SecNetworkCloudSim: An Extensible Simulation Tool for Secure Distributed Mobile Applications

Boubakeur Annane[1], Adel Alti[2,3], Osman Ghazali[1]

[1]School of Computing, University Utara Malaysia, Sintok, Kedah, Malaysia
[2]Department of Management Information Systems, College of Business & Economics Qassim University, Buraidah, KSA
[3]Department of Computer Science , Faculty of Sciences , Ferhat Abbas Setif-1 University , Algeria.

***Abstract*:** Fueled by the wide interest for achieving rich-storage services with the lowest possible cost, cloud computing has emerged into a highly desired service paradigm extending well beyond Virtualization technology. The next generation of mobile cloud services is now manipulated more and more sensitive data on VM-based distributed applications. Therefore, the need to secure sensitive data over mobile cloud computing is more evident than ever. However, 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 paper introduces a new NetworkCloudSim extension called SecNetworkCloudSim, a secure mobile simulation tool that is deliberately designed to ensure the preservation of confidential access to data hosted on the mobile device and distributed cloud's servers. Through high-level mobile users' requests, users connect to an underlying proxy, which is considered an important layer in this new simulator, where users perform secure authentication access to cloud services, allocate their tasks in secure VM-based policy, manage automatically the data confidentiality among VMs and derive high efficiency and coverage rates. Most importantly, due to the secure nature of proxy, user's distributed tasks can be executed without alterations on different underlying proxy's security policies. We implement a scenario of follow-up healthcare distributed application using the new extension

***Keywords*:** Virtualization, Data security and privacy, Cloud simulation tools, Mobile Cloud.

1. Introduction

Cloud computing attracted wide attention as the technology able to delivering IT services and resources on-demand over the Internet. It effectively reduces the deployment cost and increases the scalability by using cloud services and hardware resources with no specific time and requirements. Due to the huge capacity of cloud data-centers and servers, mobile users use powerful cloud resources like CPU, Memory and storage capacity to handle their applications and the results back to mobile devices [1, 2]. Today, cloud technologies are penetrating into different mobile applications domains, overcome the limitations of mobile devices and bring the next generation of technology named Mobile Cloud Computing (MCC).

Based on the advances of virtualization, the cloud-computing infrastructure can efficiently be managed to highly exploit by many users. The virtualization is considered as one of the important components in cloud computing. It augments the utilization rate of using the computing resources of the cloud [3, 4]. Mobile users offload their critical applications to Virtual Machines (VMs) hosted on multiple servers. Such VMs may contain many intensive tasks of the same application that exchange sensitive data. Many studies [5-11; 23-24] showed that users' sensitive data can be stolen and altered while exchanged between distributed VMs or even processed within the VMs. Therefore, it is important to prevent critical threats that affect shared private data hosted on both mobile users' devices and cloud service providers.

Due to the need for a well understanding of the cloud paradigm technology capabilities, a number of cloud simulators are available today such as CloudSim [12], Green Cloud [13], iCanCloud [14], GroudSim [15], NetworkCloudSim [16], secCloudSim [17]. 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 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 are offering cloud security features to control access data [18]. This requires knowledge of security algorithms details, which is not all users, have and requires spending significant amounts of time implementing 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. The main contributions of this paper are:

1. To provide conceptual model for preserving the confidential access to shared data over distributed cloud's servers. The model includes crypto-hashing mechanisms, supporting secure VMs-based allocation and communication by using proxy-based access control.
2. To provide SecNetworkCloudSim, a secure, mobile and open-source simulation tool for accessing and managing VMs of the distributed mobile application over cloud platforms by protecting the VM's interaction with new communication policy at runtime. The SecNetworkCloudSim extends a NetworkCloudSim simulator with a new security layer comprised of (i) a Mobile Users Control Access component that can be used for restraining the data access only to authorized users that having main security requirements, (ii) a secure Hypervisor component for protecting the VMs and allocating them in secure servers, and (iii) and a secure VM's communication component for protecting the

communicating tasks that deployed in various servers.

3. To Illustrate of wide applicability of SecNetworkCloudSim, a distributed healthcare mobile application is introduced over the network mobile Cloud platform that supports the security of data and VMs communication management.

The rest of the paper is organized as follows: Section 2 presents some of the related simulation tools. Section 3 introduces SecNetworkCloudSim. Section 4 presents an evaluation of our proposed simulator. Finally, Section 5 concludes this paper.

2. Related Work

From an analysis perspective, CloudSim is a well-known simulator toolkit that provides modeling and simulation of the cloud environment and its resources [12]. CloudSim is widely adopted by researchers and organizations such as HP labs in the USA. CloudSim is able of simulating the variety component such as data centers, hosts, users, virtual machines, service brokers and cloudlets, 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 types of research work including the development of new policies, editing the existing one, implementing mechanisms and approach and make several tests under different scenarios before trying on the real systems. However, the major disadvantage of this tool is the absence of graphical panel. Figure 1 presents an overview of CloudSim architecture.

Datacenters are considered as the main components that provisioning the computing resources for tenants in the 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 [13] is destined for such issue of energy awareness of datacenters. The disadvantage of the Green Cloud is its TCP/IP implementation in the data center network, which requires high memory requirements and large simulation time. Green cloud scales well when the simulation overhead is low. Figure 2 presents Green Cloud architecture.

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 permits to develop and evaluate the performance of any proposed approach in a 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 [14]. The simulator is concentrated on the Amazon cloud provider to do experiments on such a 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 3 shows the general architecture of iCanCloud.

Due to the process-based approach, which executes each separate thread in host machine and the lack of 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 [15]. 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.

NetworkCloudSim [16] extends CloudSim to model distributed application with the communication aspect while the datacenters' resources process the tasks. Mainly, the NetworkCloudSim considers all tasks as first-class entities called network cloudlets to integrate both computation and communication. Furthermore, in NetworkCloudSim, a VM is connected to other VMs by various tools: root, aggregate, and edge, which make up a real network model inside the datacenter. NetworkCloudSim allows researchers to model large-scale distributed application such as message passing applications that requires tasks communication and sharing data between each other. This tool allows modeling and simulating various network on the cloud environment. Figure 4 shows the general architecture of NetworkCloudSim.

Therefore, 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 [17]. 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 basic 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 5 shows the general architecture of secCloudSim.

All the above simulators for both cloud computing and MCC attracted researchers to utilize their potential capabilities in different research issues such as load balancing, power consumption, offloading, security and privacy issues. Table 1 shows a comparison between the most popular cloud simulators selecting the right simulator tool is very necessary to evaluate the research project. From the table above, current simulators are cost-free environments aimed to model and simulate cloud-computing environments whereas they still not fully meet the expectation of researchers. Specifically, current they present notable limitations: (i) they are not able to model and simulate security-aware VM allocation policies and (ii) they did not provide secure communication aspects between VM-based tasks of complex distributed mobile applications. Thus, these limitations can be overcome by using SecNetworkCloudSim.

