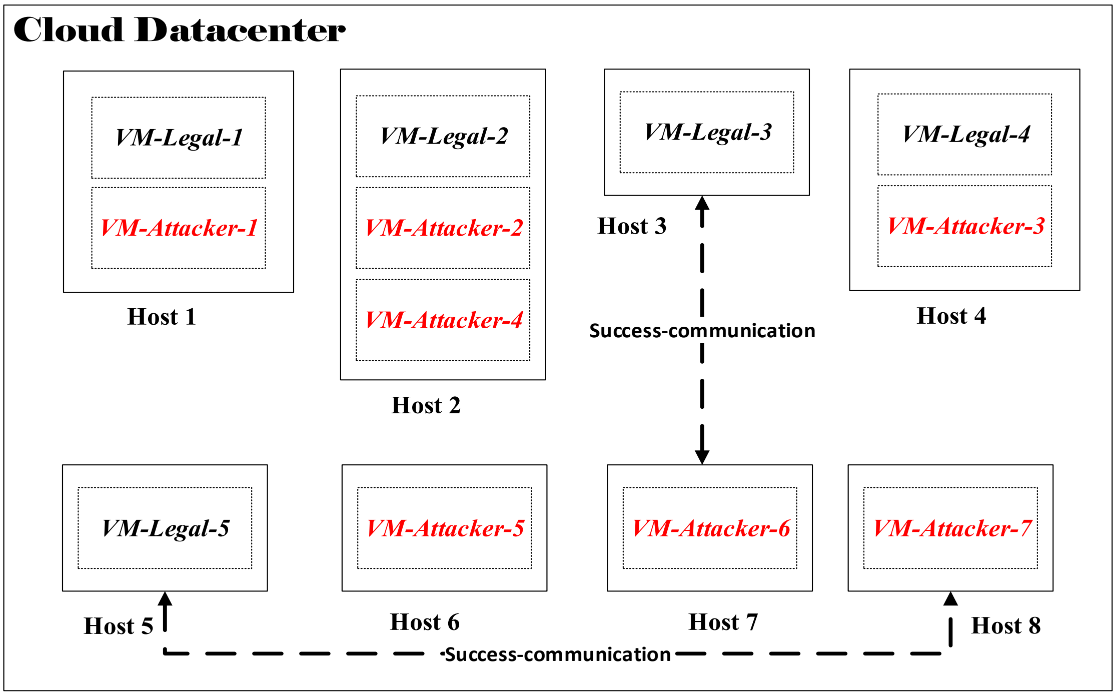


Figure 4.4. Computation example of attacks efficiency and coverage using the communication aspect.

The difference between this research and [12, 83] research work is that their policy does not provide an exhaustive solution since their metrics do not consider the malevolent communication attacks.

In this example, the success co-location of VMs attacker is 3 (success target) and new detected attacker is 0. For the VMs interaction, there is 2 VMs interaction, where it is considered as successful malicious communication launched by VMs attacker. Otherwise, the VMs attacker has no unsuccessful malicious communication. Regarding VMs legal, there is two interaction received by them, which are considered as unsuccessful legal communication. Whereas successful legal communication is equal to 0. Thus, the efficiency is 3/9 (instead of 3/7 in [83]), and the coverage is 3/7 (instead of 3/5in [83]).

4.4.2 Problem Definition and Security Modeling

Assume that several attackers intend to communicate or co-locate with legal VMs. Therefore, the main objective of an attacker is to maximize the coverage and/or efficiency rate. When the VM attacker communicates with a legal VM and both of them are located on different hosts, it is considered that VM attacker is co-located with VM legal. Hence, considering the communication aspect between VMs, the problem is how to reduce the efficiency and the coverage of the attacker as well as how to increase the number of users using the cloud servers without increasing the coverage and efficiency of attack. Then, by using our secure approach, we are able to measure both the attacker's efficiency and coverage to identify the remote co-resident attackers. The problem takes into account the following descriptions:

- DC = $\{dc_1, dc_2\ldots, dc_N\}$, denotes N datacenters of a cloud system. Each datacenter dc_i contains a set of M servers S = $\{S_1, S_2\ldots S_M\}$ having the same hardware resources.
- The set VM = $\{vm_1, vm_2\ldots, vm_P\}$ of P virtual machines. For each virtual machine $vm_{i\ (i=1,\ 2,\ldots p)}$ can be allocated to a server $S_{j\ (j=1,2,\ldots M)}$ by the projection X: V x U → S.
- The set U = $\{u_1, u_2,\ldots, u_k\}$ of k users. Each user can be categorized into two types: legal or attacker, denoted as L and A respectively, where for each legal user, a set of legal VMs started by L at time t denoted by VM (L, t) while for attacker a set of VM attackers started by A at time t denoted by VM (A, t).

We propose three security policies: (1) the first mobile user access control policy is based on hashed encrypted access keys to get a powerful user's access technique, (2) the second secure VM allocation policy is based on minimum CPU utilization and minimum channels communication with secure VM communication policy is based on secure communication channel access through hashed encrypted communication keys that offer high security once exchange sensitive data among VMs. These policies improve the efficiency and coverage of attack while the interaction of VMs deployed on different servers and communicate with each other. In the following figure, there are two main actors that Proxy-3S handles their information in order to provide access to service cloud or data in VMs.

The first actor is a mobile user. When the mobile user configures its session, he provides the MAC address that automatically retrieved from the OS of its mobile device D_{user} ①. When the Proxy-3S receives the MAC address, it generates a random secret number called Secret deployment Iteration “S_It_1” ② to use it as input with MAC address (second input) in key generation function G_{Ku} (Generation Key User). S_It_1 is a random number greater than 1000. The generation function G_{Ku} produces two different keys: a private user access key Ku_{pv} and a public user access key Ku_{pb} ③. The current keys is stored in the key registry by the proxy. The private key is unique for each user, and is sent once (or updated when needed as a new session) when a user asking for deploying a task ④⑤. In order to authorize a mobile user to deploy a task, a private user access key is used and the signed MAC address $Sig@Mac_d$ is constructed ⑥. This signed address is sent together with the corresponding task identifier to the proxy ⑦. On the other hand, regarding the proxy side to accept a task deployment from a user, it must checks the signed MAC address ⑥. Then, in the cloud and using the symmetric properties of Hash-Diffie Hellman encryption, the signature must be verified using only Hash MAC address stored in the key registry⑧. If the signature is checked, the proxy asks the hypervisor to deploy the user’s task⑨. The diagram flow is described in the top of Figure 4.5.

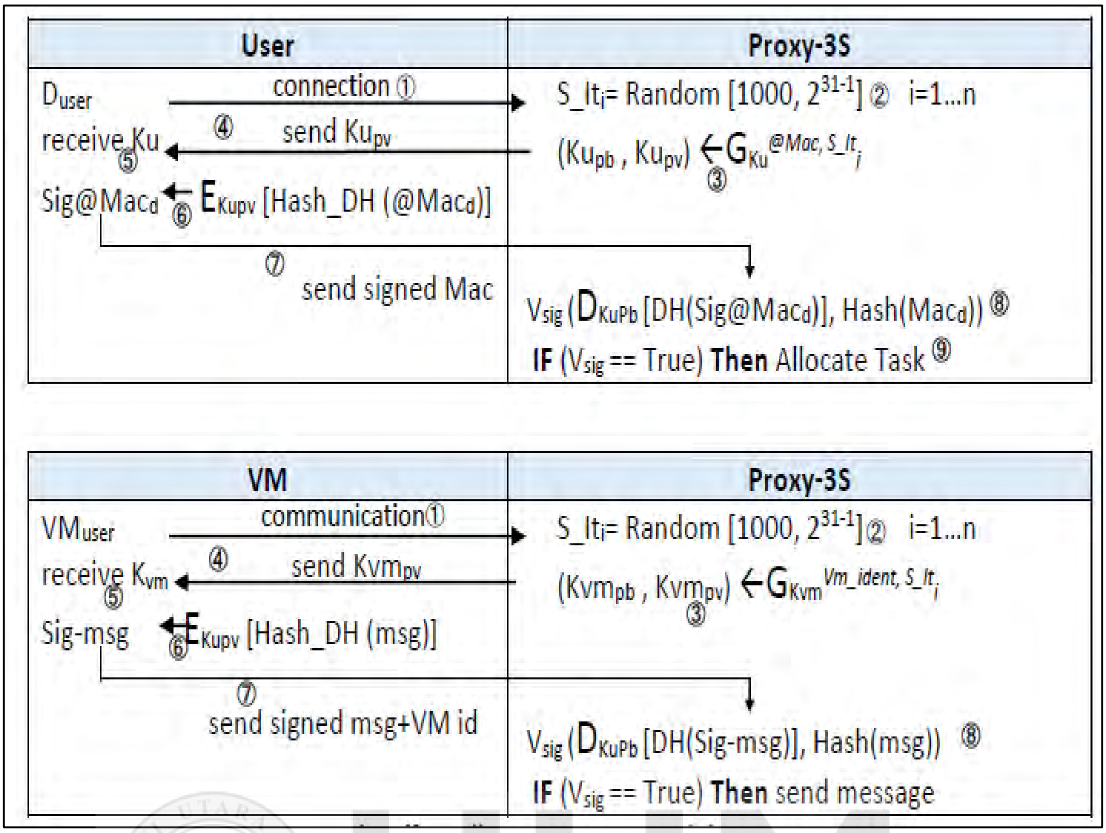

User
Proxy-3S
Duser
connection ①
S_Iti= Random [1000, 2^31-1] ② i=1…n
receive Ku ⑤
④ send Kupv
(Kupb , Kupv) ← GKu @Mac, S_Iti ③
Sig@Macd ← ⑥ EKupv [Hash_DH (@Macd)]
⑦ send signed Mac
Vsig (DKuPb [DH(Sig@Macd)], Hash(Macd)) ⑧
IF (Vsig == True) Then Allocate Task ⑨
VM
Proxy-3S
VMuser
communication①
S_Iti= Random [1000, 2^31-1] ② i=1…n
receive Kvm ⑤
④ send Kvmpv
(Kvmpb , Kvmpv) ← GKvm Vm_ident, S_Iti ③
Sig-msg ← ⑥ EKupv [Hash_DH (msg)]
⑦ send signed msg+VM id
Vsig (DKuPb [DH(Sig-msg)], Hash(msg)) ⑧
IF (Vsig == True) Then send message

Figure 4.5. Hash-Diffie Hellman encryption and decryption process

The second actor is a VM. When the VM intends to communicate with another VM whether allocated on the same host or different host. It provides its VM identifier that proxy receives it with the request of communication VM_{user} ①. The proxy generates a random secret number of deployment Iteration “S_It_1” ② to use it as input with VM identifier in key generation function G_{Ku} (Generation Key User). The generation function G_{Ku} produces two different keys: a private VM access key Kvm_{pv} and a public VM access key Kvm_{pv} ③. The current VM keys are stored in the VMs key registry of the proxy. The private key is updated when needed as new communication session ④⑤. The message that VM intend to send is Hashed and encrypted to produce a signed message (hard to decrypt by the adversary) ⑥. Further, the signed message with encrypted identifier is sent to the proxy and the proxy decrypted and compare with the existing Hash message. If the signature message is correct then the message will be authorized to be sent for VM receiver ⑦⑧⑨. The diagram flow is described at the bottom of Figure 4.5.

The security of the proposed technique depends on the session identifier in the generation phases after encrypting with the Hash-Diffie Hellman. In addition, each attacker VM attempt to extract the sensitive data must know the same session identifier key S_It established in the generation function. Otherwise, the VM attackers will be denied. The key size of the proposed scheme is large enough to resist brute-force attacks. In other words, if an attacker tries to find out the session key used for accessing VM's data, the attacker needs to perform a number of operations of (1000 x 2^{31}-1)! To this end, we could conclude that the proposed model provides high confidentiality for the sensitive VM's data.

4.5 Proxy-based security policies details

We present our proxy-based security policies over distributed VMs that run on different hosts on the cloud (see Figure 4.6). The objective of these policies is to combine the Hash-Diffie Hellman and secure VM algorithms to automatically protecting users' data integrity and confidentiality in the cloud. Considering the high capabilities of defending against the confidentiality and integrity of data attacks, we will often choose a more practical Hash-Diffie Hellman method. The research used the long-term usability of the SHA-256 algorithm. This is the strongest hashing algorithm used by popular cryptocurrencies such as Bitcoin, Bitcoin Cash, Counterparty, MazaCoin [140]. So far, SHA-256 secure hashing algorithm is never compromised in any way as well as its implementation is easier than the predecessor hash function SHA-1 and provides a hash that cannot be broken. Hence, this motivates controlling the mobile user's access and protecting VM's data by SHA-256 Diffie-Hellman in our approach. National Institute of Standards and Technology has provided various security check test to verify the correctness of data integrity in communication aspect [154]. Proxy-3S consists of three policies collaborating for ensuring the protection of distributed VMs and sensitive data integrity and confidentiality. Proxy-3S takes the user's requests and the information of the devices as inputs. Once the whole requested tasks are executed securely and legally, the result is sent back to the user. This is done on the following three main steps:

- Step1 (Figure 4.6): the proxy calls MUAC Authentication policy (see Algorithm 1) to provide authorization decision for users to get access to the proxy. This is done by checking the user identifier and password with the available authorized users.
- Step2 (Figure 4.6): the proxy calls Secure VM Allocation policy (see Algorithm 2) to secure VMs allocation against malicious attacks based on various extended allocation strategies (most, least and random). The VMs allocation policy checks and compares the secret key with the available secret keys required for the VM allocation.
- Step3 (Figure 4.6): the proxy calls VM Communication Protection policy (see Algorithm 4) to protect the user's sensitive data and access requests on the handled VMs as well as during the communication of the connected VMs.

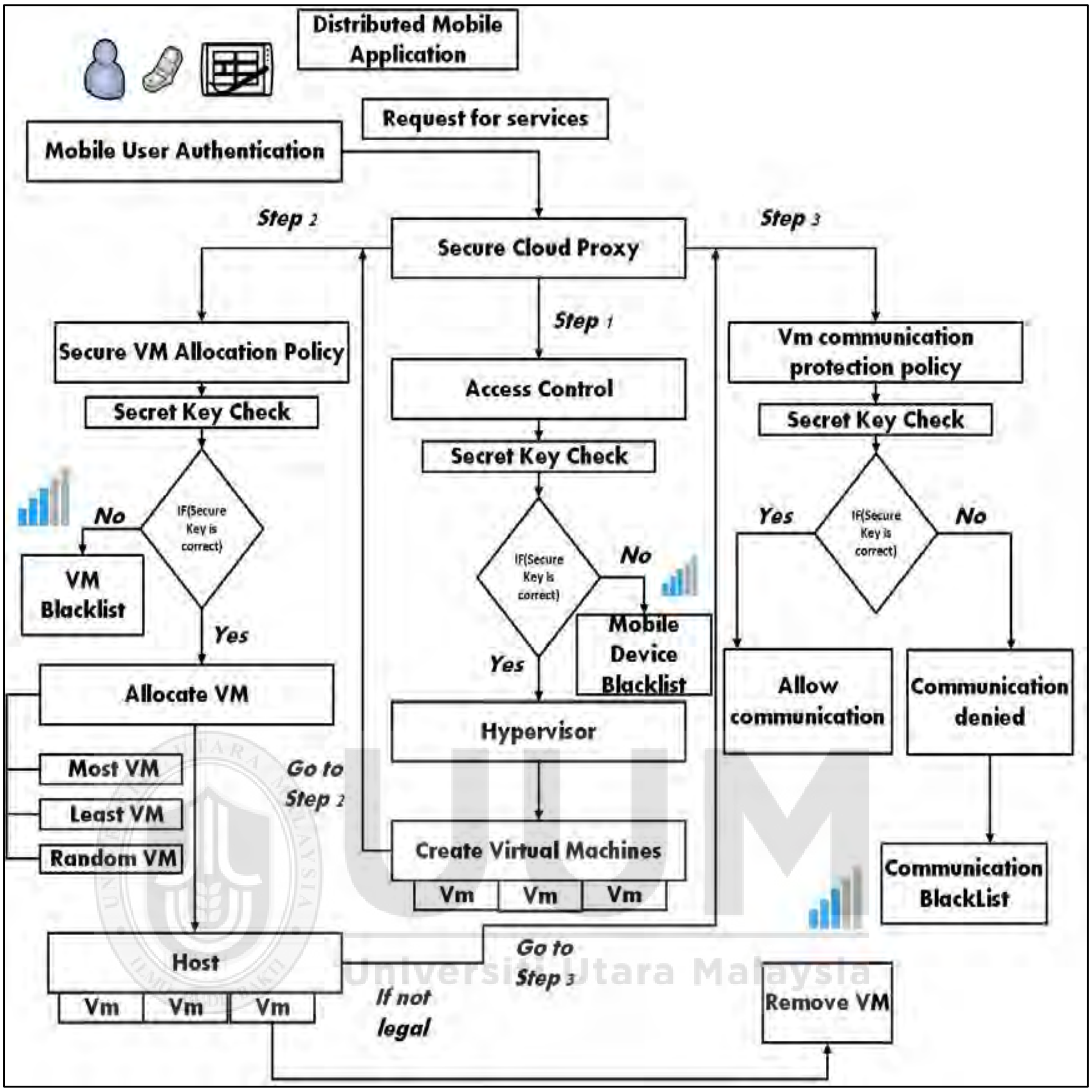
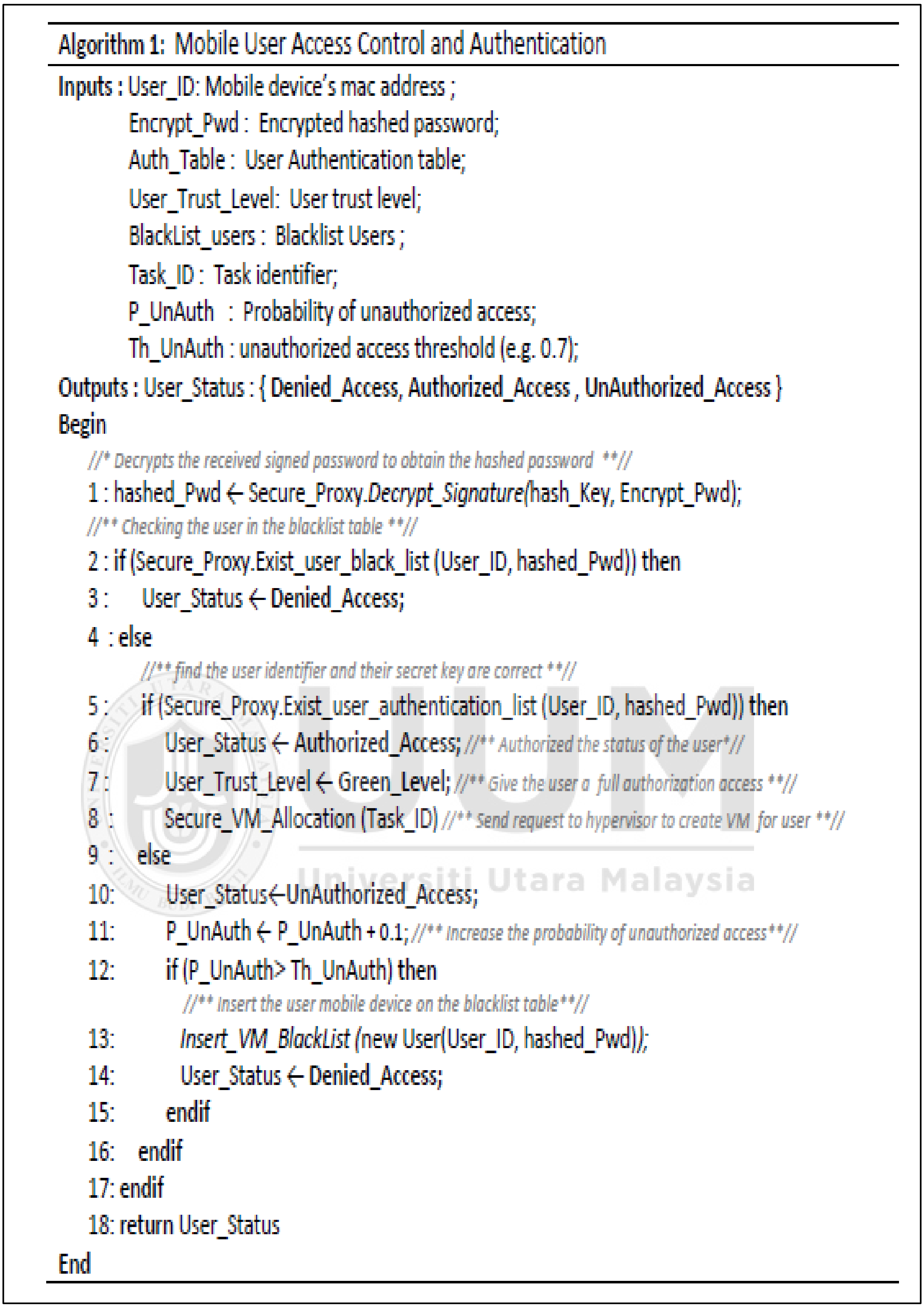


Figure 4.6. Proxy-based policies details.

4.5.1 Mobile User Access Control and Authentication Algorithm

Algorithm 1 depicts the user authentication policy to secure the user's sensitive data on the cloud.



```
Algorithm 1: Mobile User Access Control and Authentication
Inputs : User_ID: Mobile device's mac address ;
         Encrypt_Pwd : Encrypted hashed password;
         Auth_Table : User Authentication table;
         User_Trust_Level: User trust level;
         BlackList_users : Blacklist Users ;
         Task_ID : Task identifier;
         P_UnAuth  : Probability of unauthorized access;
         Th_UnAuth : unauthorized access threshold (e.g. 0.7);
Outputs : User_Status : { Denied_Access, Authorized_Access , UnAuthorized_Access }
Begin
   //* Decrypts the received signed password to obtain the hashed password **//
   hashed_Pwd ← Secure_Proxy.Decrypt_Signature(hash_Key, Encrypt_Pwd);
   //** Checking the user in the blacklist table **//
   if (Secure_Proxy.Exist_user_black_list (User_ID, hashed_Pwd)) then
       User_Status ← Denied_Access;
   else
       //** find the user identifier and their secret key are correct **//
       if (Secure_Proxy.Exist_user_authentication_list (User_ID, hashed_Pwd)) then
           User_Status ← Authorized_Access; //** Authorized the status of the user*//
           User_Trust_Level ← Green_Level; //** Give the user a  full authorization access **//
           Secure_VM_Allocation (Task_ID) //** Send request to hypervisor to create VM  for user **//
       else
           User_Status←UnAuthorized_Access;
           P_UnAuth ← P_UnAuth + 0.1; //** Increase the probability of unauthorized access**//
           if (P_UnAuth> Th_UnAuth) then
             //** Insert the user mobile device on the blacklist table**//
             Insert_VM_BlackList (new User(User_ID, hashed_Pwd));
             User_Status ← Denied_Access;
           endif
       endif
   endif
   return User_Status
End
```

It takes the user request and its device information as input. The signed password is sent over an authenticated session using the secure interface to a Proxy-3S for decryption. The algorithm applies Hash-Diffie Hellman encryption techniques and returns the authorization decision. This is done by following four major steps:

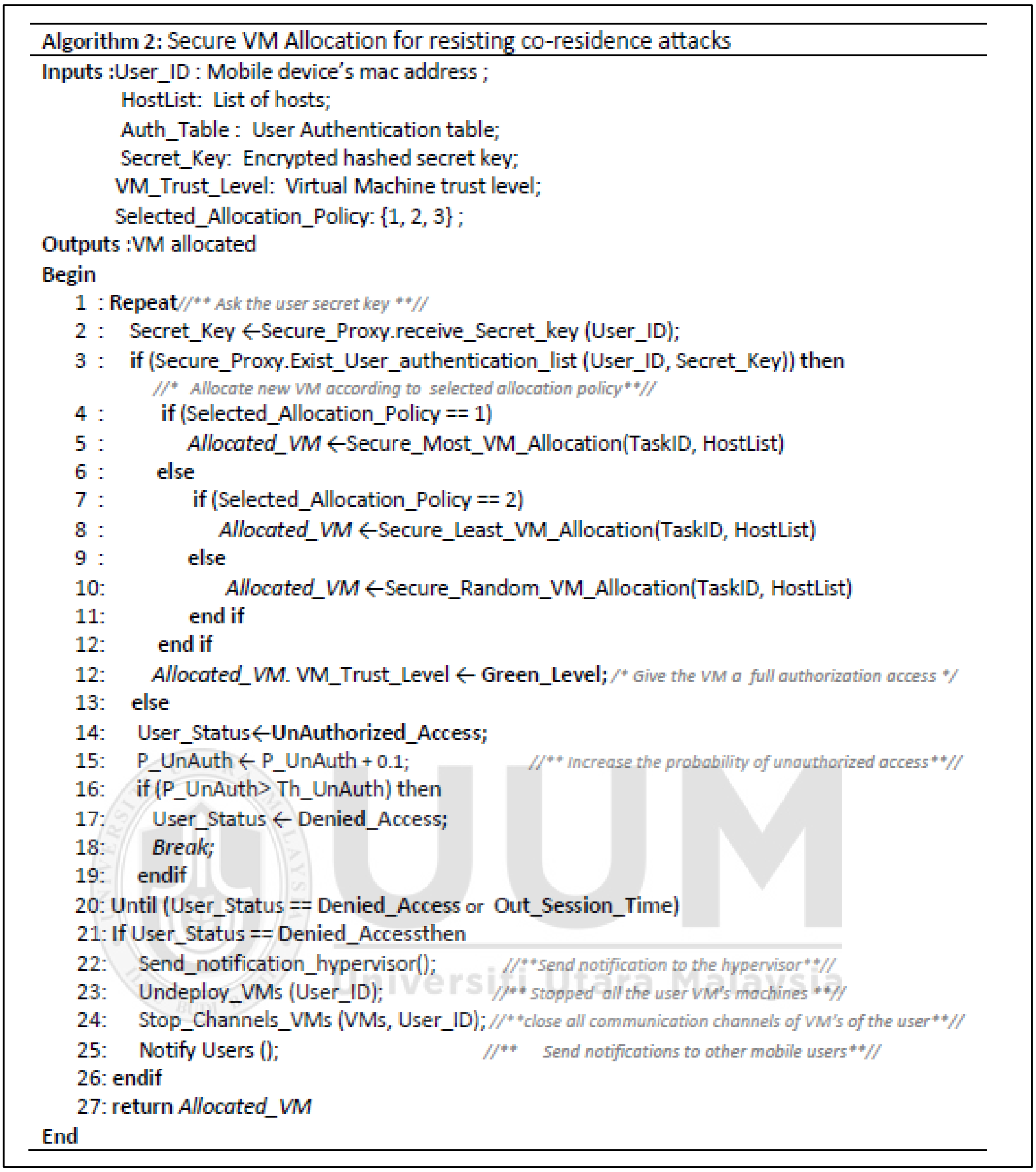
- Step 1 (line 1): The Proxy-3S receives the identifier (encrypted signature) of the mobile user. It decrypts the encrypted signature to extract out the hash codes.
- Step2 (lines 2 - 3): The Proxy-3S checks the blacklist table and denies access if the user's mobile device is on the list.
- Step3 (lines 4 - 17): It compares the extracted hash code (line 5) with the available authorized hash codes. The access control module uses a User Authentication table for all managed mobile users. Once the Proxy-3S finds the hash code, it gives the user a full authorization access level (i.e.: Green level), then it sends the requests to the hypervisor in order to create a VM for the user. After the VM is created on the cloud's host, the intensive task is allocated on the VM. If the Proxy-3S does not find any matched hash code, it increments the probability of unauthorized access. The increment of 0.1 is applied once any checking fail of hash codes occurs. Proxy-3S checks the user status if the current probability access is upper than the probability of unauthorized access threshold (e.g., 0.7) then inserts the user's mobile device on the blacklist table.

The User Authentication table contains hash encrypted codes. With the hash code, it is impossible to know the MAC address and password of the user. In this case, the malicious cloud provider cannot know the user's name and password, and if the database of the cloud is accessed, no one can know the right values. The hashing preserves the integrity and confidentiality.

4.5.2 Secure VM allocation Algorithm for resisting co-residence

The pseudo code of Secure VM Allocation policy for resisting co-residence is presented in Algorithm 2. It is used to allocate the VMs securely and isolate them from malicious VMs. The input consists of the authorization decision, which is the output of the mobile user's access algorithm and the intended task to be deployed. The algorithm ensures the security of VM-based task. The output of the proposed algorithm consists of a list of deployed VMs based on their final checking decisions. In general, Secure VM Allocation can be divided into the following steps:

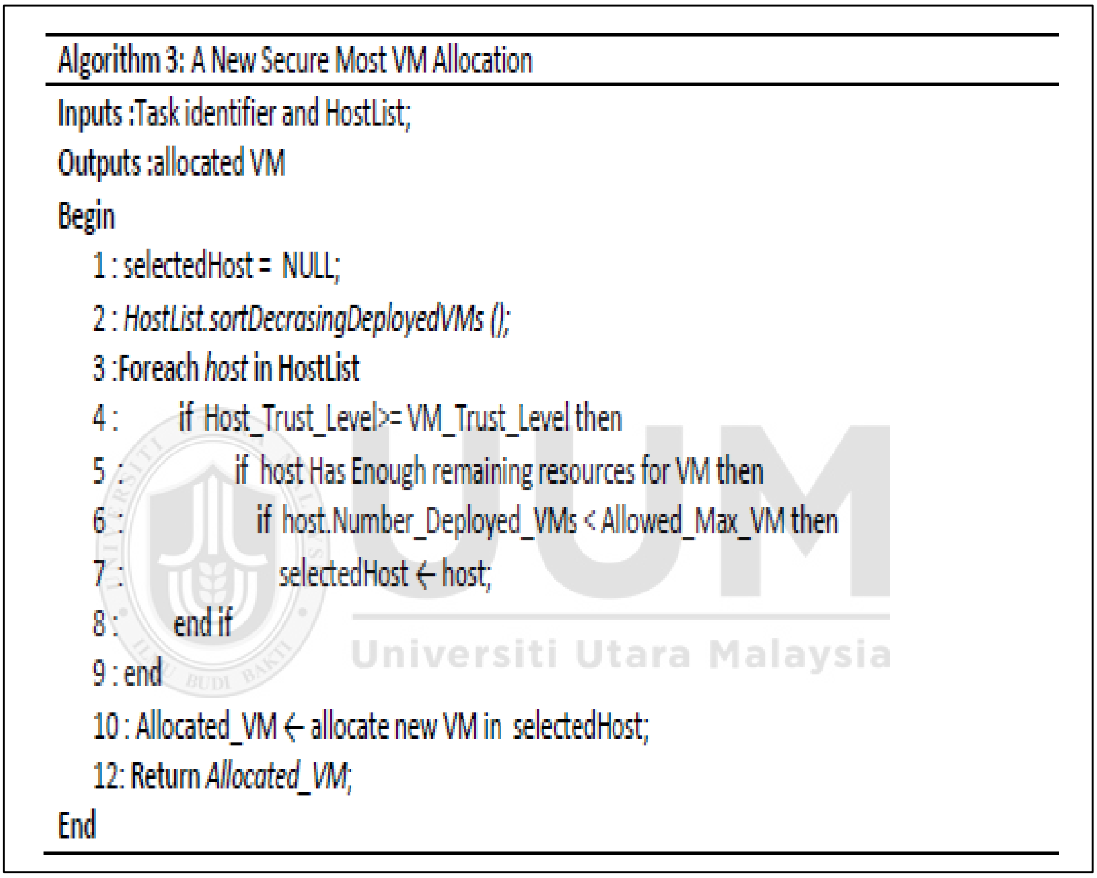
- Step 1 (lines 2): Proxy-3S is responsible to allocate the VMs of the user on the cloud servers. It uses three well-known VM allocation policies either: most, least, random [12]. These policies are enriched including Hash-Diffie Hellman encryption technique. Proxy-3S selects the best policy that provides the VM's data security:



```
Algorithm 2: Secure VM Allocation for resisting co-residence attacks
Inputs :User_ID : Mobile device's mac address ;
         HostList:  List of hosts;
         Auth_Table :  User Authentication table;
         Secret_Key:  Encrypted hashed secret key;
        VM_Trust_Level:  Virtual Machine trust level;
        Selected_Allocation_Policy: {1, 2, 3} ;
Outputs :VM allocated
Begin
  Repeat//** Ask the user secret key **//
    Secret_Key ←Secure_Proxy.receive_Secret_key (User_ID);
    if (Secure_Proxy.Exist_User_authentication_list (User_ID, Secret_Key)) then
      //*  Allocate new VM according to  selected allocation policy**//
      if (Selected_Allocation_Policy == 1)
         Allocated_VM ←Secure_Most_VM_Allocation(TaskID, HostList)
      else
          if (Selected_Allocation_Policy == 2)
             Allocated_VM ←Secure_Least_VM_Allocation(TaskID, HostList)
          else
             Allocated_VM ←Secure_Random_VM_Allocation(TaskID, HostList)
          end if
      end if
     Allocated_VM. VM_Trust_Level ← Green_Level;/* Give the VM a  full authorization access */
   else
    User_Status←UnAuthorized_Access;
    P_UnAuth ← P_UnAuth + 0.1;                //** Increase the probability of unauthorized access**//
    if (P_UnAuth> Th_UnAuth) then
      User_Status ← Denied_Access;
      Break;
    endif
  Until (User_Status == Denied_Access or  Out_Session_Time)
  If User_Status == Denied_Accessthen
    Send_notification_hypervisor();          //**Send notification to the hypervisor**//
    Undeploy_VMs (User_ID);                  //** Stopped  all the user VM's machines **//
    Stop_Channels_VMs (VMs, User_ID); //**close all communication channels of VM's of the user**//
    Notify Users ();                        //**    Send notifications to other mobile users**//
  endif
  return Allocated_VM
End
```

1. **Secure most allocation policy**: where this policy allocates the VMs on the cloud server that contains the greater number of VMs. We improve this policy by considering two parameters (as seen in Algorithm 3): the security (i.e. hash and encryption algorithms to secure high trusted host that deploy the new VM) and energy saving (i.e. CPU utilization of hosts, data traffics of communication channels). Once the new VM has been deployed, the sensitive data is sent to the VM to be handled.

2. **Secure least allocation policy**: where this policy allocates the VMs on the cloud server that contains the less number of VMs. We improve this policy by considering two parameters: the security (i.e. hash and encryption algorithms to secure high trusted host that deploy the new VM) and energy saving (i.e. CPU utilization of hosts, data traffics of communication channels).



Algorithm 3: A New Secure Most VM Allocation

```
Inputs :Task identifier and HostList;
Outputs :allocated VM
Begin
    selectedHost = NULL;
    HostList.sortDecrasingDeployedVMs ();
    Foreach host in HostList
        if Host_Trust_Level>= VM_Trust_Level then
            if host Has Enough remaining resources for VM then
                if host.Number_Deployed_VMs < Allowed_Max_VM then
                    selectedHost ← host;
        end if
    end
    Allocated_VM ← allocate new VM in selectedHost;
    Return Allocated_VM;
End
```

3. **Secure random allocation policy**: where this policy allocates the VMs on the cloud server randomly. We improve this policy by considering two parameters: the security (hash and encryption algorithms to secure high trusted host that deploy the new VM) and energy saving (.e. CPU utilization of hosts, data traffics of communication channels).

- Step2 (lines 3-27): Proxy-3S must check the user authentication table in order to allow or to stop the allocation of VM. It requests the user identifier and its encrypted secret key in order to check the authenticity of the corresponding user. The checking algorithm presents two situations:

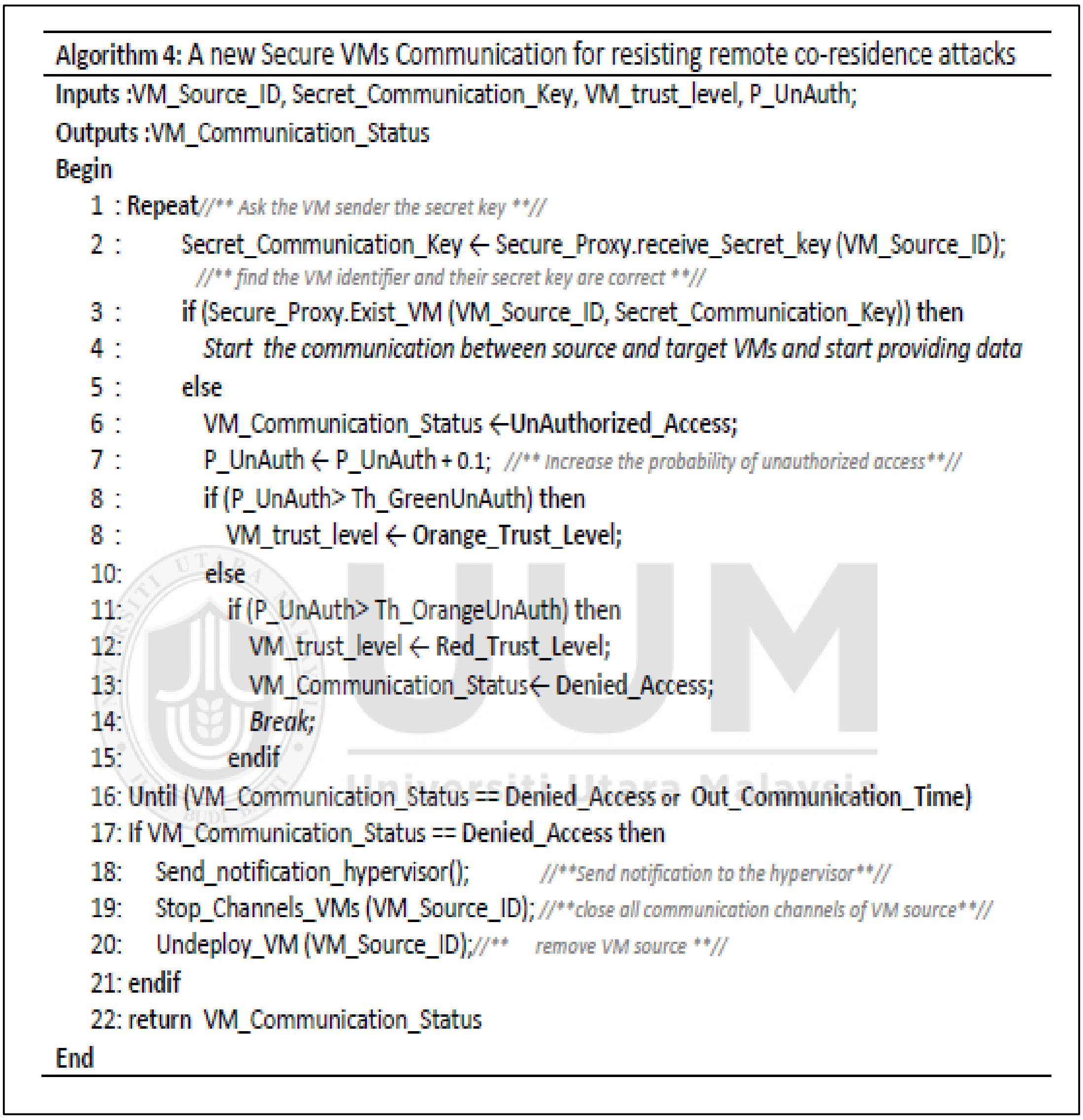
- If the Proxy-3S fails to authenticate user, it increments the user unauthorized probability. If the current unauthorized probability is upper than the probability of unauthorized allocation (e.g., 0.7), it sets the current user as Malicious user (and puts it in user blacklist table) and stops the VM allocation process on the host. It also broadcasts the identity of the malicious user to all other users.
- If the Proxy-3S recognizes the user is legal. Proxy-3S sends a request to the hypervisor in order to allocate the VM. It gives a session time to any VM hosted on the cloud. This session time is to control the VM and its behavior continuously. If a VM does not react_correctly in the server, Proxy-3S removes it from the host.

4.5.3 Secure VMs Communication Algorithm for resisting remote co-residence

The pseudo code of Secure VMs Communication policy for resisting co-residence is presented in Algorithm 4. Through algorithm 4, it is possible to authorize VMs to access to the exchanged sensitive data. The algorithm takes as input VMs List to collaborate. The communication decision returned by this algorithm encompasses the permission to access the exchanged data. This is done in following three major steps:

- Step1 (lines 2): When the VM requests another VM for exchanging data, the Proxy-3S checks the secure keys of VM that intend to communicate through its table called VM Communication Authentication table.

- Step2 (lines 2-4): If the Proxy-3S checks the correctness of exchanged keys, it asks the Hypervisor to allow communication among VMs to exchange sensitive data.



Algorithm 4: A new Secure VMs Communication for resisting remote co-residence attacks

```
Inputs :VM_Source_ID, Secret_Communication_Key, VM_trust_level, P_UnAuth;
Outputs :VM_Communication_Status
Begin
   1 : Repeat//** Ask the VM sender the secret key **//
   2 :       Secret_Communication_Key ← Secure_Proxy.receive_Secret_key (VM_Source_ID);
               //** find the VM identifier and their secret key are correct **//
   3 :       if (Secure_Proxy.Exist_VM (VM_Source_ID, Secret_Communication_Key)) then
   4 :         Start the communication between source and target VMs and start providing data
   5 :       else
   6 :         VM_Communication_Status ←UnAuthorized_Access;
   7 :         P_UnAuth ← P_UnAuth + 0.1;  //** Increase the probability of unauthorized access**//
   8 :         if (P_UnAuth> Th_GreenUnAuth) then
   8 :           VM_trust_level ← Orange_Trust_Level;
   10:         else
   11:           if (P_UnAuth> Th_OrangeUnAuth) then
   12:             VM_trust_level ← Red_Trust_Level;
   13:             VM_Communication_Status← Denied_Access;
   14:             Break;
   15:           endif
   16: Until (VM_Communication_Status == Denied_Access or  Out_Communication_Time)
   17: If VM_Communication_Status == Denied_Access then
   18:     Send_notification_hypervisor();       //**Send notification to the hypervisor**//
   19:     Stop_Channels_VMs (VM_Source_ID); //**close all communication channels of VM source**//
   20:     Undeploy_VM (VM_Source_ID);//**    remove VM source **//
   21: endif
   22: return  VM_Communication_Status
End
```

- Step 3 (lines 5-22): If the Proxy-3S fails to identify the VM sender, it increments the VM trust level. If the current trust level is upper than the probability of unauthorized communications (i.e., 0.7 thresholds), it sets the current communication as denied and removes the VM.

4.6 Conclusion

In this chapter, we have presented a new Three Policies secure cloud Proxy called Proxy-3S for preventing the access of malevolent users to cloud services as well as ensuring the data integrity and confidentiality of user’s sensitive data across collaborating VMs on the cloud environment.

To our best knowledge, none of the existing works takes into consideration the communication aspect between the VMs that performing distributed application’s tasks on various hosts. In addition, we control the sensitivity of data across different distributed VMs. This will improve the efficiency and coverage of attack. Moreover, the chapter presented Proxy-3S with its three security policies. The first policy is intended to manage the access rights of the user when deploying their VMs. The second policy is aimed to protect the hypervisor and preventing the break-down of isolated VMs. The third policy provided secure communication between VMs that transferring sensitive data between each other.

In the next chapter, we will present a new design of an extended secure simulation tool that gives opportunities to simulate the different researchers’ security approaches and policies.

CHAPTER FIVE

SecNetworkCloudSim: AN EXTENSIBLE SIMULATION TOOL FOR SECURE DISTRIBUTED MOBILE APPLICATIONS

5.1 Introduction

Chapter Four identified the security metrics that will be used to evaluate the performance of the secure approach in cloud computing. Further of presenting the security metrics, it presented three well-detailed algorithms that can be used to secure the VMs of users and quantify the proposed approach performance. In order to evaluate our security approach. A Java-based prototype is implemented on a powerful extensible simulator named: SecNetworkCloudSim which is an extended simulation tool of NetworkCloudSim in order to allow managing VMs security and data confidentiality automatically.

The organization of this chapter is as follows: Section 5.2 presents the extended secure layers in NetworkCloudSim simulator. Section 5.3 provides the design implementation of SecNetworkCloudSim which presents the main integrated classes in NetworkCloudSim. Further, the Simulation execution workflow of the SecNetworkCloudSim is presented in this Section. Finally, Section 5.4 concludes the chapter.

5.2 An Extended Security Simulation Tool

Due to the need of well understanding the cloud paradigm technology capabilities, a number of cloud simulators are available today such as CloudSim [124], Green Cloud [127], iCanCloud [128], GroudSim[129], NetworkCloudSim [130], secCloudSim

[131]. Such simulators are fruitful for cost analysis and advanced architecture as Cloud Computing. Whereas other simulation-based tools focus on power energy consumption, scheduling and allocation mechanisms, communication and networking between VMs.

Augmenting cloud computing simulators with mobile distributed tasks that handle sensitive data is considered an important challenge. Arguably, the greatest difficulty boils down to perform the different usage scenarios with different amounts of data over both unsecured VMs and malicious users in order to evaluate and analyze security algorithms. To overcome this challenge, few simulators [155] are offering cloud security challenges to control access data on top of the security techniques. However, this requires knowledge of security algorithms details which is not all users have and also requires spending significant amounts of time writing secure cloud infrastructure code. To clarify more, it is the task of a scientist to find solid and complete security mechanisms to encrypt/ decrypt and well protect sensitive data.

5.2.1 SecNetworkCloudSim: Secure Network Cloud Simulator

Key to the development of the simulator as a system for integrated security is the adoption of advanced researchers' policies and approaches that covers both the VM allocation and communications aspects while ensuring the protection of the user's private data among different virtual machines. In particular, the proposed simulator capable of managing the security of VMs, creating and allocating VMs based policies, deploying tasks and securing the data exchange between different VMs whether inside host or distributed hosts, as well as various network data centers. One of the main goals of this research is to extend the NetworkCloudSim architecture enables to model the

VM-based distributed mobile applications and guarantee secure access to the user's sensitive data.

After studying several cloud simulators tools, we choose to extend secure layers in NetworkCloudSim. NetworkCloudSim is considered one of the most powerful toolkit that gives ability to codify the behavior of very complex cloud system [130]. NetworkCloudSim does not provide classes to consider the security aspect. Since it does not integrates secure algorithms and policies on the cloud-computing environment to secure communications between distributed VMs. Otherwise, user's confidential data could be put at risk. Overcoming these limitations, we extend NetworkCloudSim simulator with new layer called Cloud Three Security Proxy (Proxy-3S) with three security policies in order to provide flexible simulation of the various security policies especially that those that focus on data client protection on the cloud side.

In this section, we present the NetworkCloudSim extension that ensures data security on the cloud. The extended NetworkCloudSim enables us to model VM-based distributed mobile applications and guarantee secure access to private data.

5.2.2 Main Functional building modules

Based on the principles specified in the architecture model of NetworkCloudSim, Figure 5.1 illustrates the main functional building modules of the SecNetworkCloudSim.

In particular, in the User code layer, regarding the application we integrate the following functional building modules:

1. **Distributed Application Configuration module:** in order to configure the application with multiple intensive tasks running on different cloud servers, the simulator allows the user to configure the application and defines the cloudlet's requirements in terms of needed resources like RAM, CPU (Pes cores), bandwidth and storage capacity. Moreover, the user can set the name, domain and environment platform of the application.