3. 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 various network data centers. The main goal of this work 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 in NetworkCloudSim various security layers. NetworkCloudSim is considered one of the most powerful toolkits that give the ability to codify the behavior of very complex cloud system. 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 a new layer called Cloud Three Security Proxy (Proxy-3S) with three security policies in order to provide a flexible simulation of 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.

3.1 Main Functional building modules

Based on the principles specified in the architecture model of NetworkCloudSim, Figure 6 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.
2. Mobile User Access Control Policy module. The simulator supports the integration of user's control access that receives, 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 the available list of hash codes.

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. Distributed Application Configuration module provides a 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 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 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 is the core-building module that enables security insights based on early detection of VMs attackers using advanced communication policies. This latter use 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 machine communication interacts with a secure networked data center module.
5. The 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 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 consists of many tasks collaborating with each other insecure 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.

3.2 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.

3.2.1 Design of SecNetworkCloudSim

Figure 7 presents a generic overview of the class diagram of SecNetworkCloudSim. 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 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. The provided task is defined as the capacity needed for 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 enables to identify the VMs whether the VM is legal or attacker. The Secure Cloud Proxy removed the attacker from the host. Different security policies allow ensuring the data protection and privacy of legal VM, as detailed as follow:

- Mobile User Control Access 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 to cloud host and deploying its VMs, he sends an encrypted signature. The mobile user control access decrypts it and tries authenticating with all of authentication signatures list.
- VMM Protection class is useful 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 class 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 VMs Blacklist list.
- VMM Allocation Policy is used to allow the hypervisor allocating 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. 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 the 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 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 contains the list of unauthorized mobile users.
- VMM key check contains the session identifiers of all VMs hosted on the Cloud. Each VM executes in particular session time. For each session, the secure proxy-based cloud manages specific VMs and its secret keys.
- Network Host describes all servers located in a 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

3.2.2 General Architecture of Distributed Mobile VM-based Mobile Application

The aim of the proposed approach is to handle the security of the distributed mobile application and to achieve high-security service for mobile users, which makes sensitive data more secure from any unauthorized VMs on the cloud. The main contribution of this work is a new secure cloud proxy called Proxy-3S, which is a middleware between the client-side and the cloud side that controls the unauthorized access of users and malicious VMs.

We handled the user's application that comprises of many resources--intensive tasks. Figure 8 shows the steps involved in dealing with the user's request through proxy-3S. After sending the request for offloading the tasks to the cloud resources. The proxy 3S receives the request and starts the authentication checking. The first step is ensured by the Mobile User Access Control component, if the mobile user's authentication is verified then the second component Secure Allocation VMs intervenes else the access is denied. The Secure Allocation VMs component sends the order to the hypervisor to deploy the requested intensive task on the available VMs. The hypervisor allocates the VMs safely and secures the data inside from being stolen. Further, it provides secure isolation between different VMs while sharing the same resource. Therefore, it applied the security allocation policies (i.e. secure least VM policy, secure most VM policy, secure random VM policy), if the allocation operation succeeds then it is shifted to the third step, else a denied allocation is sent. The secure VMs communication component is responsible for the security of exchanged sensitive data among VMs and its status. It starts by checking the identity of target VM if the VM exists in the VMs blacklist, the proxy-3S declares the possibility of threat, the system actively denies the communication risk and notifies the other VMs. If VM is legal, its status must be checked. Here, if the status is Green than the data is shared and the access is authorized.

3.2.3 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 9.

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* refer to iteration and *Ts* is simulation time.

Once the simulation scenario starts to run on SecNetworkCloudSim simulator, the proxy-3S receives the new MAC address of the 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 the security performances of the scenario such as efficiency and coverage of the application

3.2.4 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 has several threats, which affect the application. An attacker can deploys malicious VMs in either the same or the different server. Authors in [19, 20] studied experimental deployments of VMs in different scenarios using different VMs 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 [19, 20], two metrics have proposed for detecting the attack, namely efficiency and coverage. We used these two metrics and improved in our approach as follows. (Table 2 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 [19, 20] 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:

$$E(\,|\,RC(A,t)\,|\,) = \frac{|ST\_VM(A,t)| - ND}{|VM\ (A,t)| + |VM\_in(A,t)|} \quad (1)$$

Where:
RC: remote co-located VMs.
ST_VM: VMs attackers which succeed to co-locate with the target.
ND: new detected attacker.
VM_in: VMs attacker interactions

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 with each other while deployed in different hosts on the cloud, the coverage metric of [19, 20] 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:

$$C(\,|\,RC(A,t)\,|\,) = \frac{|ST\_VM(A,t)| - ND}{|VM\ (L,t)| + |VM\_in(L,t)|} \quad (2)$$

Where:
RC: remote co-located VMs.
ST_VM: VMs attackers which succeed to co-locate with the target.
ND: new detected attacker.
VM_in: VMs attacker interactions

4. Case Study and Experimentation Results

We validate the extended simulator using distributed health care intensive mobile application that runs on both mobile devices and cloud environment. The protection of the users' data is ensured via extended security layers on the popular cloud simulator: NetworkCloudSim [16]. The effectiveness of this secure tool is demonstrated against different common attacks based on two security metrics named coverage and efficiency. The next sub-section illustrates the details experiments in order to observe the efficiency of SecNetworkCloudSim

4.1. e-health distributed mobile application

Here is an illustration of how the security protection is made through the main secure module of SecNetworkCloudSim named "Proxy-3S" for elderly patients using e-health distributed Healthcare system (Figure 10).

In real-world e-health distributed the mobile application, the software that controls the unauthorized access of users and malicious VMs in the cloud is required to run at a very high speed as well as being able to handle complex computations scenarios. The proposed simulator tool takes care of evolving different scenarios and involves the following three steps:

1. Applying the mobile user's access control to unauthorized user's access,
2. Considering the increased offloaded tasks and their secure allocation within current VMs
3. Providing new mechanisms for the detection of the remote VM's attacks. In the end, it calls the proxy-3S that considers the responsibility of it to protect the sensitive data of patients from unauthorized access and malicious VMs in the cloud.