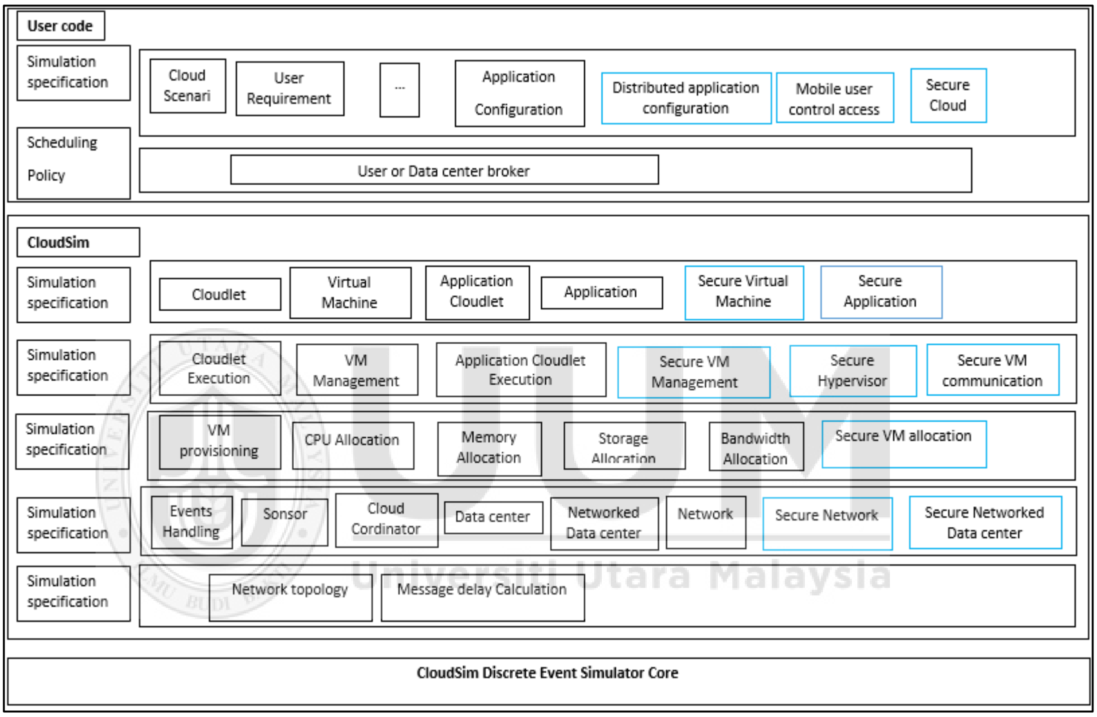


Figure 5.1. SecNetworkCloudSim architecture

2. **Mobile User Control Access (MUCA) policy module:** The simulator supports the integration of user's control access that receive, encrypt/decrypt and check security information of the mobile user. Typically, this module uses the hash SHA 256 and Diffie-Hellman algorithms. The user sends the encrypted password (signature) and the module checks the validity of the signature by decrypting the signature to hash codes and verifying it with available list of hash codes. Fig 5.2 shows the result of secure user authentication with the hash Diffie-Hellman schema. The access of mobile device

to the cloud service needs the MAC authentication process. MUCA receives the MAC address, uses the SHA-256 to encrypt and protect it from leakage in distributed environment (line 3). After that, the same algorithm hash both the ciphertext and the signature again (line 5). MUCA provides the signature to the mobile user as a secret password. The experimental results show that the proposed simulation tool's module (MUCA) identifies early signs of attacks.

```
Line 1    User Address MAC: 1234.5628.1234.5678
Line 2    Hash of the Mobile device MAC address:
Line 3    949f411e20378b55b2d0bb1d17bb32b3b932bdf059ca2d2126406831cca38c72
Line 4    Encryption step....
Line 5    Encryption signature.... [B@233c0b17
Line 6    Decryption step....
Line 7    Final result....
Line 8    949f411e20378b55b2d0bb1d17bb32b3b932bdf059ca2d2126406831cca38c72
```

Figure 5.2. Results of secure user authentication with hash Diffie-Hellman schema

In the middle layer of NetworkCloudSim, we integrate the following functional modules in order to protect the user information provided by mobile applications from unauthorized access and malicious users, as well securing the offloading process of applications' tasks using VM in the cloud and ensuring the distributed communications security between VMs allocated in different hosts:

1. **Secure Hypervisor:** This module provides secure allocation management system (e.g. creates, runs and destroys a virtual machine on the cloud environment) and enables the isolation and separation between different virtual machines. However, many attacks can break isolation and extract sensitive data from legitimate virtual

machines, so the secure hypervisor provides a robust encryption key management schemas that include AES 128 bit encryption schema. The latter also checks all the VMs that run on the cloud server in session time. If a virtual machine provides an uncorrected key so the module will directly tear out resources and destroy the VM.

2. **Secure Virtual Machine:** The simulator provides the full lifecycle of encryption keys to virtual machines and protecting them from attacks. The VM lifecycle includes the control of the VM behavior when it takes a long time activities than a threshold time interval. Therefore, the VM considers as an attacker.

3. **Secure Virtual Machine Allocation:** The Secure Virtual Machine Allocation module is the functional module that secures the VMs allocation based on three security policies: secure VMs most allocation, secure least VMs allocation and secure VMs random policy. These policies control VMs of the users, explicitly allocate the VMs on the safe cloud host where they deployed and isolated from VMs attacker.

4. **Secure Virtual Machines Communication:** The security of VMs communication is the core-building module that enables security insights based on early detection of VMs attacker using advanced communication policies. This latter uses the Hash-Diffie Hellman algorithm and a robust communication policy to secure information exchanges against malicious VMs and threats over unsecured channels. Further, the module detects the VM that pretends like legal VM but in the communication explore malicious behavior. The secure virtual machines communication interacts with a secure networked data center module.

5. **The Secure Networked Datacenter:** This Secure Networked Data Center provides a secure network between different datacenters that engage in the interaction of the user's confidential and private data.

6. **Secure VM management:** The module provides isolation between VMs in order to avoid the co-location with attacker VMs. This module gives a trust status level to highly manage the deployment and communication aspect by leaving the processing of VM or remove it from the cloud host.

7. **Secure application:** secure application consist of many tasks collaborating with each other in secure manner. The hash and Diffie-Hellman algorithms provide the cloudlets that communicate a secure way to send data between them without any interception from a third malicious party.

8. **Secure Network:** includes the cryptography methods either symmetric or asymmetric.

5.3 Design and implementation of SecNetworkCloudSim

In this section, we present the main classes of SecNetworkCloudSim, which are also composed of many functional classes to ensure the data security on the cloud and enable the modeling of distributed mobile applications, hosts (mobile devices or cloud), VMs, cloudlets, a secure proxy and VMs security policies.

5.3.1 Modeling of SecNetworkCloudSim

Figure 5.3 presents a generic overview of the class diagram of SecNetworkCloudSim architecture. In SecNetworkCloudSim, **class User** represents the mobile user that intends to leverage and access to cloud services. Thereby, the cloud services access is restrained only to authorized users that having main security requirements: identifier, password and mobile device identifier (i.e. @MAC). When a user sends its request (i.e. offloads the intensive task for process on virtual machines, stores private data and demands-resources allocation), he provides the address Mac of mobile address and his password. If the user authentication is verified by Secure Cloud proxy then the user's task is deployed, else a denied access is sent.

Mobile Device class represents the mobile devices (e.g. PC, smartphone, Tablet, etc.) which they have limited computing resources. It is uniquely identified by a fixed Mac address. A User might switch between different mobile devices.

Distributed Mobile Application class describes the resources-intensive application that needs to be allocated in different distributed virtual machines and servers in the datacenter. A distributed application consisting of a set of distributed intensive tasks deployed on different hosts in a target environment.

Task class represents the computational element of distributed mobile applications. We distinguish between light and intensive tasks. The light task is performed on the mobile devices themselves, whereas the intensive task is handled on the cloud computing resources and results back afterward. The intensive task can be classified in provided and required tasks. Provided task is defined as the capacity needed to the task consumer to handle the fundamental cloud computing resources.

Cloudlet class is the job submitted by the mobile user for processing a task on the cloud. The cloudlet in SecNetworkCloudSim is characterized by the job's length, time and cloudlet type. Each cloudlet has its own cloudlet ID and runs in a specific VM. A VM can host and run several cloudlets.

Secure Cloud Proxy represents the key class that manages both the user's data access and communication aspects of the VMs' security in terms of the algorithms that monitor the security policies to protect the sensitive data against unauthorized access. The unauthorized access probability threshold is used to identify the VMs whether the VM is legal or attacker. The attacker is denied access and removed from the host. Different security policies have been utilized in order to ensure the data protection and privacy of legal VM, as detailed later.

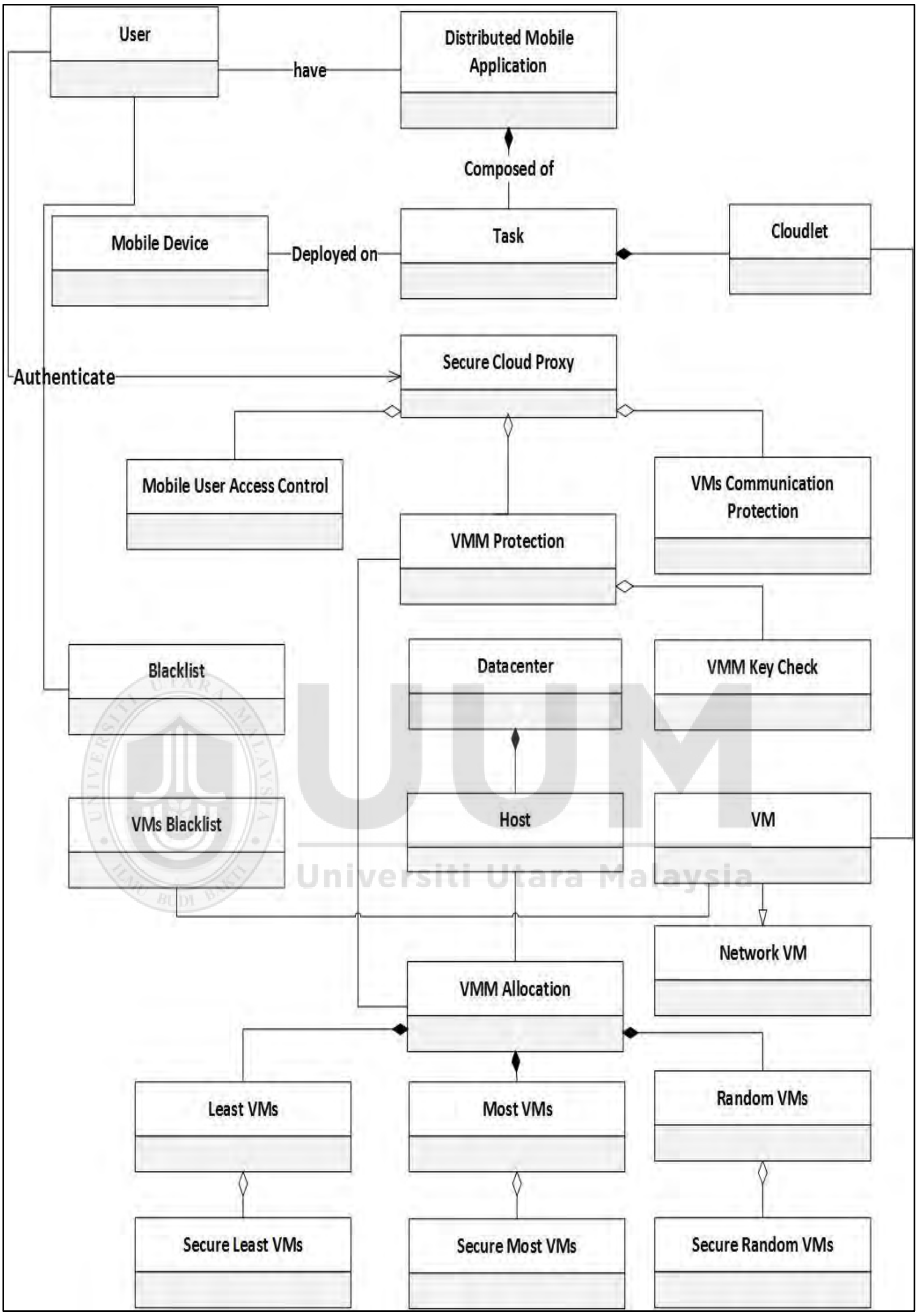


Figure 5.3. SecNetworkCloudSim class diagram

Mobile User Control Access: this class is useful for legitimate users to access the services provided by the cloud through their hashed and encrypted password with

Diffie-Hellman. This class maintains a table of *users' authentication* signatures that it checks for authentication. When a mobile user wants to access the cloud host and deploying its VMs, he sends an encrypted signature. The mobile user control access decrypts it and tries authenticating with all of the authentication signatures list.

VMM Protection: is used for managing the authorized VMs that hosted on the cloud's servers. A green status gives the VM high permission to interact with other VMs or access secret data. If a VM exceeds the probability of unauthorized threshold, a denied access is reported and a VM is added to VMs Black List.

VMs Communication Protection: is used to secure communication between two or more VMs intend to exchange sensitive data. Such class gets both VM's identifiers and the session number to check the validity of VMs for allowing them to communicate with each other. If a VM fails to offer the right secure session key, the trust level will decrease and the target VM is added to the VMs Blacklist list.

VMM Allocation Policy: after verifying the authentication of the user by the mobile user control access. The hypervisor allocates the available VM to the user's application tasks. The host to be allocated by the secure cloud proxy is the one that considers among trust hosts list. Further, the secure cloud proxy deallocates the VM when the process is complete. From such-class, we can obtain the host identifier that a given VM is executed. Also, the VMs belonging to a particular user.

Secure Least VM Policy: is used to avoid the co-location attack issue. The VM is allocated to the host that has least VMs processing within it. However, a selected host will not be selected again for new VMs allocation. If all hosts are running VMs, the host that has more free processing unit will be selected to deploy the new VMs. In addition, hashed and encrypted Diffie-Hellman key must be used for legitimate VMs to minimize unauthorized accessing on private data and managing the security allocation facilities.

Secure Most VM Policy: is used to achieve low power consumption. The VM is allocated to the host that has the most VMs processing within it in order to reduce host power consumption. For new VMs allocation, the selected host will be selected again to deploy them. Further, hashed and encrypted Diffie-Hellman key must be used for legitimate VMs to minimize unauthorized accessing on private data and managing the security allocation facilities.

Secure Random VM Policy: This class is used to randomly allocate the VMs to available hosts. However, a selected host will be either selected or not selected again for new VMs allocation. This random allocation strategy disturbs the attacker's goal strategy. Moreover, hashed and encrypted Diffie-Hellman key must be used for legitimate VMs to minimize unauthorized accessing on private data and managing the security allocation facilities.

VMs Blacklist contains the list of unauthorized VMs that fails to provide the correct secure key and their probability threshold exceeds a certain unauthorized access value.

Users Blacklist: This class contains the list of unauthorized mobile users.

VMM key check: This class contains the session identifiers of all VMs hosted on the Cloud. Each VM executes in a particular session time. For each session, the secure proxy-based cloud manages specific VMs and its secret keys.

Network Host: This class describes all servers located in datacenter in terms of hardware. It details the different information such as storage and memory size, the type of processing e.g. single or multi-core machine, etc. The different allocation policies for sharing the physical resources among VMs and distribution of user’s tasks.

5.3.2 Simulation Execution Workflow

The proposed SecNetworkCloudSim provides means to ease the evaluation of security policies to derive performance insights over exchanged sensitive data between distributed VMs. The steps to successfully protecting the applications' tasks on the cloud as well as their sensitive information through Proxy-3S are shown in Figure 5.4. After creating the network datacenter and configuring its multiple communication network switches across multiple hosts, the network datacenter broker is created in order to collect and submit VMs to the hosts. It also shows the number of cloud resources provisioned to a mobile application. The latter selects a suitable cloud host to meet the required application's quality services. In this stage, the Proxy-3S is created in order to protect the application tasks and their sensitive information while deployed on VMs as well as manage the whole security aspects whether on the cloud side or mobile side. The developer creates VMs on two sides: mobile device and cloud. The latter can change the specification of the simulated cloud and mobile VMs according to a specific user's requirements. The requirements depend on user specific-scenarios and configurations set. *I* refers to iteration and *Ts* is simulation time.

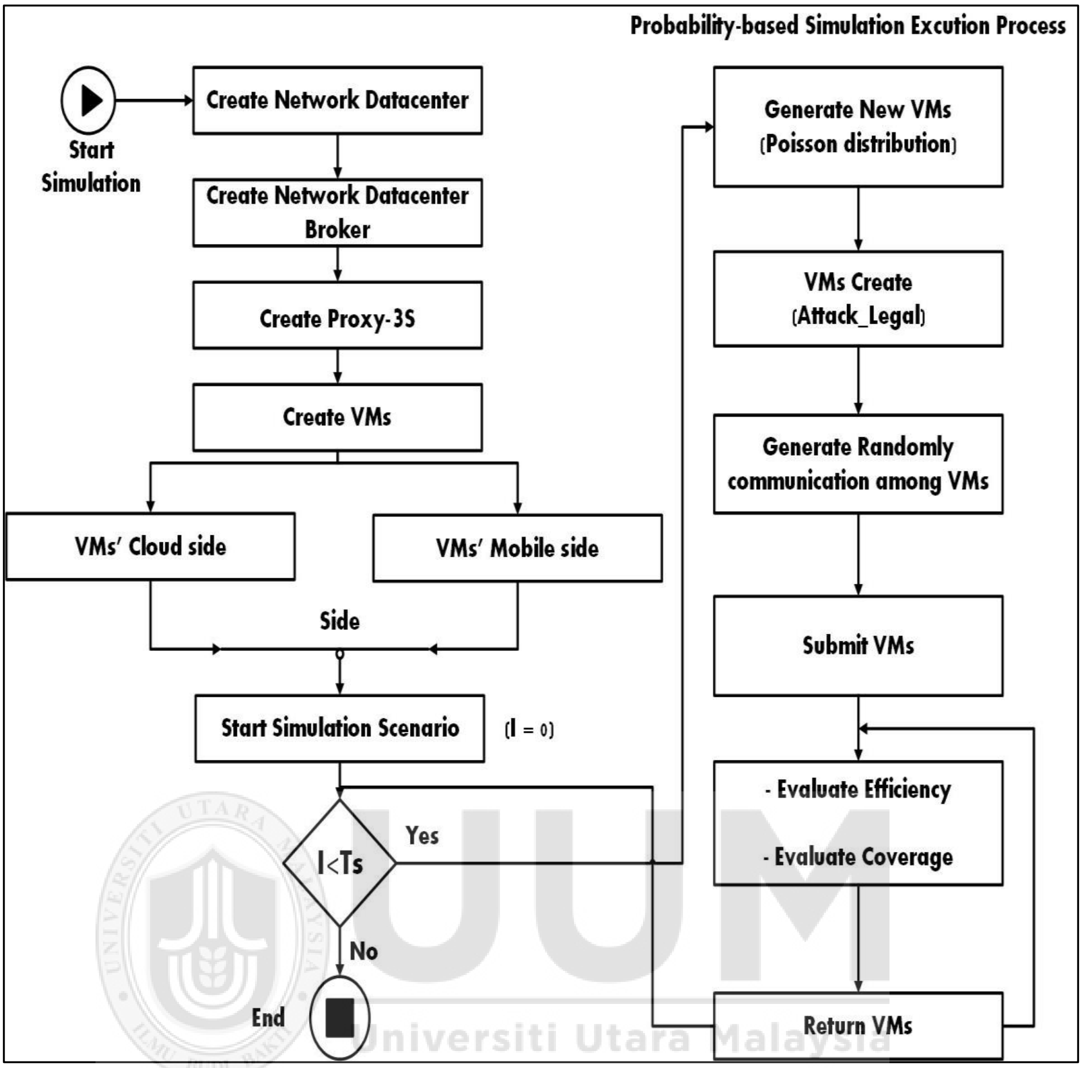

Probability-based Simulation Excution Process
Start Simulation
Create Network Datacenter
Create Network Datacenter Broker
Create Proxy-3S
Create VMs
VMs' Cloud side
VMs' Mobile side
Side
Start Simulation Scenario
(I = 0)
I<Ts
Yes
No
End
Generate New VMs (Poisson distribution)
VMs Create (Attack_Legal)
Generate Randomly communication among VMs
Submit VMs
- Evaluate Efficiency
- Evaluate Coverage
Return VMs

Figure 5.4. SecNetworkCloudSim simulation execution workflow

Once the simulation scenario starts to run on SecNetworkCloudSim simulator, the proxy-3S receives the new MAC address of mobile user's devices hashed, encrypted with Diffie-Hellman and inserted into cloud service accounts. The Proxy-3S sends the signature to the mobile user for access cloud service. During the simulation phase, we have used the Poisson event-based simulation model to generate VMs legal and attackers, communications channels among all VMs whether VMs legal with VMs legal, VMs attacker with VMs attacker and VMs Legal with VMs attacker. While finishing the simulation process, the simulator provides its report and evaluates security performances of the scenario such as efficiency and coverage of the application.

5.4 Summary

During this chapter, we presented SecNetworkCloudSim which is a NetworkCloudSim simulation tool extension. Firstly, this new extension allows the integration of security policies as sub-level in each task which permits the modeling of VMs based intensive distributed applications. Secondly, ensures the secure access of users' sensitive data shared between distributed VMs through the cloud. The chapter highlighted the main functional building modules of SecNetworkCloudSim and its software architecture. It provided design and implementation of the SecNetworkCloudSim through the class diagram and different involved classes to speed up the accurate simulation results across collaborating distributed tasks. The simulation execution workflow of SecNetworkCloudSim is already prepared to illustrate the main stages of security process once the distributed application executed on SecNetworkCloudSim.

In the next chapter, we will present some experimental results and comparisons with other existing approaches in the literature.

CHAPTER SIX

PROXY-3S PERFORMANCE ANALYSES AND EVALUATION IN DISTRIBUTED MOBILE ENVIRONMENT

6.1 Introduction

Our challenge is to provide a system that secures distributed mobile applications by integrating policies and monitoring the access to sensitive data shared between collaborative tasks using Proxy-3S.

In this chapter, we present the implementation and validation of the proposed approach's performance compared to related works. Section 6.2 gives the implementation and evaluation of Proxy-3S using real-world healthcare distributed intensive applications. Section 6.3 provides the different experimental configurations used to evaluate the efficiency and coverage of Proxy-3S. Section 6.4 examines the response time and security checking time-of the proposed system. Finally, Section 6.5 summaries the chapter.

6.2 Implementation and Validation

This section presents the validation of the proposed security approach in the context of distributed intensive tasks related to mobile applications, particularly in a health-care. We have extended a popular cloud simulator: NetworkCloudSim [130] with new concepts that provide more protection of sensitive data shared among distributed VM-based intensive tasks against different common attacks. The protection is done via enhanced security VM-based policies to achieve a high-security level on both cloud and mobile devices. The experiments were conducted by constructing healthcare distributed applications to demonstrate the efficiency and the effectiveness of the presented approach.

6.2.1 Prototype implementation

SecNetworkCloudSim is implemented in Eclipse Java Development Tool (JDT) that extends NetworkCloudSim. NetworkCloudSim is a cloud simulator that simulates advanced application models such as message passing applications and network model for datacenters. The current available cloud simulators (e.g., NetworkCloudSim) have no supports both VM's security and mobility features. These limitations may provide inaccurate results for sensitive distributed applications communicating tasks. Thus, we have extended NetworkCloudSim with Proxy-3S and its security policies, which allows the mobile users to grant access to the third-party intensive tasks of their applications in the cloud. It integrates Proxy-3S as a third party between the mobile users and their shared VM-based intensive tasks on the cloud. Its one of the main purposes is to ensure automatic application-user authentication for the deployed intensive tasks and to control the unauthorized access of users. Proxy-3S is located on the cloud side to manage the security and privacy of sensitive data and to enable secure communication among VM-based tasks applications. The extended NetworkCloudSim comprises up to three main modules, namely the Mobile client, the Proxy-3S and the cloud. The cloud is considered as a virtual network associated with the number of hosts that are located in datacenters. To accomplish such a system, three policies were developed within the Proxy-3S. The first one is the Mobile User Control Policy, which is intended for mobile users. The second one is the hypervisor Policy for managing the secure allocation of VMs that embedded intensive tasks. The last one is the VMs communication protection policy for ensuring the robust data exchanges among VMs while getting the authorization. In the mobile user control policy, a user provides the identifier and device MAC address. The following features may be performed on the deployed application:

- Offering a high-security level on both cloud and remote sites tasks.
- Providing an easy way to secure VM-based distributed applications in one-use-password facilities.
- Ensuring secure and legal private data exchanges between collaborative VMs.
- Providing robust VM-based oriented tasks processes for multi-users.

6.2.2 An Illustrative Case Study: Health Care System

This section illustrates the increase of efficiency and coverage of the attackers through health-care case study while their thin VMs intend to retrieve the sensitive medical data from legal user VMs. The case study describes a distributed health mobile application that follow-up diabetes disease for patients. The security management of such applications and sensitive communications depends on distributed tasks in remote sites either in the mobile devices or in the cloud platforms. It defines three fundamental actors: users/patient, physicians: professional staff, and doctors. They are allowed to exchange and share sensitive medical data on the cloud. The application is built with two android-GUI presentation tasks. The first one is intended for patients and the second one for physicians. The application allows users/patients to send their heart information, which will be posted as message encrypted and signed using Hash-Diffie Hellman. Patients can include weight, temperature and blood pressure. It allows a doctor and a cardiologist in other sites to follow the updates of a user/patient. In order to provide such features, many intensive tasks are offloaded on VMs and always running on the cloud, e.g. temperature analysis task, blood analyses applied on sensitive and private patient data. Figure 6.1 shows the application with several android-GUI tasks deployed on three mobile devices and six intensive-computation tasks intensive because the mobile device cannot support their execution due to its limited resources. In four different VMs on a different cloud server (e.g., Blood analyses task, temperature analysis task, weight analysis task and Diabetes Test task) depicted respectively as host#1, host#2 in Figure 6.1.

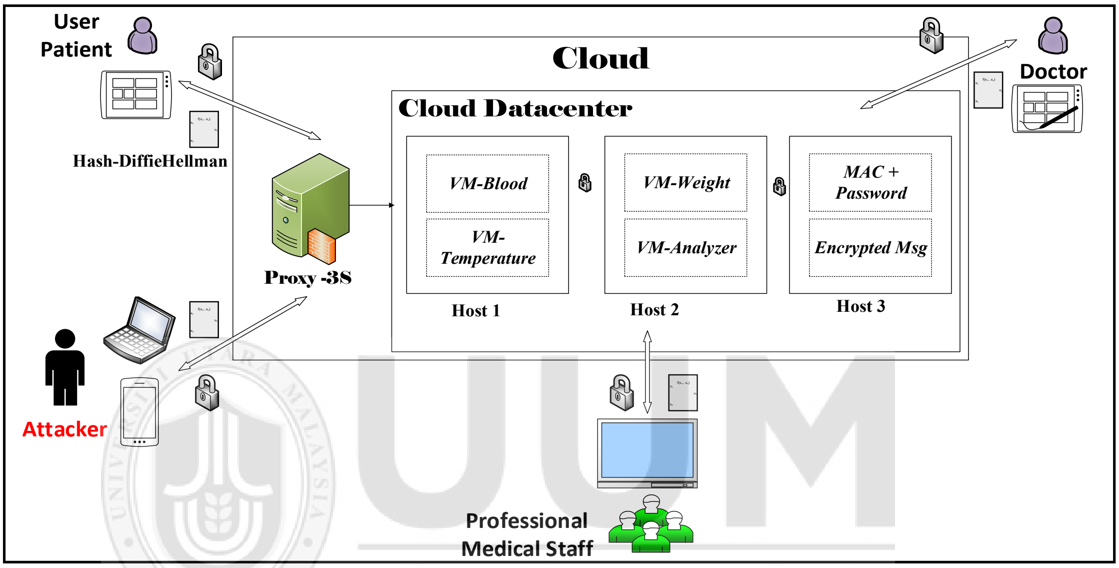


Figure 6.1. The architecture of distributed healthcare mobile application using Proxy-3S

Only the authorized doctors/cardiologists must access the patient’s medical record. However, the access could not be reliable, hence we use Proxy-3S deployed in the cloud to control the unauthorized access of users and to control user’s VMs against malicious VMs. In this case, we seek to improve the security of the distributed mobile application and to preserve the confidentiality of sensitive information outsourced in the cloud.

We handled the case where a cardiologist successfully accessed the sensitive information of patients and there are no malicious attacks. The steps to successful outsourcing sensitive information as well as the communication process tasks through

Proxy-3S are shown in Figure 6.2. The steps followed in VM communication protection policy are almost the same used in the request of sensitive information (see Figure 6.3). It starts by checking the target VM which could be the valid VM's identifier and password. Next, if the VM's authentication is verified then the VM data access allows else the data access is denied.

The Hash-Diffie Hellman secret key is used to encrypt/decrypt the message, it must be transparent to users. Next, the VMs communication protection policy monitors the security violations described in the previous section. So, it starts by checking off the accessed VM (e.g., diabetic test task). If the protection policy does not find any communication access violation, it authorizes the VM to share their sensitive data. The process ends when the system delivers the final results to the doctor.

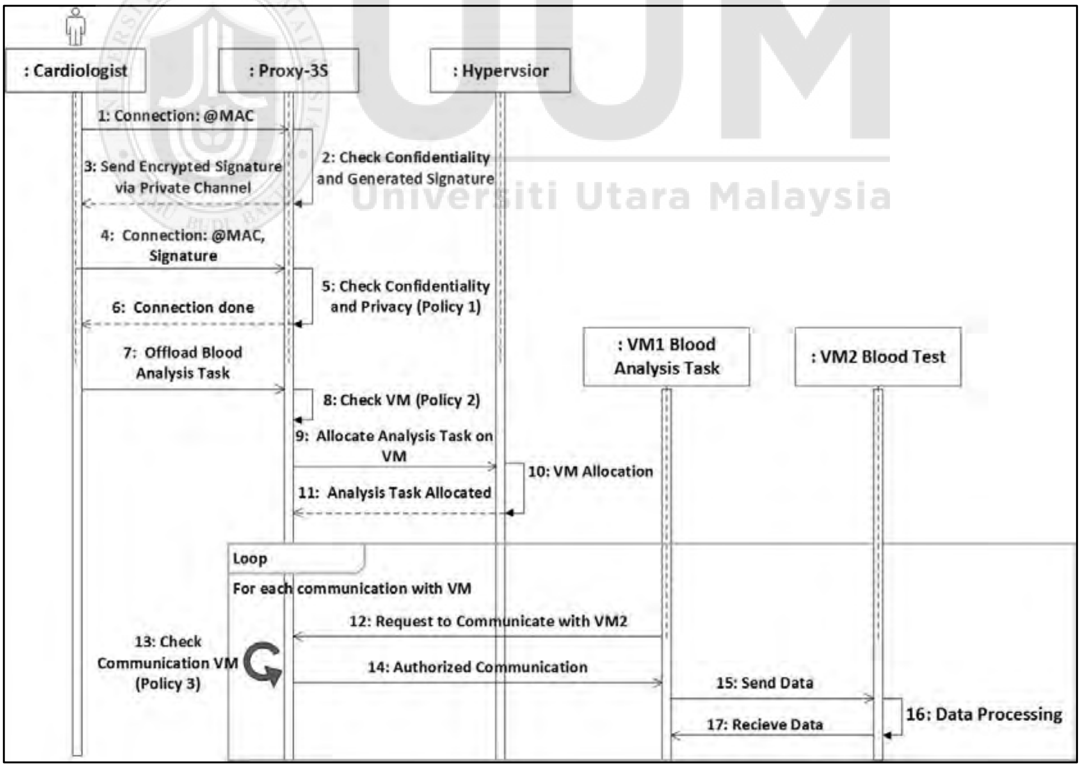

: Cardiologist
: Proxy-3S
: Hypervsior
1: Connection: @MAC
2: Check Confidentiality and Generated Signature
3: Send Encrypted Signature via Private Channel
4: Connection: @MAC, Signature
5: Check Confidentiality and Privacy (Policy 1)
6: Connection done
7: Offload Blood Analysis Task
8: Check VM (Policy 2)
9: Allocate Analysis Task on VM
10: VM Allocation
11: Analysis Task Allocated
: VM1 Blood Analysis Task
: VM2 Blood Test
Loop
For each communication with VM
12: Request to Communicate with VM2
13: Check Communication VM (Policy 3)
14: Authorized Communication
15: Send Data
16: Data Processing
17: Recieve Data

Figure 6.2. Actions performed and communication flow among actors and tasks

The professional medical staff has many VMs communicates with VMs of a doctor in order to exchange patient's data between each other. We suppose that VM attacker inside cloud server 1 co-located with VMs of professional medical staff. This later intends to communicate with VM-legal of a doctor called Analyzer task. The successful communication of the VM attacker is considered as remote co-location with VM legal. This means, despite the VMs are located in the different hosts, but the success communication deems the VM attacker is beside the VM legal which make the possibility of retrieving its data. Proxy-3S increases the interaction of efficiency and the coverage attacker which consequently increases the efficiency and the coverage of the attacker (see Figure 6.3).

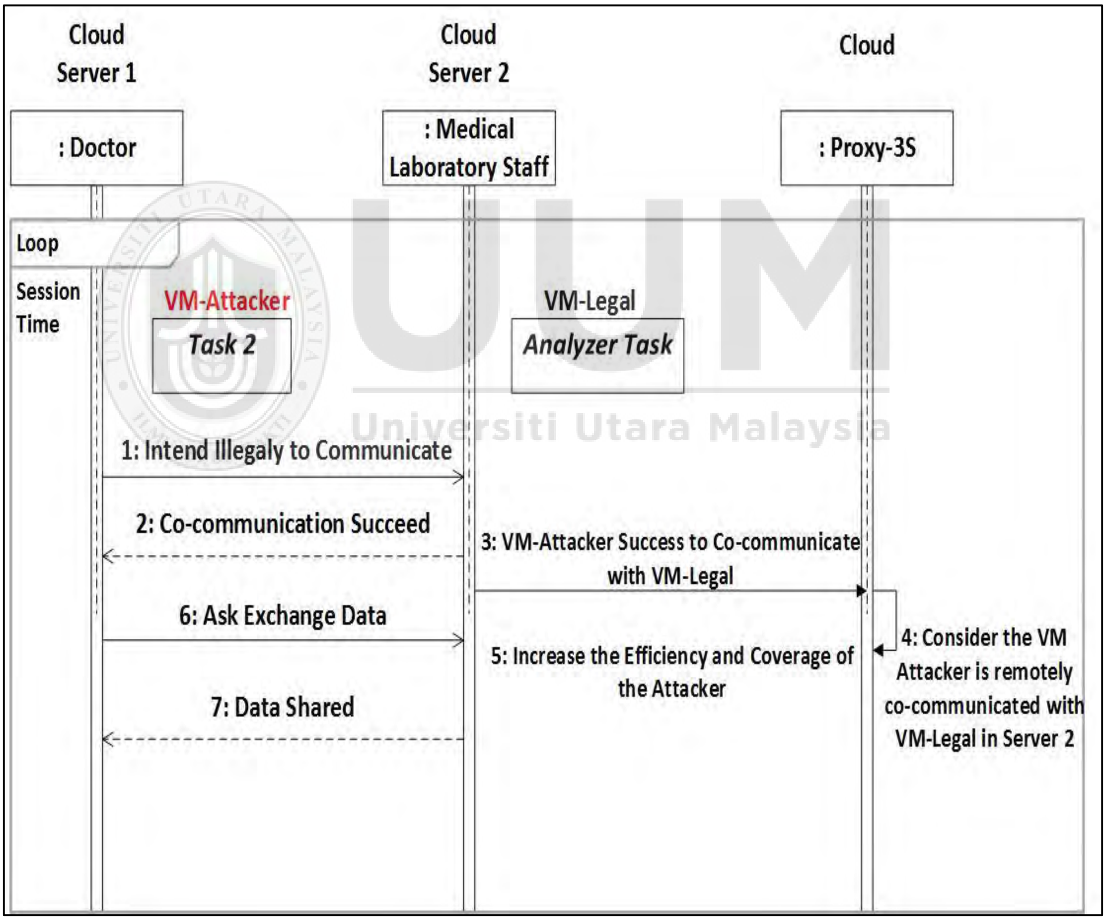

Cloud Server 1
Cloud Server 2
Cloud
: Doctor
: Medical Laboratory Staff
: Proxy-3S
Loop
Session Time
VM-Attacker
Task 2
VM-Legal
Analyzer Task
1: Intend Illegaly to Communicate
2: Co-communication Succeed
3: VM-Attacker Success to Co-communicate with VM-Legal
4: Consider the VM Attacker is remotely co-communicated with VM-Legal in Server 2
5: Increase the Efficiency and Coverage of the Attacker
6: Ask Exchange Data
7: Data Shared

Figure 6.3 Increase of efficiency and coverage communication of the attacker

6.3 Experimental evaluation

In order to verify and evaluate the above Proxy-3S approach and its policies, various experimental evaluations were conducted in order to evaluate the ability of Proxy-3S to protect the user's sensitive data while handled in thin VMs from the attacker threats.

6.3.1 Experimental configurations

All our experiments have been performed on an Intel Core i3-5005U CPU 2.00 GHz, 4 GB RAM, Laptop using Windows 8 (64 bit) system and Eclipse, an integrated development environment (Eclipse IDE). The proposed secure proxy starts running as a third party between the cloud and mobiles devices. Proxy-3S starts to encrypt all the mobile users address MAC and save the encrypted passwords in their own table. It checks the mobile users' passwords and detects a number of attacks. Proxy-3S has a list of legal users and attacker users. For evaluation, we use five configurations: the first configuration consists of one datacenter with 150 servers. Each server has 2-Xen VMs with a total of 300VMs, 8 cores, and 2 GB RAM and 1T storage capacity. This configuration also consists of one edge switch 150 ports connected to the cloud hosts. The second configuration (configuration B) consists of one datacenter with 300 servers, 1 Edge Switch, and 600 virtual machines. The third configuration (configuration C) consists of one datacenter with 450 serves, 1 Edge Switch, and 900 VMs. The fourth configuration consists of one datacenter with 600 serves, 1 Edge Switch, and 1200 VMs. Last configuration which considers a large scale of 1500 VMs, 750 servers in one datacenter and 1 Edge Switch.

All servers are connected with a bandwidth of 10 G bit/s. Each task of mobile healthcare application is executed on individual VMs for all configurations. Table 6.1 shows the details of each system configuration.

Table 6.1
Experimental configurations

Configurations	**Servers**	**Edge**	**VMs**	**CPU cores (Pes)**	**Storage capacity (GB)**	**Bandwidth(GB)**	**RAM (MB)**
Configuration 1	150	1	300	8	1000000	10000	2048
Configuration 2	300	1	600	8	1000000	10000	2048
Configuration 3	450	1	900	8	1000000	10000	2048
Configuration 4	600	1	1200	8	1000000	10000	2048
Configuration 5	750	1	1500	8	1000000	10000	2048

In the next sections, to verify the security and the execution time of proposed Proxy-3S approach and Secure VMs allocation policy, we deployed a distributed mobile healthcare application in a simulated environment as well as we evaluated and compared Proxy-3S with secure VMs allocation policy [12, 83]. For an attacker, the goal is to maximize either their efficiency or coverage of their illegal VMs.

6.3.2 Evaluating the efficiency of Proxy-3S

We have evaluated and compared the efficiency of Proxy-3S with/without secure communication policy in five configurations as detailed in Table 6.1. The goal is to minimize both expired events ratio and missed situation ratio and to minimize the execution time. The high efficiency ratio shows the poor security of the approach. Table 6.2 shows the efficiency results and the execution time of Proxy-3S with/without secure communication performed on five cloud configurations. Proxy-3S with/without secure communication have different evaluation strategies concerning VM’s policies and VM’s attacks identification. The Proxy-3S approach without secure communication controls only the user access policy and allocates VMs in a same host to check all co-located attacks. In contrast, the Proxy-3S with secure communication consider VMs exchanges as much as the number of user's communication tasks that run simultaneously at the same time in different hosts. The more malicious VMs and remote co-located identifications attacks can be performed simultaneously.

It is obvious that Proxy-3S with secure communication yields better efficiency on all configurations involved in the simulation. It is slightly higher than the efficiency results of Proxy-3S with secure communication. This caused by the absence of VM secure communication policy between communicating VMs deployed on different hosts. Thus, a system cannot detect malicious communications between VMs attacker and legal. However, the execution time of Proxy-3S without secure communication is slightly lower than the execution time of Proxy-3S with secure communication. The reason is that the VM security checking requires all the communications types and checking the security keys of all VMs deployed on different hosts in the cloud. Figure 6.4 shows the evaluation of Proxy 3S efficiency with/without secure communication policy under different configuration.

Table 6.2

Evaluating the efficiency and execution time on different configurations using Proxy-3S

Configurations	**Proxy-3S without secure communication**		**Proxy-3S with secure communication**	
	Efficiency	**Execution Time (ms)**	**Efficiency**	**Execution Time(ms)**
Configuration 1	0.4729	41048	0.3560	44142
Configuration 2	0.4966	77728	0.3930	78353
Configuration 3	0.4800	113239	0.3754	118272
Configuration 4	0.4648	364484	0.3650	367999
Configuration 5	0.4624	463530	0.3698	467572

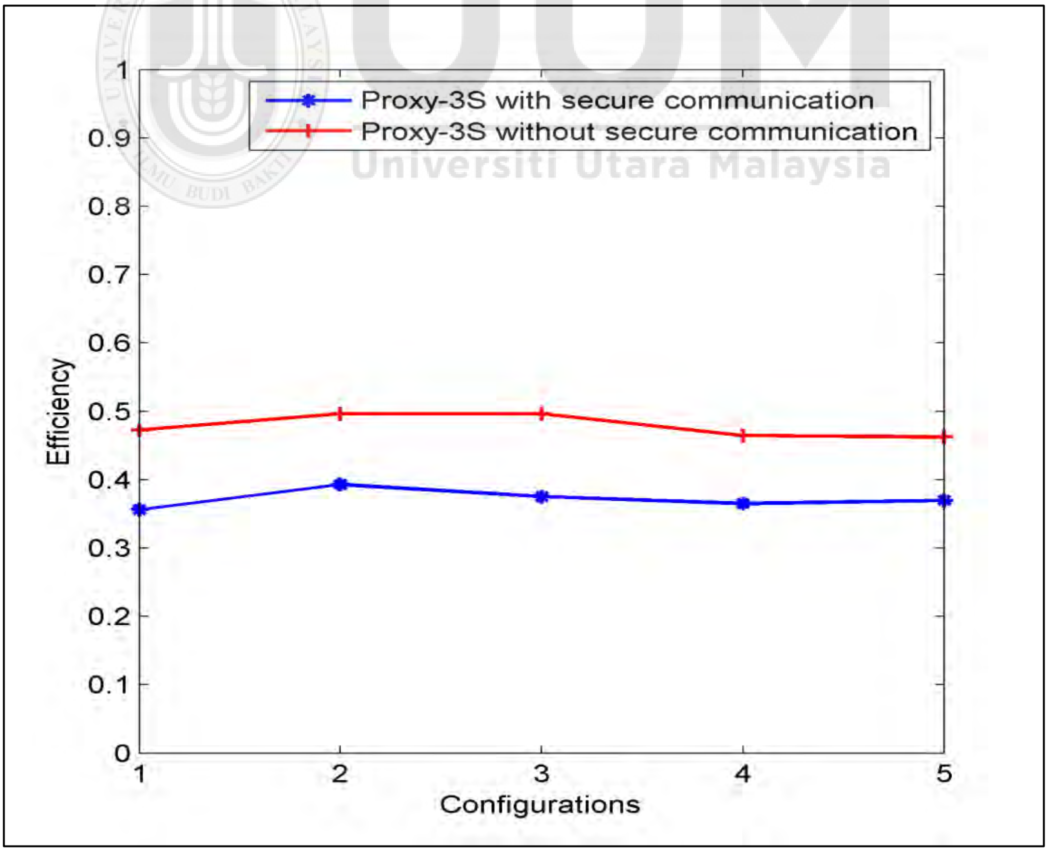


Figure 6.4. Evaluation of efficiency on different configuration using Proxy-3S (with/without) secure communication

6.3.3 Evaluating the coverage of Proxy-3S

We have evaluated and compared the performance of Proxy-3S with/without secure communication policy in terms of the coverage and execution time for five given configurations. The detail configuration parameters can be found in Table 6.1. Table 6.3 shows the coverage and execution time using Proxy-3S of five configurations. Proxy-3S with secure communication archives the lowest values of coverage in all evaluated configurations. In all configurations, the Proxy-3S with secure communication can complete the security task in about 4.4 - 11.8 ms. Figure 6.5 shows the evaluation of Proxy-3S coverage with and without secure communication policy under different configuration

Table 6.3
Evaluating the coverage and execution time on different configurations using Proxy-3S

Configurations	**Proxy-3S without secure communication**		**Proxy-3S with secure communication**	
	Coverage	**Execution Time (ms)**	**Coverage**	**Execution Time(ms)**
Configuration 1	0.4605	41048	0.3798	44142
Configuration 2	0.4900	77728	0.4188	78353
Configuration 3	0.4800	113239	0.4147	118272
Configuration 4	0.4617	364484	0.4000	367999
Configuration 5	0.4736	463530	0.4119	467572

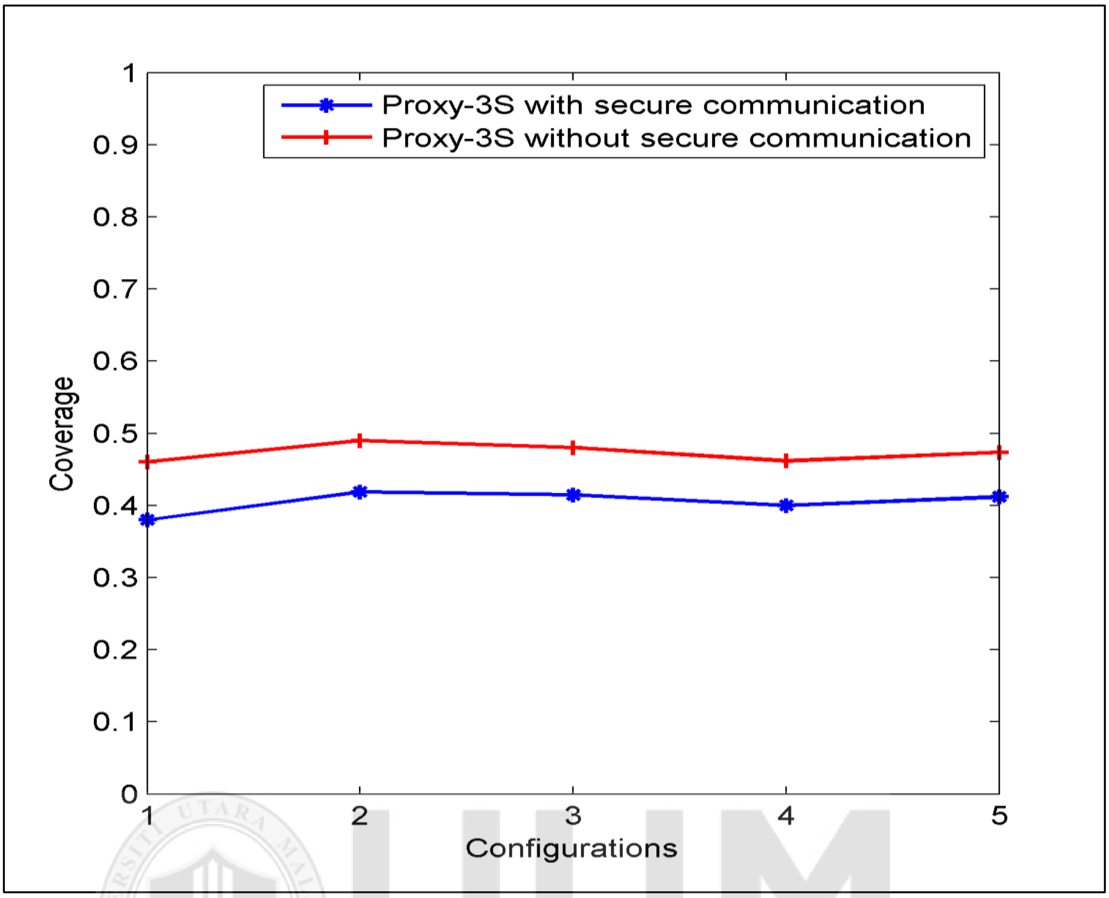


Figure 6.5. Evaluation of Coverage on different configuration using Proxy-3S (with/without) secure communication.