As shown in Figure 10, the e-health distributed mobile application is built with three mobile devices (smartphone, healthcare device, tablet) and four fundamental users (patients, Proxy-3S, professional medical staff, doctor) distributed on three different locations (patients house, cloud, hospital). There are various tasks deployed in the cloud and perform the patients' health data (e.g. VM-Blood task, VM-temperature task, VM-weight task, and VM-analyzer task). This case study raises the problem of patient's private data alterations during transmitting it over the cloud or on the Cloud (e.g. a scenario when sensitive health information transmitted from the mobile device to the cloud). In addition, we assume that attackers can play the role of the authorized users, they cloud obtain patients' health data from the cloud.

Given our current simulator using Proxy-3S, we can afford to simulate the reception of the medical data of patients without unauthorized access of users. To simulate a proxy-3S detecting an attacker, the Poisson event-based simulation model is used to generate several randomized VMs attackers as shown in Figure 11 and deploy it automatically either in the same host as legal VM or on another host. However, we have three security policies running collaboratively to catch malicious attacks of the VMs in each simulation step. The simulator applies the mobile user control policy against unauthorized access of users. The detection is a simple matching between the provided hashed encrypted password and the authorized signatures list. The proxy, in turn, called the secure VM allocation to deploy tasks in safe way. Finally, the exchanged data between VMs is controlled through VMs secure communication policy.

4.2 Experimentation results

We have implemented SecNetworkCloudSim in Eclipse Java Development Tool (JDT) that extends NetworkCloudSim. NetworkCloudSim is a very good compromise between execution speed and ease of development and simulation. The extended simulator consists of different classes. These classes implement different behavior of the VMs and different security policies. The main objective is to optimize the efficiency and coverage of remote and co-resident attacks and reinforce the security of distributed mobile applications on the cloud. The experiments are performed on a Laptop 3.4 GHz, 4Go RAM using Windows 7 (64 bit) system. The simulator is used to control the security of different allocated mobile tasks on the Cloud through several scenarios including three security policies. The goal is to derive performance insights over exchanged sensitive data between distributed VMs. For evaluation, we use five configurations. Table 3 details each system configuration.

Table 3. Experimental configurations

Config	Servers	Edge	VMs	CPU (Pes)	Storage (GB)	BW (GB)	RAM (MB)
1	150	1	300	8	1000000	10000	2048
2	300	1	600	8	1000000	10000	2048
3	450	1	900	8	1000000	10000	2048
4	600	1	1200	8	1000000	10000	2048
5	750	1	1500	8	1000000	10000	2048

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 communicates 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 11. 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 considered 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 with 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.

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 simulator and its policies' performance to the related work [19,20]. The comparison has been done by taking 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 variables (perhaps shared memory and time). VM 1 updates the shared variable by modifying the state of its task and VM 2 reads the shared variable.

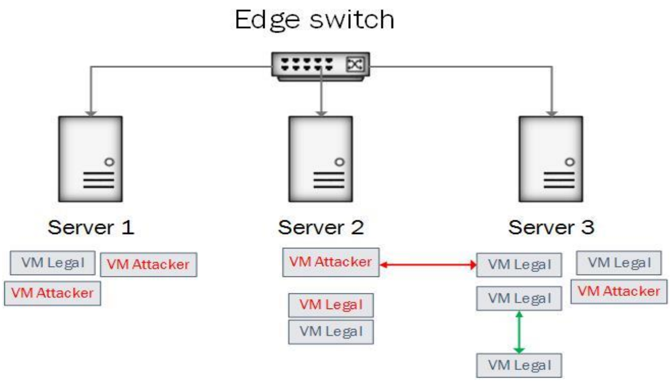

Figure 11. Intra and inter-communication of VMs of the co-located and remote VMs.

Comparison results for 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 3. As shown in Table 4 we observed that the obtained results of the proposed approach are much better than related work [19, 20]. 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 three 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 other sides, 22 legal communications have launched by legal VMs whether successful or unsuccessful communication.

After the updates that the proxy receives, the number of VMs attackers 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 attackers and the total number of all types 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 the total number of all types 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. A similar work adopts only co-residency for identifying VM's attacks.

Comparison results for medium-size configuration set VMs

We evaluated the effectiveness of Proxy-3S with a moderate size of VMs ranged from 350 to 600 in terms of efficiency and coverage. Table 5 shows the obtained results for the moderate size configuration set VMs (600). In 600 VMs, coverage and efficiency of our approach are quite better than the related work [19, 20]. The number of VMs attackers detected by the proxy is equal to 340 and the number of VMs legal is 260. From 340 VMs attackers, 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.

Comparison results for the large size configuration set VMs

In large-size VMs set configuration, we have also studied how a 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 [19, 20]. Table 6 shows 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 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 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 remotes 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.

4.3 Performances and Security Comparison Details

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

4.3.1 Comparison of execution time

We have evaluated the processing execution time with different cloud configurations after integrating security layers and compared to NetworkCloudSim. As shown in Figure 12, we observe that the processing execution time of SecNetworkCloudSim tool is very satisfactory ranging from 44142ms to 467572ms compared to NetworkCloudSim in five different cloud configurations. There is a slight difference in terms 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 the 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 spend 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.

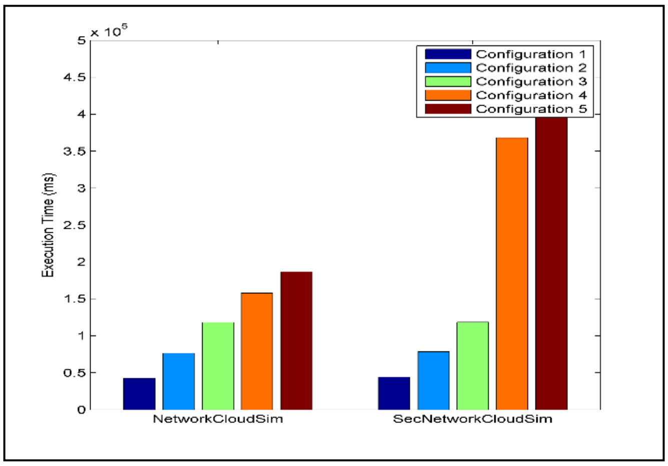

Figure 12. Comparison of execution times of SecNetworkCloudSim and NetworkCloudSim.

4.3.2 Security comparison of mobile user access control mechanism

Table 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 [19, 21] in terms of security checking time of public-private generation keys and encrypted mobile device's MAC address verification. The proposed mobile user access control is more effective and practicable than proposed related techniques. The reason is that our approach is a trust-based VM-communication model compared to work presented in [19] which are not considered. In addition, in the proposed mobile user access control, the mobile device's MAC (password) and data request is encrypted. This offers more security contrarily to work presented in [21] 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 [19, 21]. The checking security time refers to the time needed to generate RSA or Diffie-Hellman keys and time to hash a message.

Table 7. Features comparison between the proposed work and [19, 21].

Approaches	Data Security of Mobile User	Average security checking time (ms)
Co-residency [19]	No	-
Hash RSA-1024 [21]	Yes	257.2
Proposed MUAC	Yes	**75.4**

4.3.3 Drawbacks and comparison of security degree

Table 8 present advantages and drawbacks between the proposed method and related works described in [19, 21]. As we can see from tables 4, 5 and 6 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.

5. Conclusions

In this paper, we have presented the design and the implementation of new extended secure network Cloud simulator for the early detection of remote and co-resident attacks on Mobile Cloud computing. Remote and co-resident attacks cause huge and constantly increasing virtual machine information violation and damage, as well as unauthorized access of malevolent users. We add a new secure Cloud Proxy-3S that includes three security policies: VMs user's access control, VMs secure allocation and VMs secure communication for reducing these attacks by explicit consideration of the communication aspect between VMs. We illustrate the proposed extension through a scenario describing the e-health distributed mobile application using the new extension of the proposed simulator tool. The proposed simulator has been evaluated and proven in different cloud scenarios with the additional benefit of protecting exchanged sensitive information between virtual machines and decreasing efficiency attacks. Finally, we suggest integrating Blockchain for efficiently allowing users to secure their data in the distributed mobile application.

Acknowledgement

This research is funded by FRGS Research Grant Project no.14174. The authors would like to thank Universiti Utara Malaysia and Ministry of Higher Education for supporting this research

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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] B. Annane, O. Ghazali, and A. Alti, “A new secure proxy-based distributed virtual machines management in mobile cloud computing,” vol. 9, no. 43, 2019.

[11] B. Annane and O. Ghazali, “Virtualization-Based Security Techniques on Mobile Cloud Computing : Research Gaps and Challenges,” pp. 20–32, 2019.

[12] 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,” *Softw. - Pract. Exp.*, 2011.

[13] D. Kliazovich and P. Bouvry, “GreenCloud : a packet-level simulator of energy-aware cloud computing data centers,” pp. 1263–1283, 2012.

[14] 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,” pp. 185–209, 2012.

[15] S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer, “GroudSim : An Event-Based Simulation Framework for Computational Grids and Clouds,” no. 261585, pp. 305–313, 2011.

[16] S. K. Garg and R. Buyya, “NetworkCloudSim : Modelling Parallel Applications in Cloud Simulations,” no. Vm, 2011.

[17] Rehman, U. U., Ali, A., & Anwar, Z, “seccloudsim: Secure cloud simulator”. In 2014, 12th International Conference on Frontiers of Information Technology (pp. 208-213). IEEE.

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

[19] Y. Han, J. Chan, T. Alpcan, and C. Leckie, “Using Virtual Machine Allocation Policies to Defend against Co-resident Attacks in Cloud Computing,” vol. 5971, no. c, pp. 1–14, 2015.

[20] Y. Han, J. Chan, and C. Leckie, “Virtual Machine Allocation Policies against Co-resident Attacks in Cloud,” no. 1, pp. 786–792, 2014.

[21] P. Garg, “An Efficient and Secure Data Storage in Mobile Cloud Computing through RSA and Hash Function,” pp. 334–339, 2014.

[22] N. M. M. Abdelnapi, “A Hybrid Hashing Security Algorithm for Data Storage on Cloud Computing,” vol. 14, no. 4, pp. 175–181, 2016.

[23] M. E. Hussain, M. Qayyum, M. R. Hussain, and R. Hussain, Effective and Secure vWSN Applications in a Virtualized Cloud Computing Environment,’ Vol. 11, No. 1, 256 – 261, Aguest 2019, International Journal of Communication Networks and Information Security (IJCNIS).

[24] A. Y. Hendi, M. O. Dwairi, Z. A. Al-Qadi, M. S. Soliman. A novel simple and highly secure method for data encryption-decryption. 11(1), 232-238. Aguest 2019. International Journal of Communication Networks and Information Security (IJCNIS).

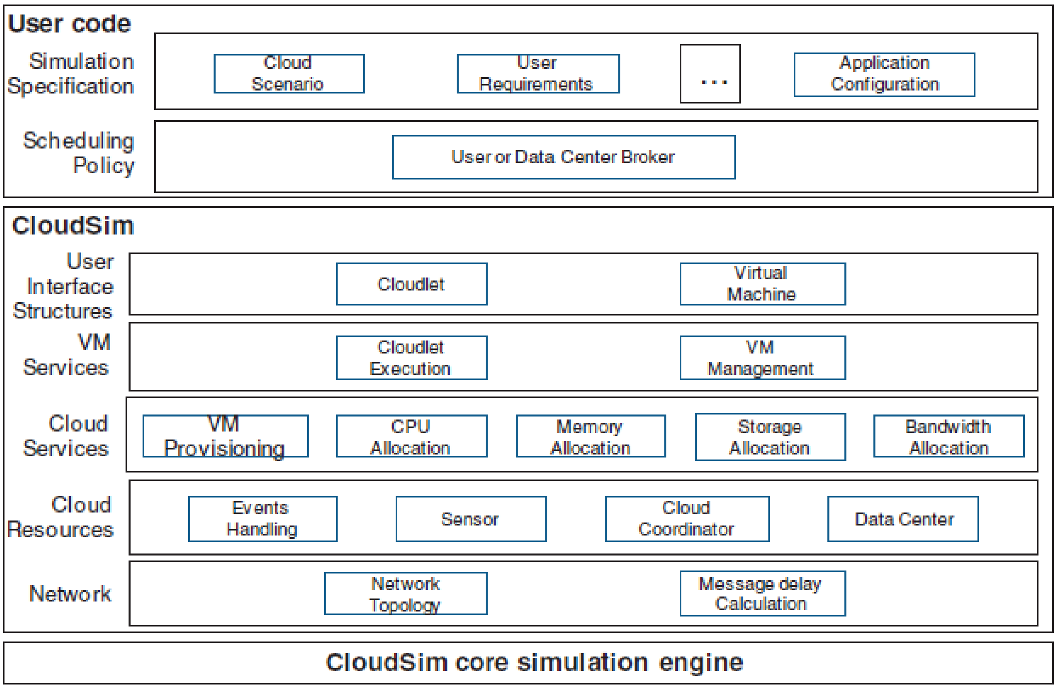

Figure 1. CloudSim architecture.