6.3.4 Efficiency and coverage comparison

In order to validate the efficiency and coverage of the proposed approach, we compare the obtained results using Proxy-3S with similar works of [12, 83]. Table 6.4 shows the improved performance of efficiency of Proxy-3S under five configurations and various number of VMs with the approach described in [12, 83] and our proposed approach. Proxy-3S improves and provides lower efficiency compared to related works. The reasons are explicit consideration of the secure communications between VMs deployed on different servers in the proposed approach while the related work considers only the efficiency inside the same server cloud. We also consider in the proposed approach an attacker VM that may be act legal want to leverage the cloud service. They can access to shared sensitive data shared among distributed VMs in as legal VMs.

We assume that attackers can acts as legal users when they the cloud and allocates their VMs on the cloud's servers. The VMs attackers co-locate with legal VMs in order to obtain sensitive information from them. When the attacker requests communication with other VMs located on the same or another server. Proxy-3S would detect a malicious behavior activity from the fake legal VM if he failed to give the right secret key. In this case, the legal VM is identified as an attacker, where the coverage and efficiency values will be different without the aspect of communications between VMs. Figure 6.6, Figure 6.7, Figure 6.8, Figure 6.9 and Figure 6.10 show the efficiency comparison between our work and related works [12, 83] respectively under various configurations with a different number of VMs spreading (300, 600, 900,1200,1500).

Table 6.4
Efficiency comparisons on different configurations and VMs

Number of VMs	Configuration 1	Configuration 2	Configuration 3
Secure Proxy-3S			
50 - 350 – 650	0.3076	0.3627	0.3466
100 - 400 – 700	0.2857	0.3621	0.3522
150 - 450 – 750	0.3372	0.3523	0.3407
200 - 500 – 800	0.3898	0.3624	0.3520
250 - 550 – 850	0.3698	0.3646	0.3460
300 - 600 – 900	0.3586	0.3573	0.3560
Secure VM allocation [12, 83]			
50 - 350 – 650	0.3333	0.4316	0.4189
100 - 400 – 700	0.3658	0.4299	0.4181
150 - 450 – 750	0.4166	0.4225	0.4111
200 - 500 – 800	0.4653	0.4242	0.4212
250 - 550 – 850	0.4218	0.4421	0.4223
300 - 600 – 900	0.4358	0.4294	0.4285

Number of VMs	Configuration 4	Configuration 5
Secure Proxy-3S		
950 - 1250	0.3618	0.3843
1000 - 1300	0.3641	0.3753
1050 - 1350	0.3647	0.3715
1100 - 1400	0.3707	0.3714
1150 - 1450	0.3704	0.3667
1200 - 1500	0.4535	0.3600
Secure VM allocation [12, 83]		
950 - 1250	0.4324	0.4628
1000 - 1300	0.4354	0.4534
1050 - 1350	0.4376	0.4484
1100 - 1400	0.4431	0.4474
1150 - 1450	0.4439	0.4424
1200 - 1500	0.4528	0.4401

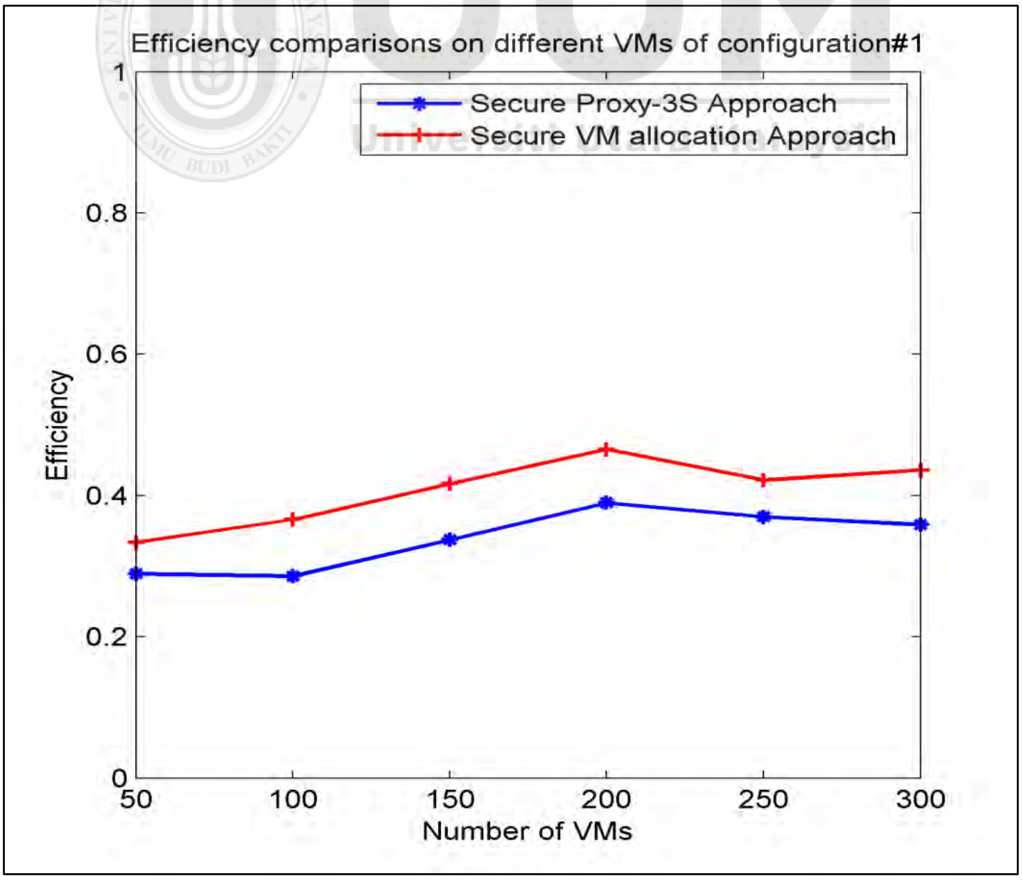


Figure 6.6. Efficiency comparison in configuration 1 (300 VMs spread).

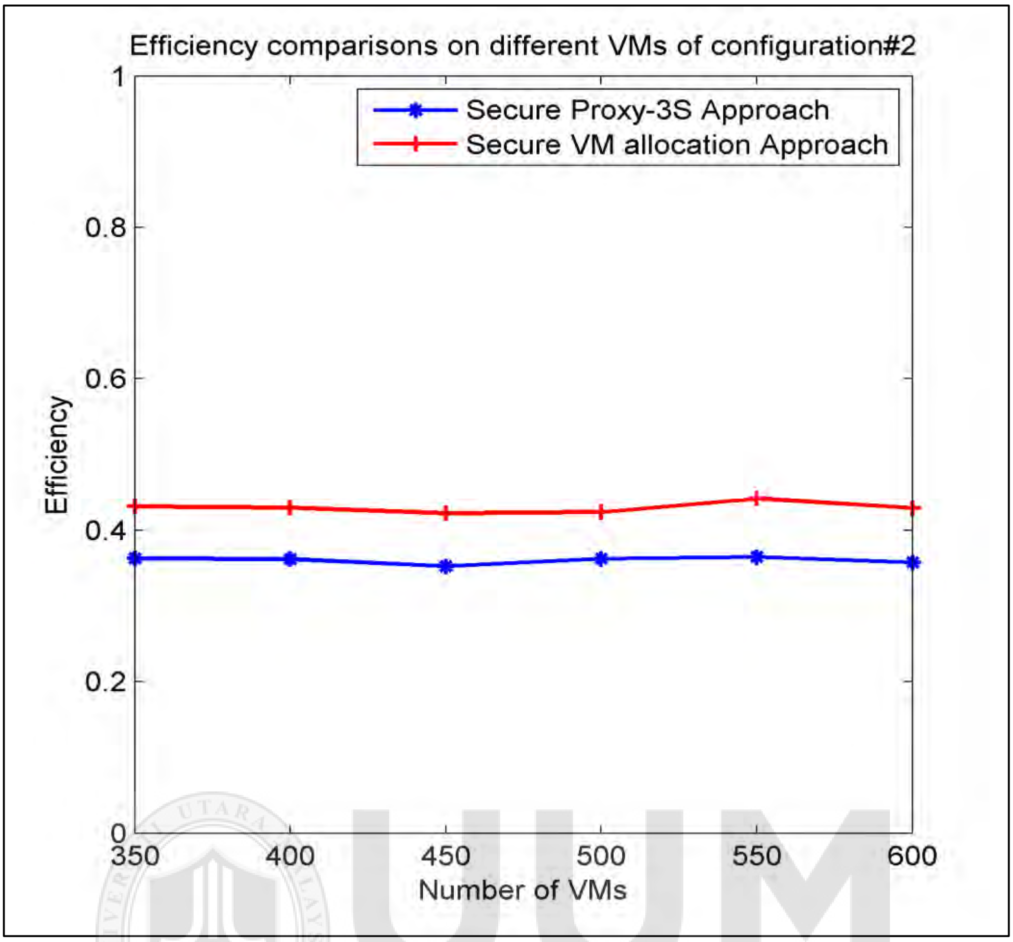


Figure 6.7. Efficiency comparison in configuration 2 (600 VMs spread).

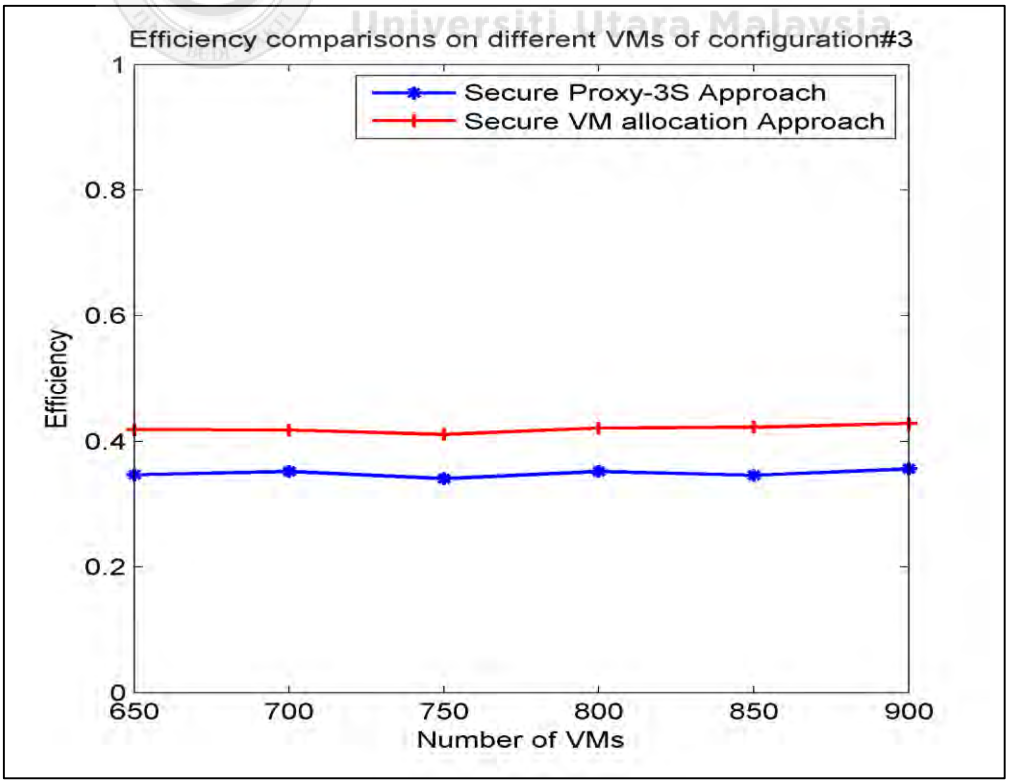


Figure 6.8. Efficiency comparison in configuration 3 (900 VMs spread).

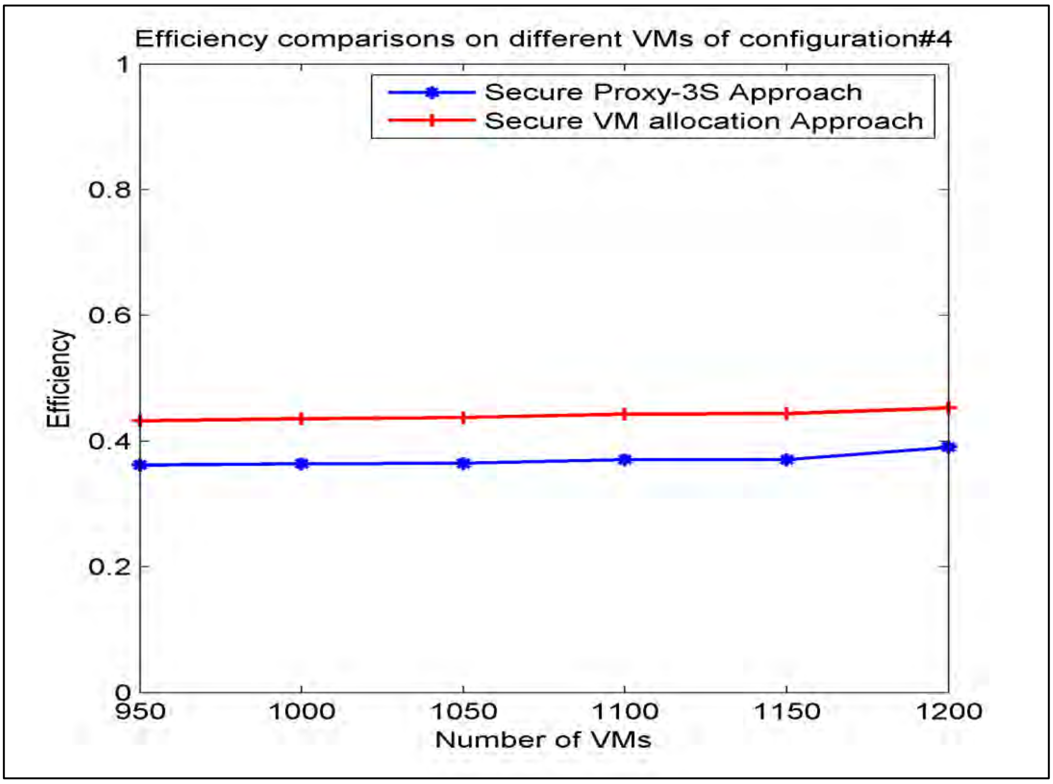


Figure 6.9. Efficiency comparison in configuration 4 (1200 VMs spread).

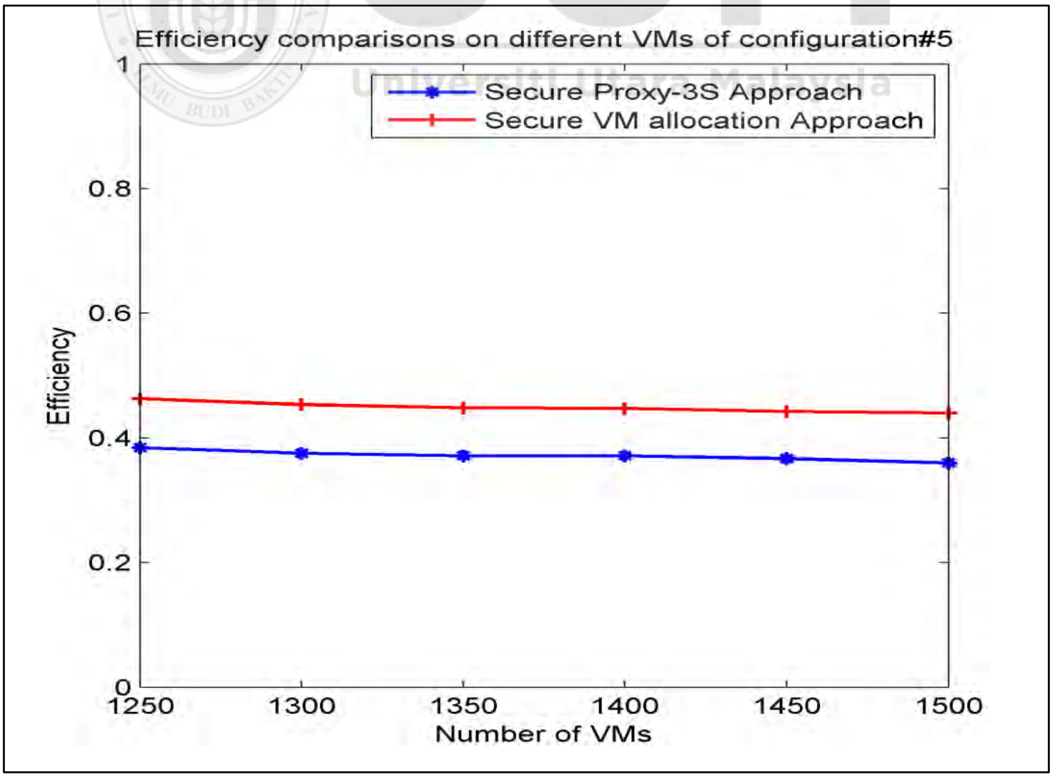


Figure 6.10. Efficiency comparison in configuration 5 (1500 VMs spread).

We have also implemented the coverage algorithm of the works [12, 83]. Table 6.5 shows the coverage result comparisons on different configurations and VMs. As we can see, the coverage for the proposed method is considerably lower in all experimented configurations compared to related work [12, 83]. Although, Proxy-3S method using the secure VM communication policy, the coverage can reach 0.45 in different configurations compared to the work of [12, 83]. This shows that the proposed method allows low efficiency and coverage values in different configuration and VMs using Proxy-3S. Figure 6.11, Figure 6.12, Figure 6.13, Figure 6.14, Figure 6.15 show the coverage comparison between this work and similar work [12, 83] respectively under various configuration with a different number of VMs spreading (300, 600, 900,1200,1500).

Table 6.5
Coverage comparisons on different configurations and VMs

Number of VMs	Configuration 1	Configuration 2	Configuration 3
Secure Proxy- 3S			
50 - 350 – 650	0.2857	0.4285	0.4484
100 - 400 – 700	0.2222	0.4190	0.4497
150 - 450 – 750	0.3372	0.4304	0.4341
200 - 500 – 800	0.4259	0.4307	0.4343
250 - 550 -850	0.4090	0.4507	0.4253
300 - 600 – 900	0.4150	0.4498	0.4456
Secure VM allocation [12, 83]			
50 - 350 – 650	0.3076	0.4730	0.5136
100 - 400 – 700	0.2542	0.4611	0.5111
150 - 450 – 750	0.3846	0.4786	0.4985
200 - 500 – 800	0.4747	0.4745	0.4945
250 - 550 -850	0.4426	0.5078	0.4910
300 - 600 – 900	0.4722	0.5109	0.5121

Number of VMs	Configuration 4	Configuration 5
Secure Proxy-3S		
950 - 1250	0.4525	0.4571
1000 - 1300	0.4503	0.4492
1050 - 1350	0.4533	0.4476
1100 - 1400	0.4515	0.4442
1150 - 1450	0.4519	0.4342
1200 - 1500	0.4535	0.4319
Secure VM allocation [12, 83]		
950 - 1250	0.5185	0.5160
1000 - 1300	0.5152	0.5098
1050 - 1350	0.5176	0.5094
1100 - 1400	0.5107	0.5045
1150 - 1450	0.5102	0.4927
1200 - 1500	0.5106	0.4936

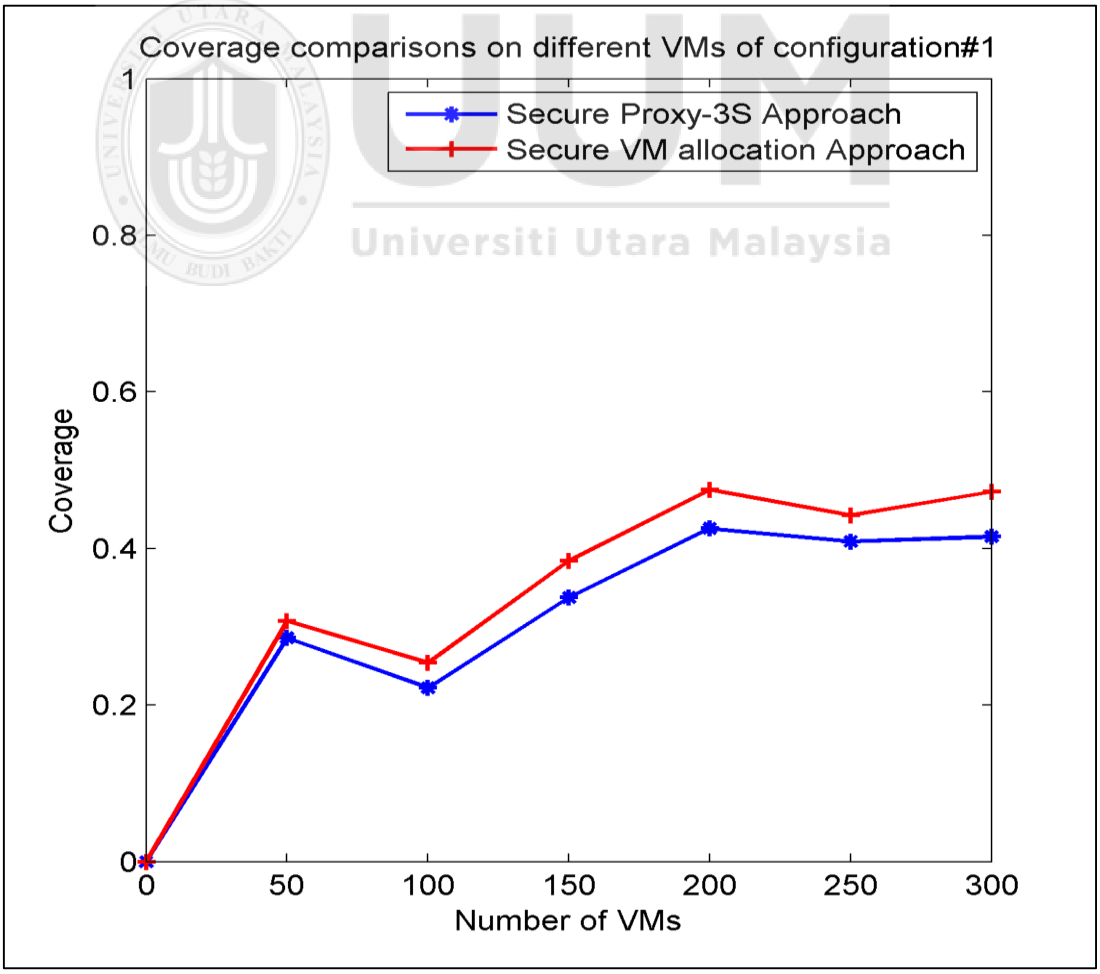


Figure 6.11. Coverage comparison in configuration 1 (300 VMs spread).