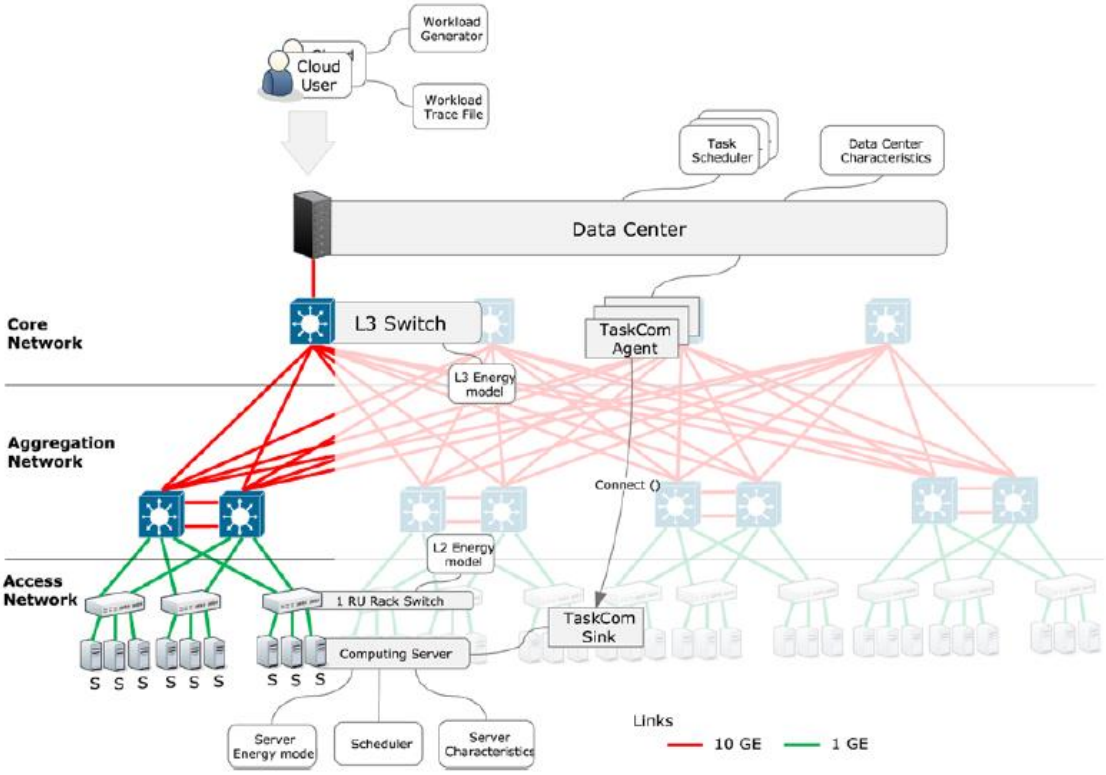

Figure 2. Green Cloud architecture.

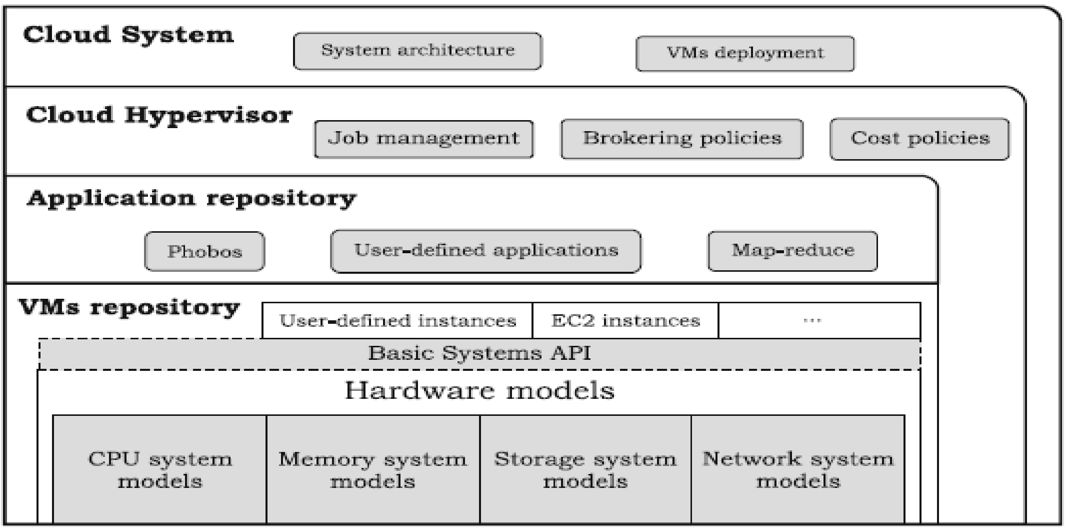

Figure 3. ICanCloud architecture.