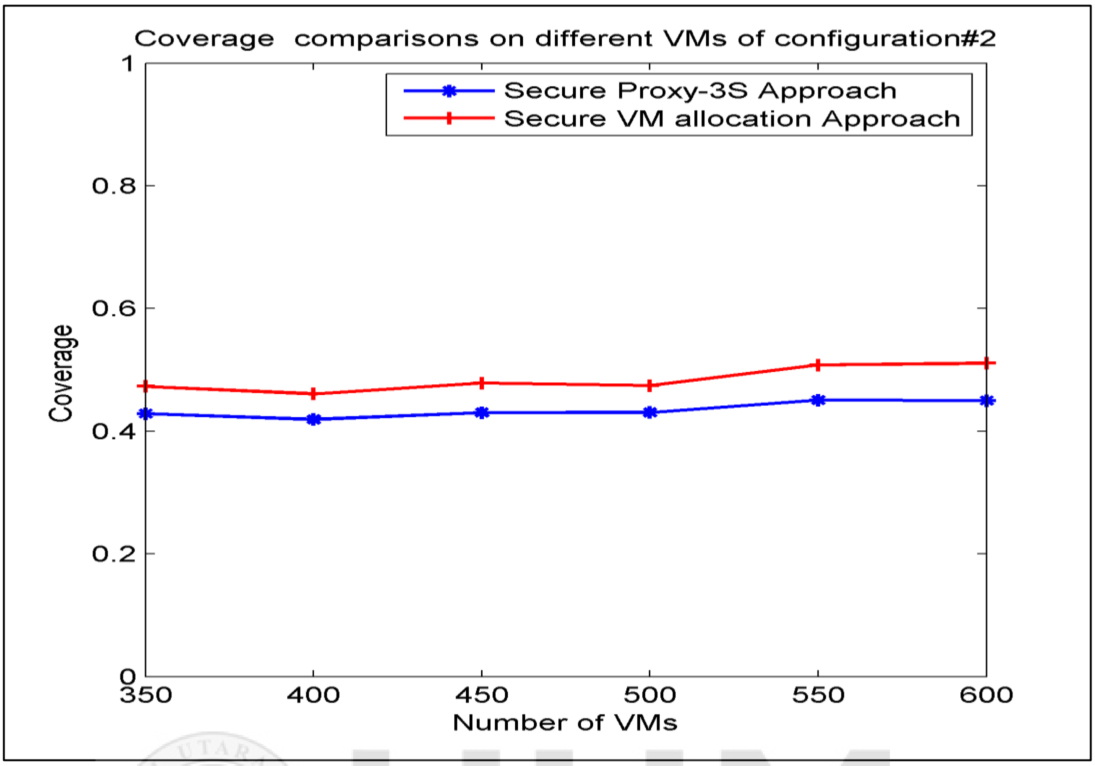


Figure 6.12. Coverage comparison in configuration 2 (600 VMs spread).

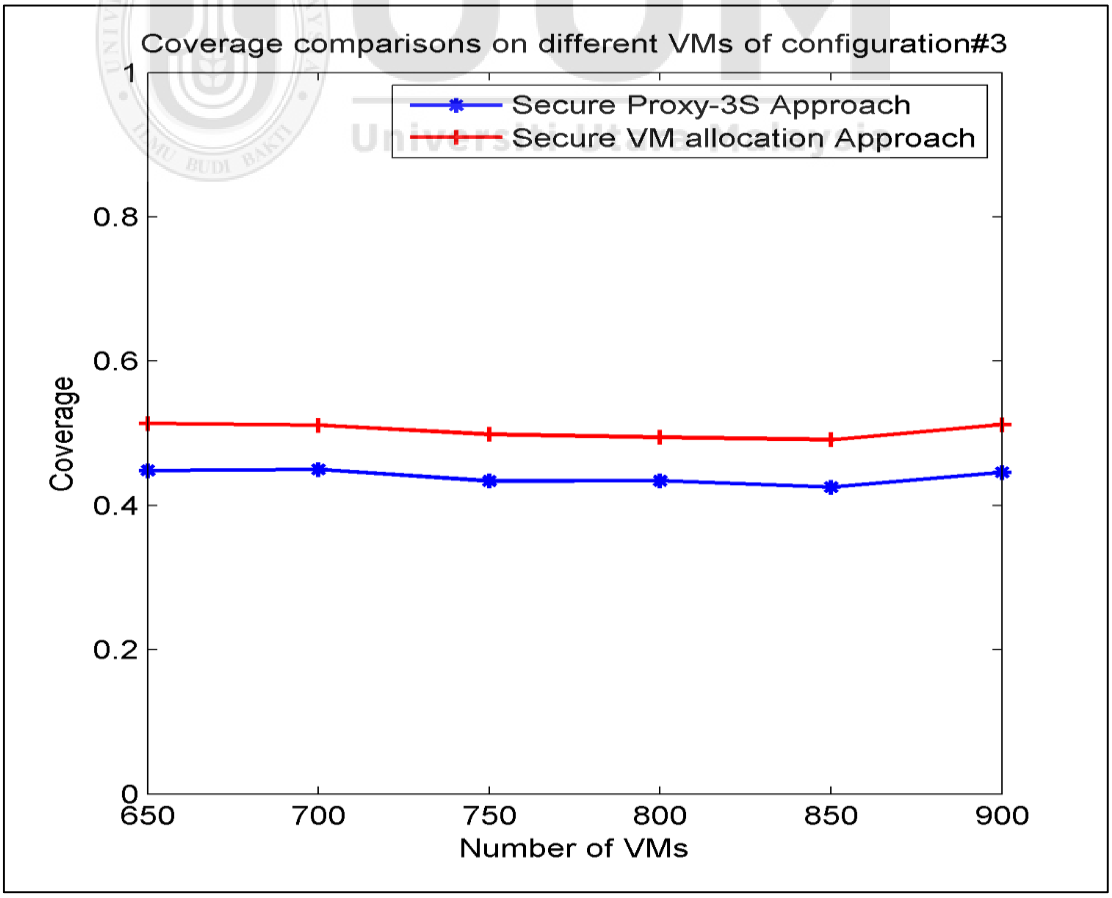


Figure 6.13. Coverage comparison in configuration 3 (900 VMs spread).

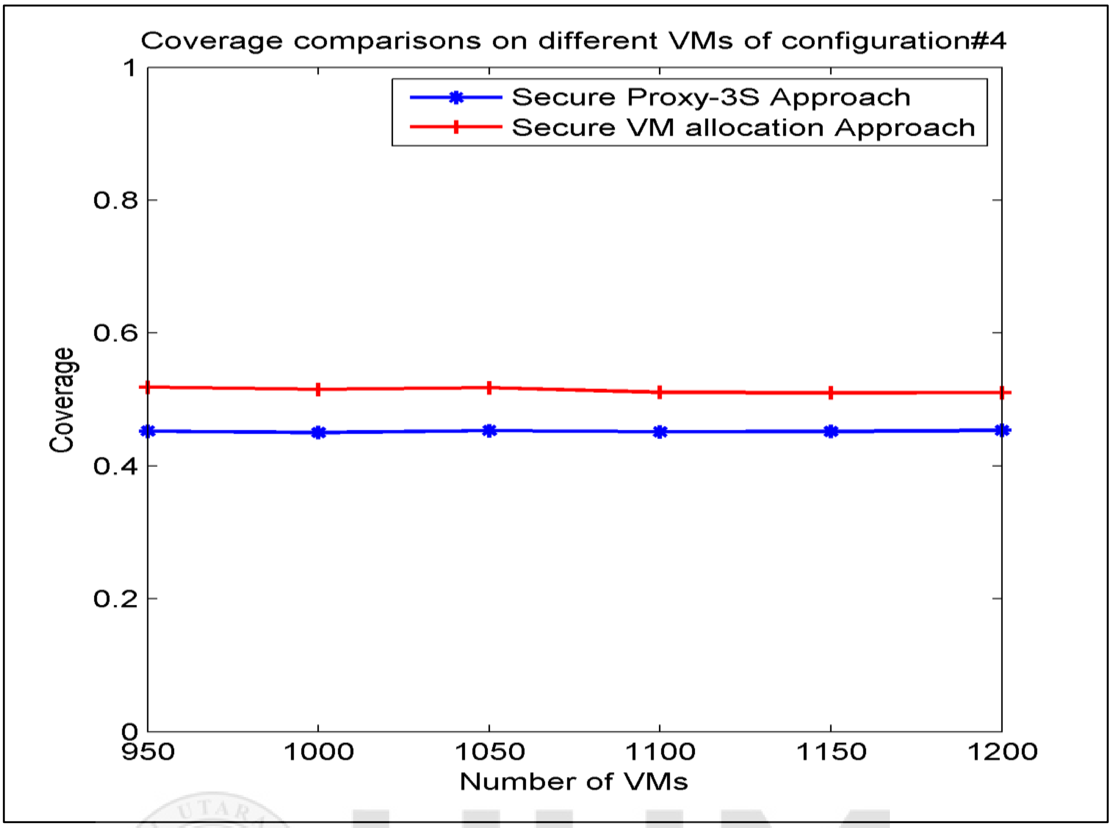


Figure 6.14. Coverage comparison in configuration 4 (1200 VMs spread).

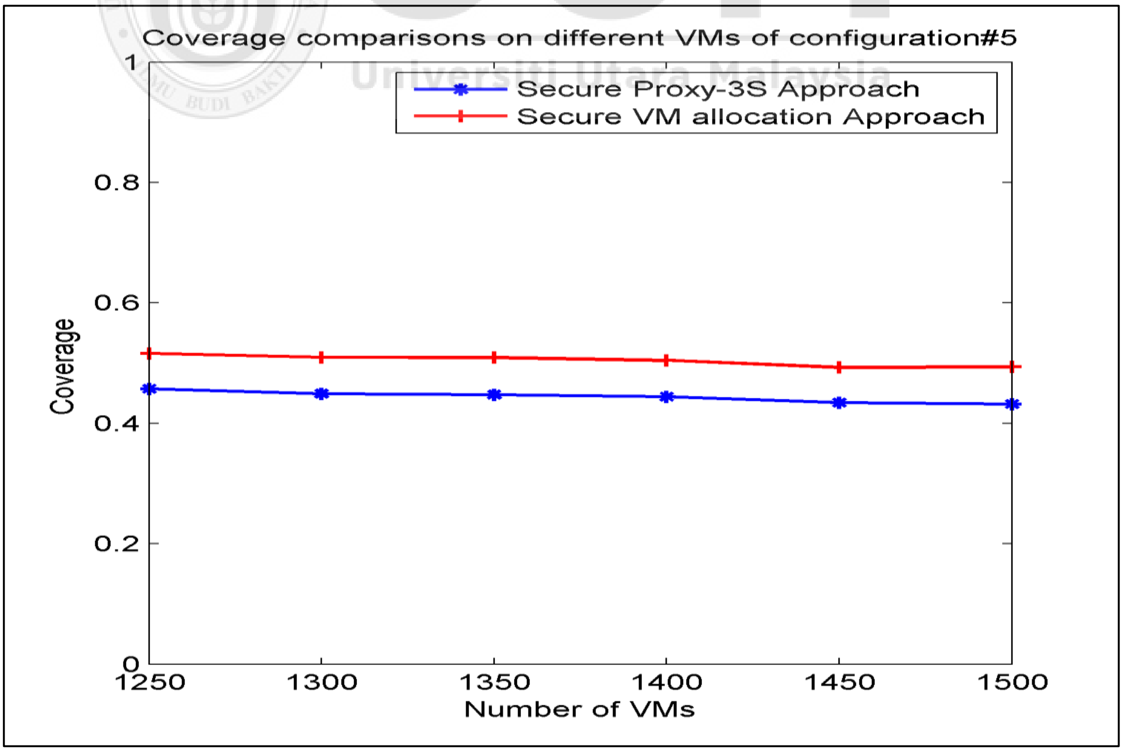


Figure 6.15. Coverage comparison in configuration 5 (1500 VMs spread).

6.4 Performances and Security Comparison Details

In order to validate the efficiency and security of the proposed approach, we compare the obtained results using our approach with some similar related works and simulation tools available in the literature.

6.4.1 Comparison of processing execution time

We have evaluated the processing execution time with different cloud configurations after integrating security layers and compared to NetworkCloudSim. As shown in Table 6.6, we observe that the processing execution time of the SecNetworkCloudSim tool is very satisfactory ranging from 44142ms to 467572ms compared to NetworkCloudSim in five different cloud configurations.

Table 6.6
Evaluating the execution time of SecNetworkCloudSim compared to NetworkCloudSim

Configurations	NetworkCloudSim (ms)	SecNetworkCloudSim (ms)
Configuration 1	42278	44142
Configuration 2	76412	78353
Configuration 3	117942	118272
Configuration 4	157849	367999
Configuration 5	186413	467572

There is a slight difference in term of the processing execution time of both results. The difference of the results is due to the security layers integrated into SecNetworkCloudSim. The execution time of NetworkCloudSim refers to estimated simulation time of distributed application running on the cloud and the time spend in order to completely finish the process. Whereas, the SecNetworkCloudSim execution time refers to the time spent by the distributed application in order to process including the checking time of the users control access, checking VM security keys, secure VMs allocation, and checking the VMs communications. However, SecNetworkCloudSim results showed a low-performance impact which does not irritate the users and does not affect the service level agreement between the user and the cloud provider. Figure 6.16 shows the comparison results of the execution time of different tools.

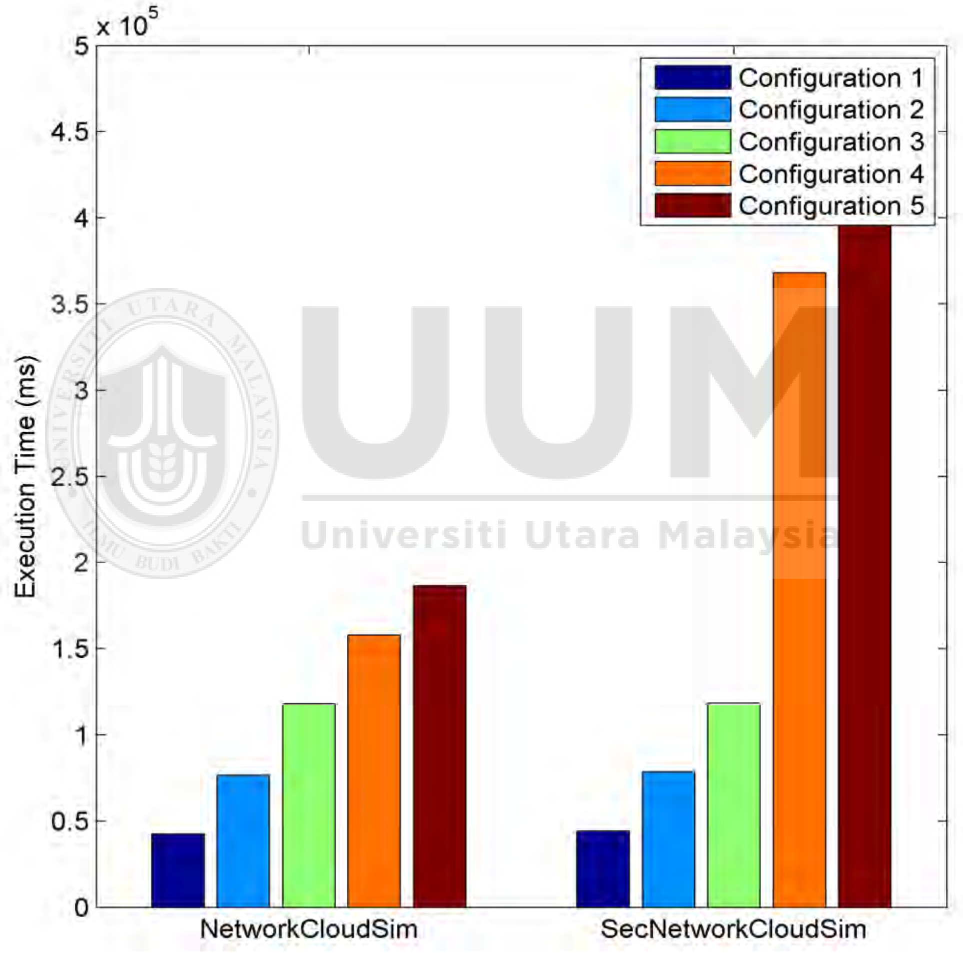


Figure 6.16 Evaluating the processing execution time of SecNetworkCloudSim compared to NetworkCloudSim using different configurations

6.4.2 Security comparison of mobile user access control mechanism

Table 6.7 shows a comparison of the security mechanisms of mobile user access control and average security results between the proposed method and related works described in [12, 156] in terms of security checking time of public-private generation keys and encrypted mobile device's MAC address verification.

Table 6.7
Features comparison between works in [12, 156] and the proposed Mobile User Access Control

Approaches	Data Security of Mobile User	Average security checking time(ms)
Co-residency [12]	No	-
Hash RSA-1024 [156]	Yes	257.2
Proposed MUAC	Yes	75.4

The proposed mobile user access control is more effective and practicable than proposed related techniques. The reason is that our approach is trust-based VM-communication model compared to work presented in [12] which is not considered. In addition, in the proposed mobile user access control, the mobile device's MAC (password) and data request are encrypsted. This offers more security contrarily to work presented in [156] where the password is embedded without encryption. Finally, the checking security time required by the proposed method is on average, while it is high in works presented in [12, 156]. The checking security time refers to the time needed to generate RSA or Diffie-Hellman keys and time to hash a message.

6.4.3 Comparison of the efficiency and coverage with details of VMs communication

The efficiency and coverage ratios are important security factors in distributed VM-based application, which reflects the ability of the system to check the security of a large number of VMs in a specific time. In order to evaluate the impact of different approaches on the efficiency and coverage ratios. We have implemented four scenarios-based VM-communication process model with different types of VM's communications (5 – 100) to detect malicious communication or VM attacker:

- VM legal communicates with VM legal
- VM legal communicate with VM legal: One of the VM-legal is an attacker but behave like legal.
- VM attacker with VM legal.
- VM attacker with VM attacker.

We distinguish between intra and inter-communication of VMs of the co-located and remote VMs respectively as illustrated in Figure 6.17. Our assumption claims that the security of communication among co-located and remote VMs implies that VM's identify, VM's location and the activity of VM should be consider into the design of VM-based distributed systems as follows:

- Direct VM's communication: either co-located or remote VMs (legal or attacker) able to communicate among each other in a specific shared time and space. The VMs are randomly selected.
- Trust-based VM's communication: consider only VMs to trust other VMs to perform communication tasks at a specific time.

- Group-based VM's communication: communication by co-located VMs to achieve a common system's task.
- Federated VM's communication: communication by co-located VMs of a given host in coherence with other co-located VMs of another host.

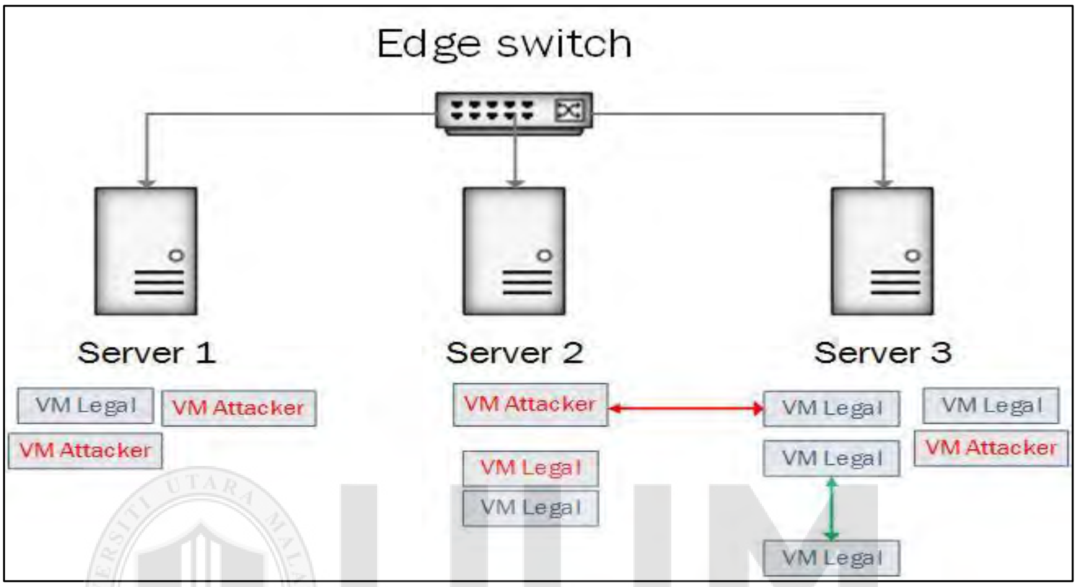


Figure 6.17. Intra and inter-communication of VMs of the co-located and remote VMs

The VM legal with the red font is a VM attacker but behaving as legal which is co-located by another VM legal. Whereas, the red arrow refers to successful malicious communication between VM attacker and VM legal (i.e. unsuccessful malicious communication refers to a communication between VM attacker and another VM attacker). While, the green arrow refers to successful legal communication between VM legal and another VM legal (i.e. unsuccessful legal communication refers to a communication between VM legal and another VM legal, and the latter is an attacker behaving as legal VM).

We have compared our proposed approach and the related work [12, 83] with different number of VMs either legal or attacker that increased gradually after each run (i.e. 50 VMs either legal or attackers). Thus, we can simulate the scalability experiment for a large number of VMs and the communication between them. With the direct mode of VMs communication, we create a specific number of VM communication (i.e. 5, 10, 15, 20… 100) permits to simulate the VMs communication scenario where VMs need to perform a common task based on their communication probability. For example, VM 1 and VM 2 in different hosts with shared variable (perhaps shared memory and time). VM 1 updates shared variable by modifying the state of its task and VM 2 reads the shared variable.

6.4.3.1 Comparison results for the smaller size configuration set VMs

We have evaluated the efficiency and coverage with different number of VMs (50 – 300) and a varied number of VMs communication (5 – 100). We use configurations as shown in Table 6.1. As shown in Table 6.8 we observed that the obtained results of the proposed approach are much better than related work of [12, 83]. In 300 VMs, the cloud system with our approach has better efficiency and coverage ratios. To clarify more, the number of attackers is equal to 165 and the legal is 135. Where 69 attackers success to co-locate with VMs legal. Once the detection of malicious VMs, malicious VMs communication is activated, the proxy detects 3 VMs attacker which was behaving as legal VMs. Moreover, the proxy detects 22 successful malicious communication that VMs attacker launched and succeed to communicate with legal VMs as well as 17 Unsuccessful malicious communication launched the VMs attacker fails to communicate with VMs legal. In another side, 22 legal communications have launched by a legal VMs whether successful or unsuccessful communication.

After this updates that the proxy receives, the number of VMs attacker and the VMs legal are changed compared to their first values. Hence, the number of the VM attacker will be equal to the total of a number of new VMs attacker and total number of all type of malicious communication (successful and unsuccessful) that launched by VMs attacker. Whereas, the VMs legal will be equal to the total of new VMs legal (minus the attacker VMs that behaving as legal) and total number of all type of legal communication that launched by Legal VMs. Therefore, in 300 VMs and after counting the new total number of VMs legal and VMs attacker, the proxy found 154 VMs legal and 207 VMs attacker respectively. Thus, the proposed approach considers inter and intra-VMs communication in several experimented configurations. The similar work adopts only co-residency for identifying VM's attacks.

6.4.3.2 Comparison results for the moderate size configuration set VMs

We evaluated the effectiveness of Proxy-3S with a moderate size of VMs ranged from 350 to 600 in term of efficiency and coverage. Table 6.9 shows the obtained results for the moderate size configuration set VMs (600). In 600s VMs, coverage and efficiency of our approach are quite better than the related work [12, 83]. The number of VMs attacker detected by the proxy is equal to 340 and the number of VMs legal is 260. From 340 VMs attacker, 130 attacker VMs have successfully co-located with legal VMs which is approximately equal to 38% of the total number of VMs attackers. In addition, the proxy detects 23 fake VMs legal, 41 successful malicious communication, 96 unsuccessful malicious communication and 95 successful legal VMs communications. Once the full number of VMs legal and attacker that including the number of all communication types has been updated, the total number of VMs legal and its communication becomes equal to 294 and the total number of VMs attacker and its malicious communication becomes equal to 502.