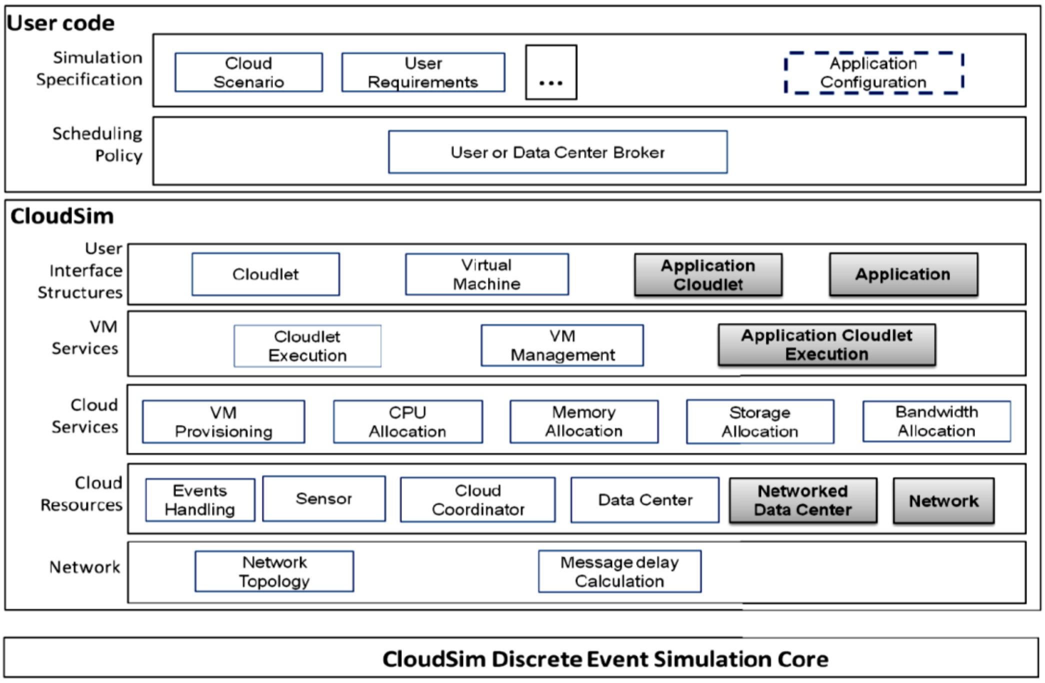

Figure 4. NetworkCloudSim Architecture

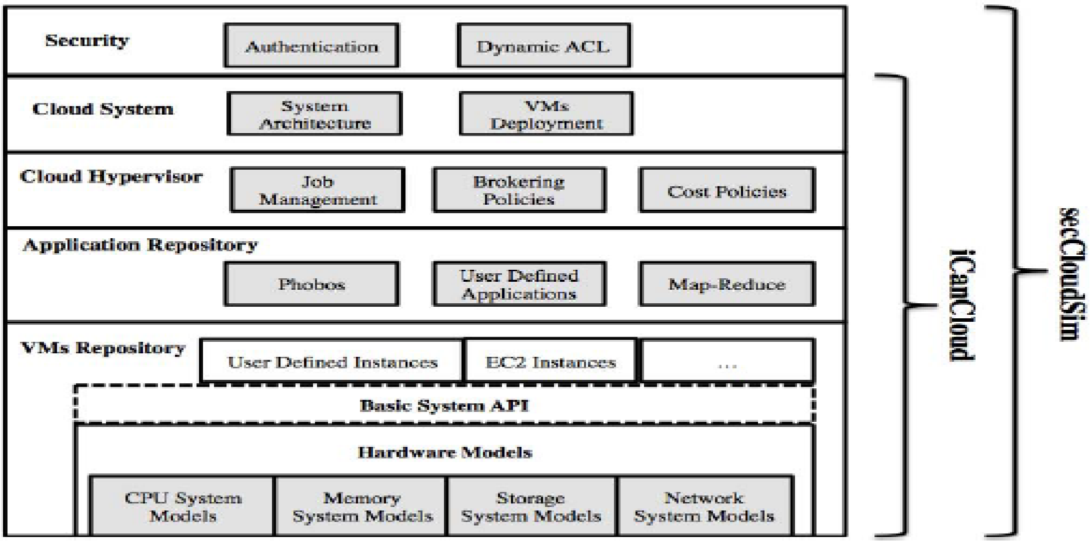

Figure 5. SecCloudSim architecture

Table 1. Cloud simulators tools' comparison.

Simulator	**Features**	**Strengths**	**Limitations**
CloudSim	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	Open Source, C++, 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	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
GroudSim	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	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	Open Source, C++	-Give the opportunity for cloud users to work on designing secure cloud simulator as a new research direction. -Allow cloud users to simulate their secure polices with the basic 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.

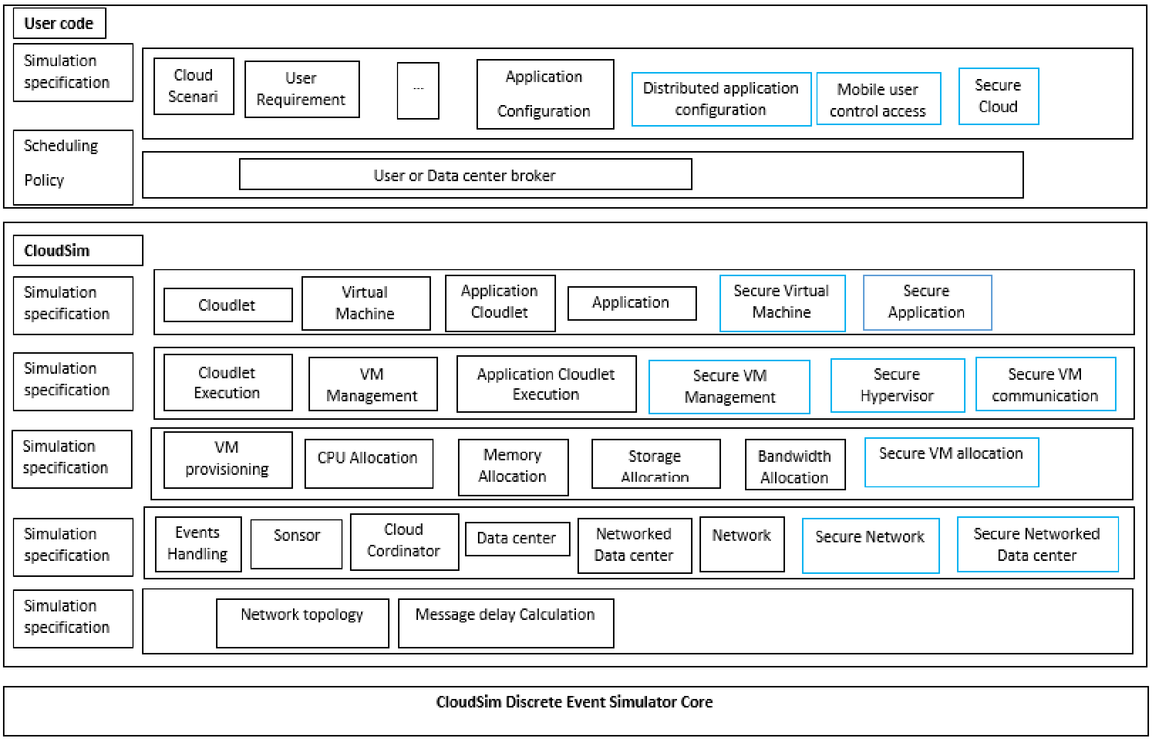

Figure 6. SecNetworkCloudSim architecture

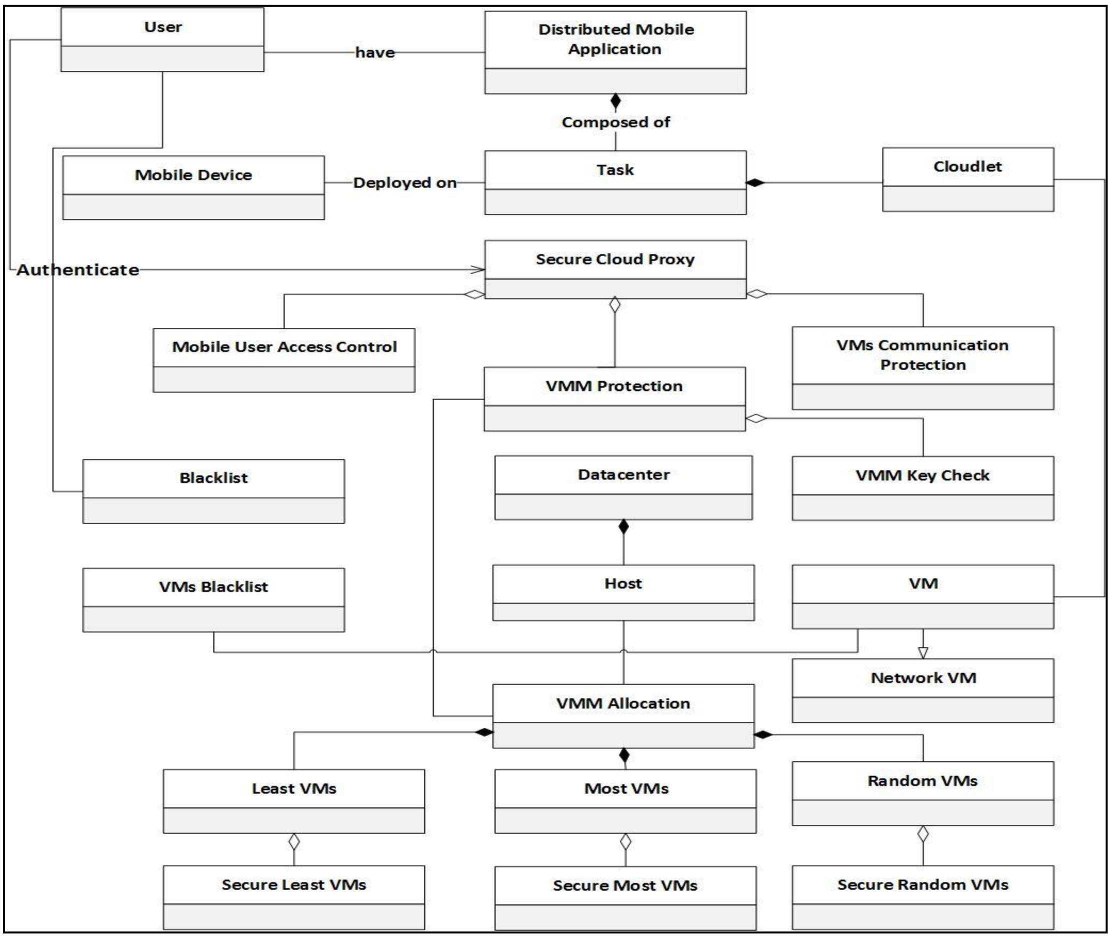

Figure 7. Secure NetworkCloudSim class diagram

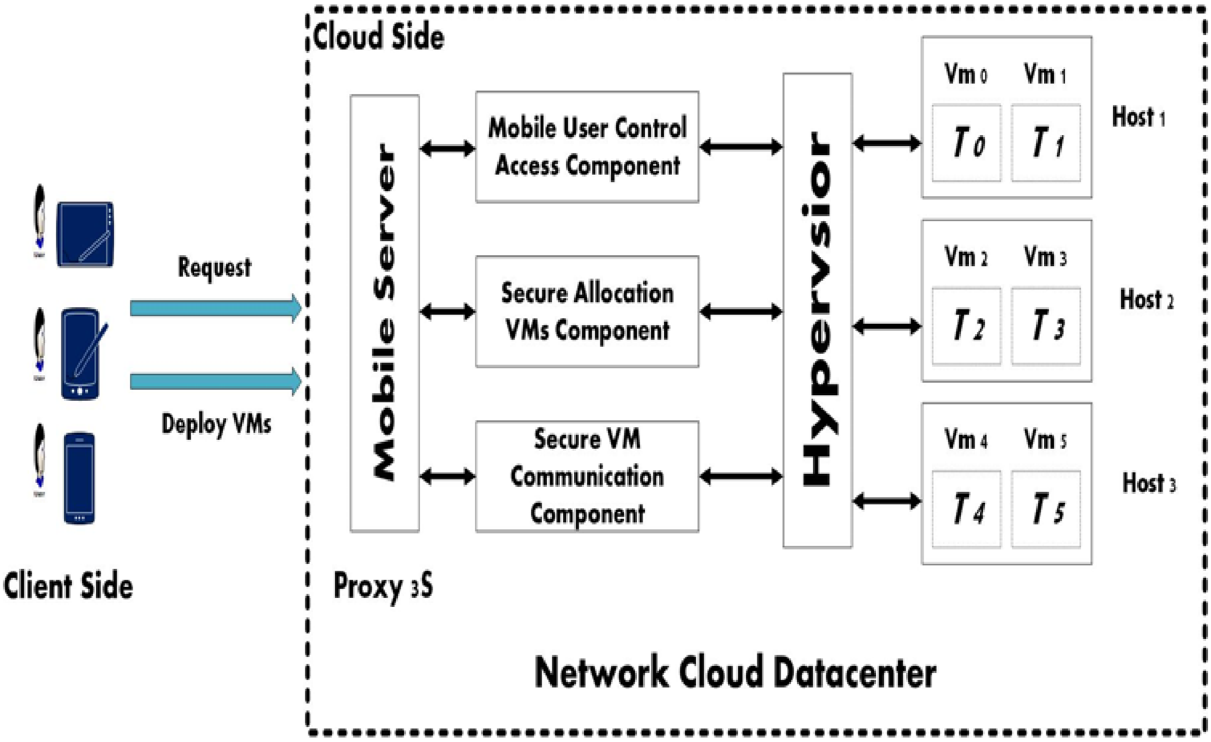

Figure 8. Steps involved in dealing with the user’s request through proxy-3S

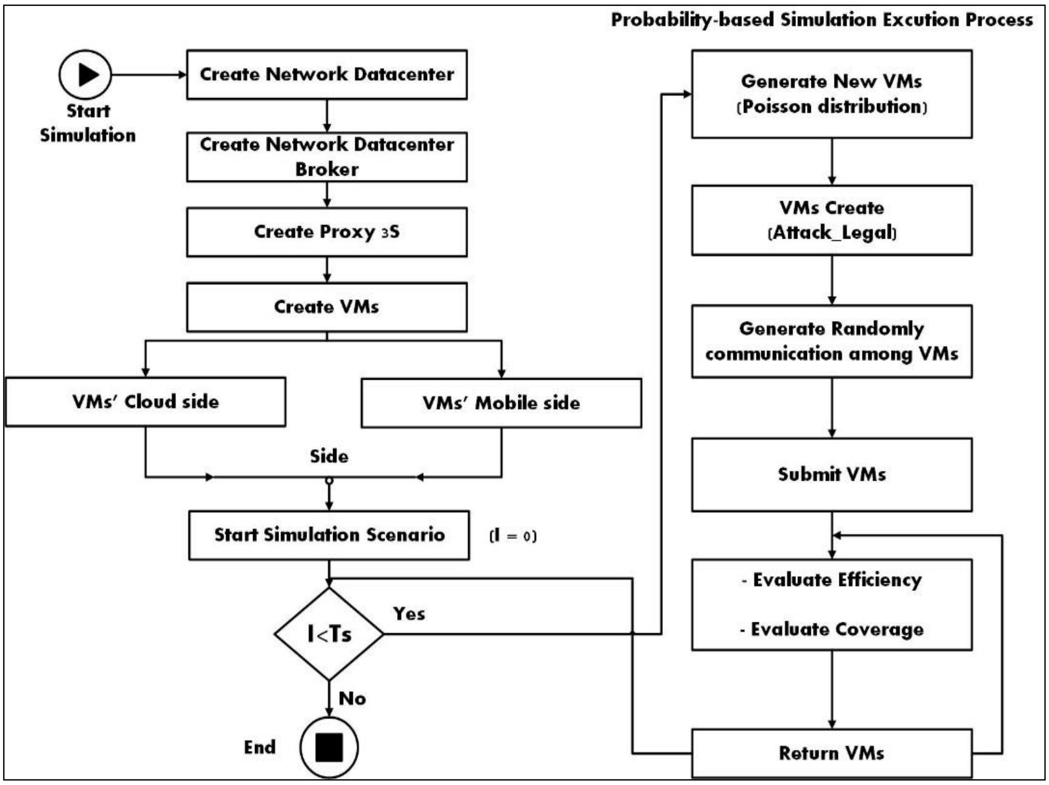
Probability-based Simulation Excution Process
Start Simulation
Create Network Datacenter
Create Network Datacenter Broker
Create Proxy 3S
Create VMs
VMs' Cloud side
VMs' Mobile side
Side
Start Simulation Scenario
(I = 0)
I<Ts
Yes
No
End
Generate New VMs (Poisson distribution)
VMs Create (Attack_Legal)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

Figure 9. Simulation execution workflow.

Table 2. 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) = ∑t VM(L, t), |Target (A)| = T
SuccTarget VM (A, t)	Sum of VMs machine attackers, which succeed to co-locate with the target, t is the exact time.
VM_interaction (A,t)	Total VMs attacker interactions channels with target VMs launched by an attacker in time t, both successful malicious communication and unsuccessful malicious communication.s
VM_interaction (L, t)	Total VMs legal interactions channels with another VMs launched by a legal VM in time t, both successful legal communication and unsuccessful legal communication.

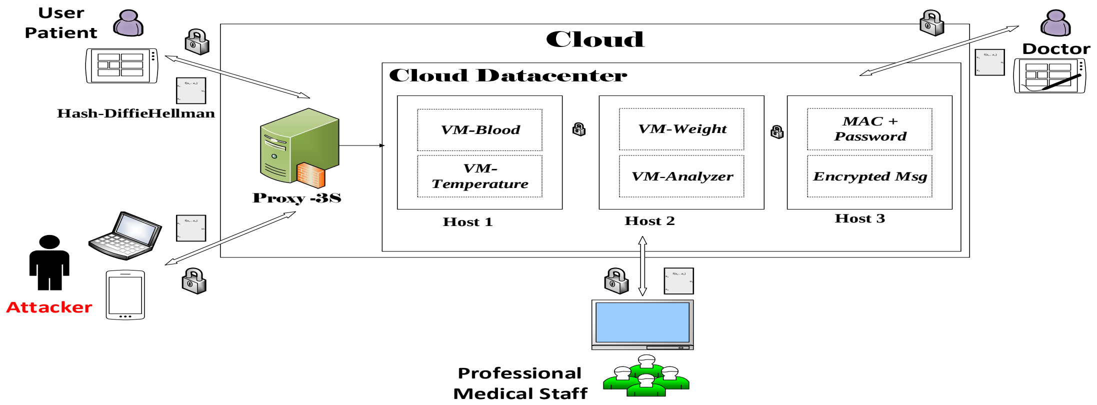

Figure 10. The architecture of the distributed Health care mobile application using Proxy-3S.

Table 4. Comparison for the smallest size configuration set between works in [19, 20] and the proposed work.

# VMs	#VMs Legal	# VMs Attacker	# Co-located VMs	# Remote-located VMs	Proposed approach		Secure VM allocation [19, 20]	
					Efficiency	Coverage	Efficiency	Coverage