6.4.3.3 Comparison results for the large size configuration set VMs

In large-size VMs set configuration, we have also studied how the large number of users' VMs affects efficiency and coverage ratios where the number of VMs grows from 650 to 900. We compare the proposed approach with similar related work [12, 83]. In Table 6.10, we show the performance of experimented works in large size configuration set VMs. We can see that the efficiency and coverage ratios for the proxy-3S are considerably low in all experiments. However, under 900 VMs, the proxy-3S detects 143 VMs attacker co-located with legal VMs. It detects also 17 fake VMs legal, which considers as VMs attacker and were co-located by another VMs attacker. Indeed, the number of remote co-location is reduced by 17 co-locations to become 126 instead of 143, which is because that the integration of the communication policy module detects more malicious and legal communications. Proposed proxy-3S detects 53 successful remote co-located with target legal VMs and 22 unsuccessful remote co-located VMs among 261 established legal communications. Once the full number of VMs legal and attacker that including the number of all communication types has been updated, the total number of VMs legal and its communication becomes equal to 875 and the total number of VMs attacker and its malicious communication becomes equal to 365.

In summary, from the performance of low efficiency and coverage ratios, we believe this is good for large-scale VMs based distributed mobile application that needs to support runtime secure allocation strategy and communication security checking in the cloud environment.

Table 6.8

Comparison for the smallest size configuration set between works in [12, 83] and the proposed work

# VMs	#VMs Legal	# VMs Attacker	# Co-located VMs	# Remote-located VMs	New Detected Attacker	Successful malicious communication	Unsuccessful malicious communication	Total legal communication	New VM legal including legal communication	New VM Attacker including attacker communication	Proposed approach		Secure VM allocation [12, 83]	
											Efficiency	Coverage	Efficiency	Coverage
50	24	26	10	9	1	2	0	2	25	29	0.3103	0.3600	0.3846	0.4166
100	47	53	23	23	0	5	6	5	52	64	0.3593	0.4423	0.4339	0.4893
150	73	77	31	29	2	10	10	10	81	99	0.2929	0.3580	0.4025	0.4246
200	95	105	45	45	0	11	10	11	106	126	0.3571	0.4245	0.4285	0.4736
250	114	136	54	53	1	17	11	17	130	165	0.3212	0.4076	0.3970	0.4736
300	135	165	69	66	3	22	17	22	154	207	0.3188	0.4285	0.4181	0.5111

Table 6.9

Comparison for the moderate size configuration set between works in [12, 83] and the proposed work

# VMs	#VMs Legal	# VMs Attacker	# Co-located VMs	# Remote-located VMs	New Detected Attacker	Successful malicious communication	Unsuccessful malicious communication	Total legal communication	New VM legal including legal communication	New VM Attacker including attacker communication	Proposed approach		Secure VM allocation [12, 83]	
											Efficiency	Coverage	Efficiency	Coverage
350	220	130	88	77	10	15	13	20	230	168	0.4583	0.3347	0.6769	0.4000
400	316	84	32	26	6	3	4	15	325	97	0.2680	0.0822	0.3809	0.1012
450	339	111	44	31	13	22	48	80	406	204	0.1591	7.70. 0	0.3963	0.1297
500	376	124	41	38	3	28	51	69	442	206	0,1844	0,0859	0,3306	0,1090
550	403	147	53	46	7	19	16	71	467	189	0,2433	0,0985	0,3605	0,1315
600	260	340	130	105	25	41	96	59	294	502	0,2091	0,3571	0,3823	0,5000

Table 6.10

Comparison for the largest size configuration set between works in [12, 83] and the proposed work

# VMs	#VMs Legal	# VMs Attacker	# Co-located VMs	# Remote-located VMs	New Detected Attacker	Successful malicious communication	Unsuccessful malicious communication	Total legal communication	New VM legal including legal communication	New VM Attacker including attacker communication	Proposed approach		Secure VM allocation [12, 83]	
											Efficiency	Coverage	Efficiency	Coverage
650	429	221	70	64	6	33	48	150	573	308	0,2077	0,1116	0,3167	0,1631
700	520	180	58	48	10	15	26	210	720	231	0,2077	0,0666	0,3222	0,1115
750	616	134	26	24	2	14	21	102	716	171	0,1403	0,0335	0,1940	0,0422
800	567	233	79	67	12	41	55	75	630	341	0,1964	0,1063	0,3390	0,1393
850	490	360	240	207	33	77	83	68	525	553	0,3743	0,3942	0,6666	0,4897
900	631	269	143	126	17	53	22	261	875	361	0,3490	0,1440	0,5315	0,2266

6.4.4 General drawbacks and comparison of security degree

Table 6.11 presents the advantages and drawbacks between the proposed method and related works described in [12, 157]. As we can see from table 6.8, table 6.9 and table 6.10 that the proposed technique is better than some recent methods available in the literature. The new security approach not only proves the effectiveness of the proposed method but also makes it practicable in spite of using different configurations applications.

Table 6.11
Performances comparison.

Works	**Purpose**	**Security degree**	**Encryption/Decryption**	**General Drawbacks**
Secure VM allocation policy [12]	VMs Co-residency	Medium	No	High efficiency and coverage of the attacks
Hybrid Hashing security Algorithm [157]	Mobile User access control	Good	Hash-RSA	Some data are embedded and high processing time
Proposed method	Remote VM's co-residency	Good	Hash- Diffie Hellman	High processing time if there are enormous VMs' communications

6.5 Summary

With the appearance of new advanced technology, preserving the security of distributed mobile applications and their handled sensitive data becomes a fundamental and necessary. In this research, we have proposed a new Three Policies Secure Cloud Proxy (Proxy-3S) for preventing the intrusion of tenants' sensitive information on the cloud environment exactly at the virtualization layer. This chapter presented an implementation and evaluation of our proposed approach compared to related works exist in the literature. Several experiments were conducted using real-world healthcare applications. Experimental results have shown that the proposed approach provides lower efficiency and coverage in preserving data integrity and confidentiality. A comparison with some recent methods shows that the proposed three security policies within proxy is more efficient and decreases the coverage of the attackers. However, the time checking complexity must be improved in future works.

CHAPTER SEVEN

CONCLUSION AND FUTURE WORKS

7.1 Introduction

This research developed a secure approach for ensuring the protection of sensitive data of the intensive distributed application that run on the mobile cloud computing environment. The approach is a software solution based on an intermediary proxy between the users that intend to leverage the cloud services and the cloud side environment. A conclusion for the research is presented in this chapter, where we review and summarize the main points including objectives, contributions and limitations of the work as well as several directions for future works.

The organization of this chapter is as follows: Section 7.1 provides a concise introduction along with the organization of the chapter. The work carried out and the important points of the research are briefly highlighted in Section 7.2. Section 7.3 presents a summary of the research contributions. The limitations of this research are presented and discussed in Section 7.4. Finally, Section 7.5 provides the future challenges that need to be tackled by researchers in order to further secure the user's data from VMs co-remote communication and VMs co-residents attack in a mobile cloud environment.

7.2 Research Summary

Mobile devices such as smartphones and tablets have gained enormous popularity and interest in people's daily life. Many sophisticated mobile applications have spurred opportunities for increased efficiency and real innovation in several domains such as medical, face recognition and mobile augmented reality. These applications require high computational resources to run intensive tasks. Due to computing and storage resources limitations, mobile devices migrate these applications to rich-resources on the cloud infrastructure. Thus, running mobile applications on the cloud servers address the mobile devices resources restrictions which produce the tremendous paradigm of Mobile Cloud Computing (MCC).

The cloud enables on-demand access to a shared pool of virtualized computing resources that users can exploit for handling and deploying their mobile device applications [158]. Cloud computing allows users to leverage on virtualization technology in which server resources (hardware) are divided into multiple virtual machines (VMs) for mobile users as well as several executions being carried out at the same time. It permits offloading of all types of computational tasks on a VM (phone clone) system level. Each VM can execute some computational-intensive tasks on behalf of the mobile device and send the result back afterward [159].

With the motive of securing virtual machines, the hypervisor is the software level that creates, runs and controls the virtual machines. This layer resides between the hardware and the users' virtual machines. However, the researchers in [18] proved that a malicious user can circumvent the hypervisor and allocate their malicious VMs. Other issues that virtualization brings when applied to mobile cloud computing environment are unauthorized access of users to cloud services, VMs communication within a virtualized environment, which all make the risk of losing users data in Cloud side [3, 4].

The users' VMs are isolated from each other even when they run on the same cloud physical machines in order to avoid sharing VMs' sensitive data, which may risk exposing confidential data. Unfortunately, attackers are finding new ways to circumvent weaker isolation of VMs, such as damaging the data confidentiality of a virtual machine and violating privacy through building attackers channels [13]. Researchers have shown that a malicious user can break the isolation by building many side channels between VMs for obtaining private data that process on legal users' virtual machine. The malicious users deliver their VMs on the same server that hosts legitimate users virtual machines in order to co-locate with them and steal their information.

Many security techniques and approaches were proposed [39, 117] to tackle security issues in virtualization. However, some solutions were based on changing or replacing the existing hardware of the cloud platforms. The solutions were not practical due to the high cost. In other solutions, some researchers [83, 30] focused on securing the VMs only in the same host (co-location issue) by introducing new secure VMs allocation policies [147]. However, none of the proposed solutions considered security issues of VMs communication in the cloud environment. Due to unsecured communication gateways, the sensitive data of distributed mobile applications face many threats from malicious behavior. Such behavior can affect the overall performance of the cloud service provider as well as lead to data alteration, revealing information, and deletion of information.

In order to address the above challenges, this research proposed a software approach based on the secure proxy. It aims to enhance the protection of sensitive data integrity and confidentiality while the sensitive data is processed on the virtual machines in the cloud. It protects the sensitive data against three common attacks on virtualization layer such as co-resident attacks, hypervisor attacks, and distributed attacks. The main contributions of this research are:

- A novel Three Policies Secure Cloud Proxy (Proxy-3S) provides a highly secure way to help protecting distributed VMs from any emerging threats and decrease the probability of the insider and outsider threats and frauds.
- To deal with the security of distributed tasks deployed on different hosts on the cloud, the Proxy-3S restricts access to the cloud only to the authorized applications by using key management encryption capabilities and secure management techniques of virtual machines. Therefore, unauthorized VMs cannot communicate with the legitimate VMs that contain sensitive tasks of the authorized applications.
- To validate the efficiency and the coverage of the proposed security techniques, a real healthcare distributed mobile application is used in the experiment. The experimental results show that the proposed approach is superior to the current solution [12] and provides robust access and protects the privacy of data in the connected VMs.

Despite the widespread release of several cloud simulators, controlling user's access and protecting data exchanges in distributed mobile applications over the cloud is considered a major challenge. This research introduces a new NetworkCloudSim extension called SecNetworkCloudSim. It is a secure mobile simulation tool extended with a new layer called Proxy-3S to ensure secure access to sensitive data hosted on the mobile devices and distributed cloud's servers. This tool enables users to specify their requests and allows them to connect through a proxy to perform secure cloud services access, deploys their tasks in the cloud using VM allocation policy, ensure the secure access of data shared between collaborative VMs through the Cloud and derive high efficiency and coverage rates. Most importantly, due to the secure aspects of proxy, user's distributed tasks can be performed without alterations on different underlying proxy's security policies.

The evaluation of SecNetworkCloudSim showed a low-performance impact compared to NetworkCloudSim, which does not irritate the users and does not negatively affect the Service Level Agreement (SLA) between the user and the cloud provider.

7.3 Research contribution

The contribution presented in this research was mainly focusing on developing a new security approach based on software solution called Three policies Secure cloud Proxy (Proxy-3S) by taking into account the efficiency, coverage and execution time metrics. Our goal is to enhance the security measures of sensitive data by verifying the efficiency and coverage during the communication. The proposed solution cannot only protect the users' sensitive data integrity and confidentiality that process inside VMs on the mobile cloud environment but also used to authenticate the transmitted sensitive data across unsecured attackers.

a. The first axis concerned with proposing a new user control access policy for preventing the unauthorized access of mobile users that would leverage the cloud service provider and allocate their malicious virtual machines.
    - A new mathematical model for user authentication was proposed where malicious users can not violate and the damage sensitive data in distributed mobile applications.
    - The use of Hash-Diffie Hellman method for checking the validity check of the user and get robust user's access control technique.

b. The second axis is interested in developing a secure virtual machine manager allocation policy for avoiding the deployment of malicious virtual machines on the cloud hosts.

   - A novel secure task deployment on the VMs strategy based secure key generation was proposed. This strategy presents a secure and robust virtual machine allocation method to provide a significant contribution in attacks authentication.
   - A new secure allocation of VMs using under the least, most and random policy was proposed. Indeed, attacks due to malicious VMs co-location with legitimate VMs is reduced using secure and safe hosts, which gives high efficiency.

c. The third axis was interested in proposing new secure virtual machines communication policy for ensuring the secure exchanges of sensitive data among VMs deployed on different cloud's hosts in the cloud environment. The principle of this policy is to perform the Hash-Diffie Hellman encryption/decryption process of exchanged sensitive information between VMs across the insecure public network by generating shared security keys, then checks the probability of unauthorized access of the VM to sensitive data. Finally, each remote co-location from the distributed communication between VMs is evaluated using new efficiency and coverage metrics. The experiments results showed better efficiency and coverage than the related work were achieved [12, 83].

d. The fourth axis was interested in developing a secure mobile cloud environment called SecNetworkCloudSim, which is an extension of the popular used NetworkCloudSim for simulation of distributed mobile intensive applications. It based on three policies: user access control, secure virtual machine allocation and secure communication between VMs. The environment exploits the well-known Poisson distribution law to generate VMs and adds support for modeling the VM-based task execution and simulating the execution workflow. Moreover, this environment could detect if the VM sensitive data is violated or not and consequently deciding the VM authorization.

In this research, the contributed approach has the below features.

- Secures VM's data against remote and co-resident attacks and reduces significantly the probability of unauthorized data exchange using new VM's communication security policy
- Improves security metrics (e.g. the efficiency and the coverage) by VM's communication aspect and uses security metrics to evaluate security policies.
- Conducts real-life experiments on the widely used simulation platform NetworkCloudSim and very cleverly through SecNetworkCloudSim to validate the approach.

The main advantages of the proposed approach compared to existing related works are better efficiency and coverage in the cloud environment after considering the communication aspect, as well as, less execution time for the VM allocation process, without forgetting that the proposed security policies achieve more robustness.

7.4 Research limitations

Although this research has successfully designed, developed, tested and evaluated with several experiments that have shown promising results for high-level data protection and good efficiency rating compared with existing works. These experiments have not tested in real worlds cloud platform but only in a simulation environment. The simulation allows researchers to do repeatable experiments to get the highest accuracy level of the performance demanded with lower cost overhead. However, the simulation does not provide optimal results like real testbeds, which give high results accuracy.

The approach is focusing on securing the virtualized environment, exactly the users' data that run inside thin VMs. However, the approach proposed has not taken into consideration the pre-offloading phase. To clarify more, the pre-offloading stage means that before the user offloads the intensive application from the mobile device to the cloud side. The attacker can sniff the data before the partitioning of the intensive application. Further, the attacker can deliver more VMs, whether on the mobile device or in the cloud. So, the checking security time can be slightly higher when verifying the eligibility of communicated VMs especially if they are allocated on the different locations where some VMs allocated in the mobile device and other VMs allocated in Cloud host. This limitation may violate the service level agreement and degrade the service quality provided for mobile cloud users.

7.5 Future Works

Beyond the solution based on security policies proposed in this research, several future directions still need to be carried out in order to protect the users' data in a mobile cloud computing environment. The following are some challenges that can be addressed in the future to extend this work:

Firstly, the approach needs to be tested and evaluated using real cloud systems (Amazon EC2, Google App Engine) and open source Hypervisors like Xen and KVM in order to evaluate the approach performance in preventing malicious users' access to the cloud, c-location and remote co-location (distributed attacks). As well, test the approach with real intensive and sensitive applications like banking application and healthcare application which will be more effective and give real results about the performance of the approach.

Secondly, reducing the security management checking time is considered as the main challenge that needs to be carried out in this proposed approach. As mention as a limitation of this research, the approach proposed will make delay for controlling and verifying the whole VMs number on the cloud, which may cause a decrease of the cloud provider service quality and lose the trust of the cloud's tenants. Thus, incorporating the optimization method will be an excellent future work to reduce the security checking time and make the approach more robust, flexible and adaptive for the current cloud systems.

Thirdly, when Proxy 3-S allocates a VM in a specific host. This VM must run on that specific host in order to be controlled by the proxy. A malicious VM can change its location in the same host to co-locate in other placement. As a future enhancement for this approach, it is necessary to detect any Co-placement of VMs. The proxy will be able to detect the malicious placement of VMs from a specific host. Depend on the identifier of the VM and identifier of the host, the proxy detects that identifier VM is not assigned to a specific identifier host.

Finally, chapter 5 presented our new secure simulator: SecNetworkCloudSim. This new simulator needs to be more tested and evaluated with different attack scenarios to evaluate the various parameters such as response time, time delay, packet rate and processing time. This simulator can analyze such attack of VMs communication and users access control, but still need more enhancement in order to make it able to analyze other exiting attacks that affect the users' data on the mobile cloud environment.

REFERENCES

[1] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, “Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing,” Information Sciences. 2017 Feb 10; 379:42-61.

[2] V. Sundararaj, “Optimal Task Assignment in Mobile Cloud Computing by Queue Based Ant - Bee Algorithm,” Wireless Personal Communications. 2019 Jan 15; 104(1):173-97.

[3] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, “Mobile cloud computing: A survey, state of art and future directions,” Mobile Networks and Applications. 2014 Apr 1; 19(2):133-43.

[4] B. Zhou and R. Buyya, “Augmentation Techniques for Mobile Cloud Computing,” ACM Computing Surveys (CSUR). 2018 Jan 4; vol.51(1):1-38.

[5] M. Deng and M. Petkovi, “A home healthcare system in the cloud – addressing security and privacy challenges,” In 2011 IEEE 4th International Conference on Cloud Computing 2011 Jul 4 (pp. 549-556).

[6] P. Li, J. Li, Z. Huang, C. Z. Gao, W. Bin Chen, and K. Chen, “Privacy-preserving outsourced classification in cloud computing,” Cluster Computing. 2018 Mar 1; 21(1):277-86.

[7] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, “DDoS attacks in cloud computing: Issues, taxonomy, and future directions,” Computer Communications. 2017 Jul 15; vol.107:30-48.

[8] Y. Wang and I. C. D. Wang, “A Survey of Mobile Cloud Computing Applications : Perspectives and Challenges,” Wireless Personal Communications. 2015 Feb 1; 80(4):1607-23.

[9] M. Ali, S. U. Khan, and A. V. Vasilakos, “Security in cloud computing: Opportunities and challenges,” Information sciences. 2015 Jun 1; 305:357-83.

[10] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing". Journal of network and computer applications. 2011 Jan 1; 34 (1):1-1.

[11] T. Dillon, C. Wu, and E. Chang, “Cloud Computing : Issues and Challenges,” In 2010 24th IEEE international conference on advanced information networking and applications 2010 Apr 20 (pp. 27-33).

[12] Y. Han, J. Chan, T. Alpcan, and C. Leckie, “Using Virtual Machine Allocation Policies to Defend against Co-resident Attacks in Cloud Computing,” IEEE Transactions on Dependable and Secure Computing. 2015 May 4; 14(1):95-108.

[13] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, “Security and privacy challenges in mobile cloud computing: Survey and way ahead,” Journal of Network and Computer Applications. 2017 Apr 15; 84:38-54.

[14] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud

computing : architecture , applications , and approaches," Wireless communications and mobile computing. 2013 Dec 25;13 (18):1587-611.

[15] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: Issues and challenges," Applied computing and informatics. 2018 Jan 1;14 (1):1-6.

[16] A. N. Khan, M. L. Mat Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," Future Generation Computer Systems. 2013 Jul 1;29 (5):1278-99.

[17] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," In Proceedings of the 16th ACM conference on Computer and communications security 2009 Nov 9 (pp. 199-212).

[18] D. Sgandurra and E. Lupu, "Evolution of Attacks, Threat Models, and Solutions for Virtualized Systems," ACM Computing Surveys (CSUR). 2016 Feb 8; 48(3):1-38.

[19] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," In Proceedings of the 34th annual international symposium on Computer architecture 2007 Jun 9 (pp. 494-505).

[20] Z. Wang and R. B. Lee, "Covert and Side Channels due to Processor Architecture * In 2006 22nd Annual Computer Security Applications Conference (ACSAC'06) 2006 Dec 11 (pp. 473-482). IEEE.

[21] A. Aviram, S. Hu, and B. Ford, "Determinating Timing Channels in Compute Clouds," In Proceedings of the 2010 ACM workshop on Cloud computing security workshop 2010 Oct 8 (pp. 103-108).

[22] B. C. Vattikonda, "Eliminating Fine Grained Timers in Xen Categories and Subject Descriptors," In Proceedings of the 3rd ACM workshop on Cloud computing security workshop 2011 Oct 21 (pp. 41-46).

[23] J. Wu, L. Ding, Y. Lin, N. Min-Allah, and Y. Wang, "XenPump: A new method to mitigate timing channel in cloud computing," In 2012 IEEE Fifth International Conference on Cloud Computing 2012 Jun 24 (pp. 678-685).

[24] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, October. Detecting co-residency with active traffic analysis techniques. In Proceedings of the 2012 ACM Workshop on Cloud computing security workshop 2012 Oct 19 (pp. 1-12).

[25] S. Yu, X. Gui, J. Lin, X. Zhang, and J. Wang, "Detecting VMs co-residency in the cloud: Using cache-based side channel attacks," Elektronika ir Elektrotechnika. 2013;19 (5):73-8.

[26] S. Sundareswaran and A. C. Squcciarini, "Detecting malicious co-resident virtual machines indulging in load-based attacks," In International Conference on Information and Communications Security 2013 Nov 20 (pp. 113-124). Springer.

[27] S. Yu, X. Gui, and J. Lin, "An Approach with Two-Stage Mode to Detect Cache- based Side Channel Attacks," In The International Conference on

Information Networking 2013 (ICOIN) 2013 Jan 28 (pp. 186-191). IEEE.

[28] H. Idrissi, M. Ennahbaoui, E. M. Souidi, and S. El Hajji, "Mobile Agents with Cryptographic Traces for Intrusion Detection in the Cloud Computing," Procedia Computer Science. 2015 Jan 1; 73:179-86.

[29] Y. Zhang et al., "Incentive Compatible Moving Target Defense against VM-Colocation Attacks in Clouds," In IFIP international information security conference 2012 Jun 4 (pp. 388-399). Springer, Berlin, Heidelberg.

[30] Y. Azar and B. Shepherd, F.B.,"Co-Location-Resistant Clouds," In Proceedings of the 6th Edition of the ACM Workshop on Cloud Computing Security 2014 Nov 7 (pp. 9-20).

[31] D. Perez-botero, J. Szefer, and R. B. Lee, "Characterizing Hypervisor Vulnerabilities in Cloud Computing Servers," In Proceedings of the 2013 international workshop on Security in cloud computing 2013 May 8 (pp. 3-10).

[32] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-VM side channels and their use to extract private keys," In Proceedings of the 2012 ACM conference on Computer and communications security 2012 Oct 16 (pp. 305-316).

[33] J. Shi, X. Song, H. Chen, and B. Zang, "Limiting Cache-based Side-Channel in Multi-tenant Cloud using Dynamic Page Coloring," In2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W) 2011 Jun 27 (pp. 194-199).

[34] Z. Hao, Y. Tang, Y. Zhang, E. Novak, N. Carter, and Q. Li, "SMOC: A Secure Mobile Cloud Computing Platform," In 2015 IEEE Conference on Computer Communications (INFOCOM) 2015 Apr 26 (pp. 2668-2676).

[35] S. Singh, Y. S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," Journal of Network and Computer Applications. 2016 Nov 1; 75:200-22.

[36] F. Zafar et al., "A survey of cloud computing data integrity schemes : design challenges , taxonomy and future trends," Computers & Security. 2017 Mar 1; 65:29-49.

[37] N. Paladi, C. Gehrmann, and A. Michalas, "Providing User Security Guarantees in Public Infrastructure Clouds." IEEE Transactions on Cloud Computing. 2016 Feb 3; 5(3):405-19.

[38] H. Liang, C. Han, and D. Zhang, "A Lightweight Security Isolation Approach for Virtual Machines Deployment," In International Conference on Information Security and Cryptology 2014 Dec 13 (pp. 516-529). Springer.