50	24	26	10	9	0.3103	0.3600	0.3846	0.4166
100	47	53	23	23	0.3593	0.4423	0.4339	0.4893
150	73	77	31	29	0.2929	0.3580	0.4025	0.4246
200	95	105	45	45	0.3571	0.4245	0.4285	0.4736
250	114	136	54	53	0.3212	0.4076	0.3970	0.4736
300	135	165	69	66	0.3188	0.4285	0.418	0.5111

Table 5. Comparison for the moderate size configuration set between works in [19, 20] and the proposed work.

# VMs	#VMs Legal	# VMs Attacker	# Co-located VMs	# Remote-located VMs	Proposed approach		Secure VM allocation [19, 20]	
					Efficiency	Coverage	Efficiency	Coverage
350	220	130	88	77	0.4583	0.3347	0.6769	0.4000
400	316	84	32	26	0.2680	0.0822	0.3809	0.1012
450	339	111	44	31	0.1519	0.0763	0.3963	0.1297
500	376	124	41	38	0.1844	0.0859	0.3306	0.1090
550	403	147	53	46	0.2433	0.0985	0.3605	0.1315
600	260	340	130	105	0.2091	0.3571	0.3823	0.5000

Table 6. Comparison for the largest size configuration set between works in [19, 20] and the proposed work.

# VMs	#VMs Legal	# VMs Attacker	# Co-located VMs	# Remote-located VMs	Proposed approach		Secure VM allocation [19, 20]	
					Efficiency	Coverage	Efficiency	Coverage
650	429	221	70	64	0.2077	0.1116	0.3167	0.1631
700	520	180	58	48	0.2077	0.0666	0.3222	0.1115
750	616	134	26	24	0.1403	0.0335	0.1940	0.0422
800	567	233	79	67	0.1964	0.1063	0.3390	0.1393
850	490	360	240	207	0.3743	0.3942	0.6666	0.4897
900	631	269	143	126	0.3490	0.1440	0.5315	0.2266

Table 8. Performances comparison.

Works	Purpose	Security Degree	Encryption/ Decryption	General Drawbacks
Secure VM allocation policy [19]	VMs Co-residency	Medium	No	High efficiency, coverage of the attacks and not consider the execution time effect
Hybrid Hashing security Algorithm [21]	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	Require high processing time