[39] S. Jin, J. Ahn, J. Seol, S. Cha, J. Huh, and S. Maeng, "H-SVM: Hardware-Assisted Secure Virtual Machines under a Vulnerable Hypervisor," IEEE Transactions on Computers. 2015 Jan 9; 64(10):2833-46.

[40] S. Y. Vaezpour, R. Zhang, K. Wu, J. Wang, and G. C. Shoja, "Journal of Network and Computer Applications A new approach to mitigating security risks of phone clone co-location over mobile clouds," Journal of Network and Computer Applications. 2016 Feb 1; 62:171-84.

[41] S. Radicati, “Mobile Statistics Report, 2014-2018,” Mob. Stat. Report, 2014-2018, vol. 44, no. 0, pp. 1–4, 2014.

[42] D. Kovachev, Y. Cao, and R. Klamma, “Mobile Cloud Computing : A Comparison of Application Models,” Inf. Syst. J., vol. abs/1107.4, no. 4, pp. 14–23, 2011.

[43] P. Nawrocki and W. Reszelewski, “Resource usage optimization in Mobile Cloud Computing,” Computer Communications. 2017 Feb 1; 99:1-2.

[44] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: State-of-the-art and research challenges,” Journal of internet services and applications. 2010 May 1;1(1):7-18.

[45] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” Future generation computer systems. 2013 Jan 1;29 (1):84-106.

[46] “https://flexiscale.com/.” [Online]. Available: https://flexiscale.com/.

[47] “https://aws.amazon.com/fr/ec2/.” [Online]. Available: https://aws.amazon.com/fr/ec2/.

[48] “https://aws.amazon.com/fr/s3/.” .

[49] C. S. Partners, “Cloud Service and Deployment Models.” IEEE, Jan 2015.

[50] L. Savu, “Cloud Computing Deployment models , delivery models , risks and research challanges,” In 2011 International Conference on Computer and Management (CAMAN) 2011 May 19 (pp. 1-4). IEEE.

[51] K. Kumar and Y. Lu, “Cover feature cloud computing for mobile users: computation save energy?,” IEEE Computer Society, Issue. 2010 Apr (04):51-6.

[52] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, “A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing,” IEEE Communications surveys & tutorials. 2012 Nov 29;15 (3):1294-313.

[53] A. Ellouze, M. Gagnaire, and A. Haddad, “A Mobile Application Offloading Algorithm for Mobile Cloud Computing,” In 2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering 2015 Mar 30 (pp. 34-40).

[54] N. M. Dhanya and G. Kousalya, “Adaptive and Secure Application Partitioning for Offloading in Mobile Cloud Computing,” In International Symposium on Security in Computing and Communication 2015 Aug 10 (pp. 45-53). Springer.

[55] B. Chun and A. Patti, “CloneCloud : Elastic Execution between Mobile Device and Cloud,” In Proceedings of the sixth conference on Computer systems 2011 Apr 10 (pp. 301-314).

[56] J. Zheng, Y. Cai, S. Member, Y. Wu, S. Member, and X. S. Shen, “Dynamic Computation Offloading for Mobile Cloud Computing : A Stochastic Game-Theoretic Approach,” IEEE Transactions on Mobile Computing. 2018 Jun 14;18 (4):771-86.

[57] S. Guo, J. Liu, Y. Yang, B. Xiao, and S. Member, “Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing,” IEEE Transactions on Mobile Computing. 2018 Apr 30;18 (2):319-33.

[58] T. H. Noor, S. Zeadally, A. Alfazi, and Q. Z. Sheng, “ Mobile cloud computing : Challenges and future research directions,” Journal of Network and Computer Applications. 2018 Aug 1; 115:70-85.

[59] P. Dixit, A. K. Gupta, and M. C. Trivedi, “Traditional and Hybrid Encryption Techniques : A Survey,” In Networking Communication and Data Knowledge Engineering 2018 (pp. 239-248). Springer, Singapore.

[60] B. B. Gupta, S. Yamaguchi, and D. P. Agrawal, “Advances in Security and Privacy of Multimedia Big Data in Mobile and Cloud Computing,” Multimedia Tools and Applications. 2018 Apr 1;77 (7):9203-8.

[61] Y. Fan, X. Lin, G. Tan, Y. Zhang, and W. Dong, “One Secure Data Integrity Verification Scheme for Cloud Storage,” Future Generation Computer Systems. 2019 Jul 1; 96:376-85.

[62] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing : architecture , applications , and approaches,” Wireless communications and mobile computing. 2013 Dec 25; 13(18):1587-611.

[63] K. Benzekki, A. El Fergougui, and A. E. Elalaoui, “A Context-Aware Context-Aware Authentication Authentication System for Mobile Cloud Computing,” Procedia Computer Science. 2018 Jan 1; 127:379-87.

[64] S. Ashraf, C. I. Luk, K. Seungmin, M. Sabzinejad, and F. Taeshik, “An improved anonymous authentication scheme for distributed mobile cloud computing services,” Cluster Computing. 2019 Jan 16; 22(1):1595-609.

[65] N. Agrawal and S. Tapaswi, “A Trustworthy Agent-Based Encrypted Access Control Method for Mobile Cloud Computing Environment,” Pervasive and Mobile Computing. 2019 Jan 1; 52:13-28.

[66] V. Koe, A. Sandor, and Y. Lin, “Offline privacy preserving proxy re-encryption in mobile cloud computing,” Pervasive and Mobile Computing. 2019 Oct 1; 59:101081.

[67] P. Singh and K. Kaur, "Secure and Efficient Enhanced Sharing of Data Over Cloud Using Attribute Based Encryption with Hash Functions," In International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments 2018 Nov 28 (pp. 102-117). Springer.

[68] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, “Computation Offloading for Service Workflow in Mobile Cloud Computing,” IEEE transactions on parallel and distributed systems. 2014 Dec 18;26 (12):3317-29.

[69] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, “A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing,” ACM SIGMETRICS Performance Evaluation Review. 2013 Apr 29; 40(4):23-32.

[70] S. Y. Vaezpour, R. Zhang, K. Wu, J. Wang, and G. C. Shoja, “A new approach

to mitigating security risks of phone clone co-location over mobile clouds," Journal of Network and Computer Applications. 2016 Feb 1; 62:171-84.

[71] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: A survey, state of art and future directions," Mobile Networks and Applications. 2014 Apr 1;19 (2):133-43.

[72] T. B. A. K. Verma, "Data security in mobile cloud computing paradigm : a survey , taxonomy and open research issues," The Journal of Supercomputing. 2017 Jun 1;73 (6):2558-631.

[73] C. N. Modi and K. Acha, "Virtualization layer security challenges and intrusion detection/prevention systems in cloud computing: a comprehensive review," the Journal of Supercomputing. 2017 Mar 1; 73(3):1192-234.

[74] S. Luo, Z. Lin, X. Chen, Z. Yang, and J. Chen, "Virtualization security for cloud computing service," In 2011 International Conference on Cloud and Service Computing 2011 Dec 12 (pp. 174-179). IEEE.

[75] F. Hu et al., "A Review on Cloud Computing : Design Challenges in Architecture and Security," Journal of Computing and Information Technology. 2011 Mar 30;19 (1):25-55.

[76] M. M. Hassan, W. N. Ismail, and B. Song, "Mobile cloud-based big healthcare data processing in smart cities," IEEE Access. 2017 May 25; 5:11887-99.

[77] J. Sahoo, "Virtualization : A Survey On Concepts , Taxonomy And Associated Security Issues.", In 2010 Second International Conference on Computer and Network Technology 2010 Apr 23 (pp. 222-226). IEEE.

[78] R. Khan, M. Othman, S. A. Madani, and I. Member, "A Survey of Mobile Cloud Computing Application Models," IEEE communications surveys & tutorials. 2013 Jul 4; 16(1):393-413.

[79] J. Shao and Z. Cao, "CCA-Secure Proxy Re-Encryption without Pairings,", In International Workshop on Public Key Cryptography 2009 Mar 18 (pp. 357-376). Springer, Berlin, Heidelberg.

[80] R. D. Pietro B and F. Lombardi, "Virtualization Technologies and Cloud Security : Advantages , Issues , and Perspectives," In From Database to Cyber Security 2018 (pp. 166-185). Springer.

[81] L. Malhotra, D. Agarwal, and A. Jaiswal, "Virtualization in Cloud Computing," J. Inform. Tech. Softw. Eng. 2014 Jun; 4(2):1-3.

[82] N. Subramanian and A. Jeyaraj, "Recent security challenges in cloud computing ," Computers & Electrical Engineering. 2018 Oct 1; 71:28-42.

[83] Y. Han, J. Chan, and C. Leckie, "Virtual Machine Allocation Policies against Co-resident Attacks in Cloud," In 2014 IEEE International Conference on Communications (ICC) 2014 Jun 10 (pp. 786-792).

[84] Fox R, Hao W. Internet Infrastructure: Networking, Web Services, and Cloud Computing. CRC Press; 2017 Oct 20.

[85] Z. Qin, H. Xiong, S. Wu, and J. Batamuliza, "A Survey of Proxy Re-Encryption for Secure Data Sharing in Cloud Computing," IEEE Transactions on Services

Computing. 2016 Apr 6 vol. 13, no. 9.

[86] A. Luotonen and K. Altis, “World-Wide Web Proxies,” Computer Networks and ISDN systems. 1994 Nov 1;27 (2):147-54.

[87] V. Vijayakumar, M. K. Priyan, G. Ushadevi, R. Varatharajan, and G. Manogaran, “E-Health Cloud Security Using Timing Enabled Proxy Re-Encryption,” Mobile Networks and Applications. 2019 Jun 15; 24(3):1034-45.

[88] W. Liangchen and Y. U. Yan, “Proxy Re-Encryption Based Multi-Factor Access Control for Ciphertext in Cloud,” Journal of Shanghai Jiaotong University (Science). 2018 Oct 1;23 (5):666-70.

[89] W. Luo, “Secure and efficient proxy re-encryption scheme based on key-homomorphic constrained PRFs in cloud computing,” Cluster Computing. 2019 Jun 15; 22(2):541-51.

[90] S. Kim and I. Lee, “IoT device security based on proxy re-encryption,” Journal of Ambient Intelligence and Humanized Computing. 2018 Aug 1;9 (4):1267-73.

[91] H. Zhu, Y. Tan, L. Zhu, Q. Zhang, and Y. Li, “An Efficient Identity-Based Proxy Blind Signature for Semioffline Services,” Wireless Communications and Mobile Computing. 2018; vol.2018.

[92] J. Li, J. Li, X. Chen, C. Jia, W. Lou, and S. Member, “Identity-based Encryption with Outsourced Revocation in Cloud Computing,” IEEE Transactions on computers. 2013 Oct 21;64 (2):425-37.

[93] R. Arora and A. Parashar, “Secure User Data in Cloud Computing Using Encryption Algorithms,” International journal of engineering research and applications. 2013 Jul;3 (4):1922-6.

[94] S. K. Nair et al., “Towards Secure Cloud Bursting , Brokerage and Aggregation,” In 2010 eighth IEEE European conference on web services 2010 Dec 1 (pp. 189-196).

[95] J. Huang and D. M. Nicol, “Trust mechanisms for cloud computing,” Journal of Cloud Computing: Advances, Systems and Applications. 2013 Dec 1;2 (1):9.

[96] N. Kilari and R. Sridaran, “A Novel Approach to Protect Cloud Environments Against DDOS Attacks,” In Big Data Analytics 2018 (pp. 515-523). Springer, Singapore.

[97] A. Jasti, P. Shah, R. Nagaraj, and R. Pendse, “Security in Multi-Tenancy Cloud,” In 44th Annual 2010 IEEE International Carnahan Conference on Security Technology 2010 Oct 5 (pp. 35-41).

[98] P. Stewin and I. Bystrov, “LNCS 7591 - Understanding DMA Malware,” In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment 2012 Jul 26 (pp. 21-41). Springer, Berlin, Heidelberg.

[99] W. A. Jansen, “Cloud Hooks : Security and Privacy Issues in Cloud Computing,” In 2011 44th Hawaii International Conference on System Sciences 2011 Jan 4 (pp. 1-10).

[100] Z. Wu and H. Wang, “Whispers in the Hyper-space : High-speed Covert

Channel Attacks inside the Cloud.", IEEE/ACM Transactions on Networking. 2014 Feb 19; 23(2):603-15.

[101] S. Sparks, J. Butler, S. Walker, and R. The, "Raising the bar for rootkit detection". Black Hat Japan. 2005 Oct;11(63):504-33.

[102] Levine JF, Grizzard JB, Owen HL. "Detecting and categorizing kernel-level rootkits to aid future detection," IEEE Security & Privacy. 2006 Feb 6;4(1):24-32.

[103] R. Roemer, E. Buchanan, H. Shacham, and S. Savage, "Return-Oriented Programming: Systems, Languages, and Applications," ACM Transactions on Information and System Security (TISSEC). 2012 Mar 1; 15(1):1-34.

[104] A. One, "Smashing the stack for fun and profit". Phrack magazine. 1996 Nov 8;7(49):14-6.

[105] Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting, "An exploration of L2 cache covert channels in virtualized environments," In Proceedings of the 3rd ACM workshop on Cloud computing security workshop 2011 Oct 21 (pp. 29-40).

[106] G. Drosatos, P. S. Efraimidis, I. N. Athanasiadis, E. D. Hondt, and M. Stevens, "A privacy-preserving cloud computing system for creating participatory noise maps," In 2012 IEEE 36th Annual Computer Software and Applications Conference 2012 Jul 16 (pp. 581-586).

[107] Q. Chai and G. Gong, "Verifiable Symmetric Searchable Encryption For Semi-honest-but-curious Cloud Servers," In 2012 IEEE International Conference on Communications (ICC) 2012 Jun 10 (pp. 917-922).

[108] P. Zhang, M. Zhou, and Y. Kong, "A Double-Blind Anonymous Evaluation-Based Trust Model in Cloud Computing Environments," IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2019 Apr 4.

[109] A. Hammoud, H. Otrok, A. Mourad, O. A. Wahab, and J. Bentahar, "On the Detection of Passive Malicious Providers in Cloud Federations," IEEE Communications Letters. 2018 Oct 30; 23 (1):64-7.

[110] A. Wailly and M. Lacoste, "VESPA : Multi-Layered Self-Protection for Cloud Resources," In Proceedings of the 9th international conference on Autonomic computing 2012 Sep 18 (pp. 155-160).

[111] N. Gruschka and G. Horst, "Attack Surfaces : A Taxonomy for Attacks on Cloud Services," In 2010 IEEE 3rd international conference on cloud computing 2010 Jul 5 (pp. 276-279).

[112] J. Szefer, E. Keller, R. B. Lee, and J. Rexford, "Eliminating the Hypervisor Attack Surface for a More Secure Cloud Categories and Subject Descriptors," In Proceedings of the 18th ACM conference on Computer and communications security 2011 Oct 17 (pp. 401-412).

[113] S. Iqbal, L. M. Kiah, M. Hussain, M. K. Khan, and K. Raymond, "On cloud security attacks: A taxonomy and intrusion detection and prevention as a service," Journal of Network and Computer Applications. 2016 Oct 1; 74:98-120.

[114] T. Islam and D. Manivannan, “A Classification and Characterization of Security Threats in Cloud Computing,” Int. J. Next-Gener. Comput. 2016 Mar 1;7(1).

[115] 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. 2013 Dec 1;4 (1):5.

[116] R. Patil, H. Dudeja, and C. Modi, “Designing an efficient security framework for detecting intrusions in virtual network of cloud computing,” Computers & Security. 2019 Aug 1; 85:402-22.

[117] H. Liang, C. Han, D. Zhang, and D. Wu, “A lightweight security isolation approach for virtual machines deployment,” In International Conference on Information Security and Cryptology 2014 Dec 13 (pp. 516-529). Springer.

[118] M. Li, Y. Zhang, K. Bai, W. Zang, M. Yu, and X. He, “Improving Cloud Survivability through Dependency based Virtual Machine Placement.”, In SECRYPT 2012 Jul (pp. 321-326).

[119] L. He et al., “Dynamic Secure Interconnection for Security Enhancement in Cloud Computing,” International Journal of Computers Communications & Control. 2016 Mar 24;11 (3):348-57.

[120] T. Zhang, Y. Zhang, and R. B. Lee, “CloudRadar : A Real-Time Side-Channel Attack Detection System in Clouds,”, In International Symposium on Research in Attacks, Intrusions, and Defenses 2016 Sep 19 (pp. 118-140). Springer.

[121] M. Azab and M. Eltoweissy, “MIGRATE : Towards a Lightweight Moving-target Defense against Cloud Side- Channels,” In 2016 IEEE security and privacy workshops (SPW) 2016 May 22 (pp. 96-103).

[122] R. B. Gomes, R. D. Medina, and F. G. Moro, “Cloud Aid – A Cloud Computing Tool for Mitigating Side-Channel Attacks,” In NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium 2018 Apr 23 (pp. 1-5).

[123] S. Jin, J. Seol, J. Huh, and S. Maeng, “Hardware-Assisted Secure Resource Accounting under a Vulnerable Hypervisor,” ACM SIGPLAN Notices. 2015 Mar 14; 50(7):201-13.

[124] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Software: Practice and experience. 2011 Jan; 41(1):23-50.

[125] F. Howell and R. Mcnab, : “A discrete event simulation library for java”. Simulation Series. 1998 Jan 11;30:51-6.

[126] R. Buyya, R. Ranjan, and R. N. Calheiros, “Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit : Challenges and Opportunities,” In 2009 international conference on high performance computing & simulation 2009 Jun 21 (pp. 1-11). IEEE.

[127] D. Kliazovich and P. Bouvry, “GreenCloud : a packet-level simulator of energy-aware cloud computing data centers,” The Journal of Supercomputing. 2012 Dec 1; 62(3):1263-83.

[128] A. N. J. L. Vázquez-poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente, “iCanCloud : A Flexible and Scalable Cloud Infrastructure Simulator,” Journal of Grid Computing. 2012 Mar 1; 10(1):185-209.

[129] S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer, “GroudSim : An Event-Based Simulation Framework for Computational Grids and Clouds,” In European Conference on Parallel Processing 2010 Aug 31 (pp. 305-313). Springer, Berlin, Heidelberg.

[130] S. K. Garg and R. Buyya, “NetworkCloudSim : Modelling Parallel Applications in Cloud Simulations,” , In 2011 Fourth IEEE International Conference on Utility and Cloud Computing 2011 Dec 5 (pp. 105-113). IEEE.

[131] U. U. Rehman, A. Ali, and Z. Anwar, “secCloudSim : Secure Cloud Simulator.”, In 2014 12th International Conference on Frontiers of Information Technology 2014 Dec 17 (pp. 208-213). IEEE.

[132] U. M. Maurer and S. Wolf, “The Diffie – Hellman Protocol,” Designs, Codes and Cryptography. 2000 Mar 1; 19(2-3):147-71.

[133] A. Pugazhenthi and D. Chitra, “Data Access Control and Secured Data Sharing Approach for Health Care Data in Cloud Environment,” Journal of medical systems. 2019 Aug 1;43(8):258.

[134] G. Orsini, D. Bade, and W. Lamersdorf, “Generic Context Adaptation for Mobile Cloud Computing Environments,” Journal of Ambient Intelligence and Humanized Computing. 2018 Feb 1;9 (1):61-71.

[135] M. E. Taylor and D. Aboagye-darko, “Security Approaches and Crypto Algorithms in Mobile Cloud Storage Environment to Ensure Data Security,” In International Conference on Artificial Intelligence and Security 2019 Jul 26 (pp. 516-524). Springer.

[136] T. Bhatia and G. Sharma, “Towards a secure incremental proxy re-encryption for e-healthcare data sharing in mobile cloud computing,” Concurrency and Computation: Practice and Experience. 2019 Sep 15:e5520.

[137] N. Doukas, O. P. Markovskyi, and N. G. Bardis, “Hash function design for cloud storage data auditing,” Theoretical Computer Science. 2019 Dec 31; 800:42-51.

[138] Y. Yang, F. Chen, J. Chen, Y. Zhang, and K. L. Yung, “A secure hash function based on feedback iterative structure,” Enterprise Information Systems, vol. 00, no. 00, pp. 1–22, 2019.

[139] Y. Yang et al., “Secure and efficient parallel hash function construction and its application on cloud audit,” Soft Computing. 2019 Sep 1; 23(18):8907-25.

[140] J. Park and J. H. Park, “applied sciences A Lightweight Hash-Based Blockchain Architecture for Industrial IoT,” Applied Sciences. 2019 Jan;9 (18):3740.

[141] P. Offermann and E. R. Platz, “Outline of a Design Science Research Process,” In Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology 2009 May 7 (pp. 1-11).

[142] Y. S. Sherif, “The Design , Analysis , And Evaluation Of Simulation Experiments,” International Journal of Modelling and Simulation. 1998, 18:4, 290-297.

[143] T. L. Paez, “Introduction to Model Validation”, Sandia National Lab. (SNL-NM), Albuquerque, NM (United States); 2008 Nov 1.

[144] J.-Y. Le Boudec, “Performance evaluation of computer and communication systems”: EPFL Press, 2011 Feb 1.

[145] A. Doosti and A. M. Ashtiani, “Mathematical Modeling : a new approach for mathematics teaching in different levels”, Islamic Azad University. 2005.

[146] D. Z. S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, “Improving Simulation For Network Research,” Technical Report 99-702b, University of Southern California; 1999 Mar 4.

[147] R. Campbell et al., “Open Cirrus TM Cloud Computing Testbed : Federated Data Centers for Open Source Systems and Services Research,” , HotCloud. 2009 Jun 15;9:1.

[148] E. Weing, H. Lehn, and K. Wehrle, “A performance comparison of recent network simulators,”, In 2009 IEEE International Conference on Communications 2009 Jun 14 (pp. 1-5).

[149] J. Chan and C. Leckie, “Security Games for Virtual Machine Allocation in Cloud Computing,”, In International Conference on Decision and Game Theory for Security 2013 Nov 11 (pp. 99-118). Springer.

[150] F. Miao, L. Wang, and Z. Wu, “A VM Placement Based Approach to Proactively Mitigate Co-Resident Attacks in Cloud,” In 2018 IEEE Symposium on Computers and Communications (ISCC) 2018 Jun 25 (pp. 00285-00291).

[151] F. Zhou and P. Desnoyers, “Scheduler Vulnerabilities and Coordinated Attacks in Cloud Computing,” Journal of Computer Security. 2013 Jan 1; 21(4):533-59.

[152] B. Gulmezoglu, T. Eisenbarth, and B. Sunar, “Co-location Detection on the Cloud,”, In International Workshop on Constructive Side-Channel Analysis and Secure Design 2016 Apr 14 (pp. 19-34). Springer.

[153] A. Agarwal, “Co-Location Resistant Virtual Machine Placement in Cloud Data Centers,” In 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS) 2018 Dec 11 (pp. 61-68).

[154] R. Guesmi, M. A. B. Farah, A. Kachouri, and M. Samet, “A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2,” Nonlinear Dynamics. 2016 Feb 1; 83(3):1123-36.

[155] U. Villano, “A Proposal of a Cloud-Oriented Security as-a-Service,”, In Conference on Complex, Intelligent, and Software Intensive Systems 2018 Jul 4 (pp. 1002-1011). Springer.

[156] P. Garg, “An Efficient and Secure Data Storage in Mobile Cloud Computing through RSA and Hash Function,”, In International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) 2014 Feb 7 (pp. 334-339). IEEE.

[157] N. M. M. Abdelnapi, “A Hybrid Hashing Security Algorithm for Data Storage on Cloud Computing,”, International Journal of Computer Science and Information Security. 2016 Apr 1; 14 (4):175.

[158] M. Vaezi and Y. Zhang, “Cloud Mobile Networks,” Springer; 2017.

[159] B. Annane and O. Ghazali, “Virtualization-Based Security Techniques on Mobile Cloud Computing : Research Gaps and Challenges,” In International Journal of Interactive Mobile Technologies, 2019, 13, No. 4, (pp. 20–32).