International Journal of Scientific & Engineering Research, Volume 14, Issue 9, January-2023
ISSN 2229-5518

A Trade-Offs of Performance and Security in Cloud Computing Environment

†*K. Maiyama and *A. Almu
*Department of Computer Science
Usmanu Danfodiyo University, Sokoto.

Abstract— Performance and security are vital components in ensuring the quality of service (QoS) in cloud computing (CC), necessary for meeting the commitments between service providers and their customers. While high performance and availability, along with top-tier security, are sought after, they often conflict with each other. This study delves into this dilemma by conducting a quantitative assessment using a generalized stochastic Petri net (GSPN). We explore a balance between performance and security in the context of the cloud computing environment using the OpenStack Cloud Platform. Additionally, we undertake numerical tests and provide relevant interpretations that strongly suggest a typical trade-off on performance and security. Our model sheds light on how security measures can detrimentally affect the performance of cloud platform service offerings.

Index Terms—Cloud Computing (CC), OpenStack, Infrastructure as a Service (IaaS), Performance, Security, Trade-offs, generalised stochastic Petri nets (GSPN), virtual machine (VM).

†**Corresponding Author email**: Maiyama.kabiru@udusok.edu.ng

1 Introduction

COMPUTING and information system security provide mechanisms to protect systems and resources against unauthorised access [1-4]. It also guarantees authorisation of access as well as maintaining confidentiality, integrity and availability of the resources to genuine users using appropriate techniques [5]. These techniques range from access control through sophisticated authentication and authorisation as well as masking of data through encryption technologies [5-8].

Cloud computing ecosystems, including IaaS cloud computing platforms (CCPs), require a robust and adequate security mechanism to ensure the preservation of privacy, integrity and availability of resources. These mechanisms are especially critical as CCPs are built using a multitenant utility model and on top of virtualisation technology. To establish trust with subscribers, cloud service providers need to protect user privacy, including information secrecy, from the adversaries. Collaborative computation and data exchange need to be coordinated to guarantee adequate enforcement of authentication and authorisation in order to avoid leakage and unauthorised access to the resources. Cloud service customers require robust end-to-end protection of their resources within the platform against both internal and external aggressions. Firewalls and encryption/decryption mechanisms are at the heart of protecting user data and ensuring the availability of the resources. However, applying strong protections through access control, encryption, and many other methods have drawbacks, as they generate additional processing overhead thereby, affecting other QoS components such as performance and HA [9, 10]. Hence, there is a need for 'optimal' security and performance trade-offs, that is, providing a balance of the two by either compromising security for improving performance or the other way round [11, 12]. In order to address the imbalance, quantitative modelling tools are required to optimise these trade-offs through analytic modelling and evaluation.

This paper adopted and extends performance analysis study of virtualisation in OpenStack Infrastructure as a Service (IaaS) CCP [13], to investigate the impact of security on performance. In this context, a performance and security trade-off model [14] is proposed for the CCP to examine and measure security and performance attributes. To this end, an optimal trade-off between performance and security will help in attaining and maintaining acceptable SLAs. This will help in ensuring smooth processing and responding to the client request for a virtual machine (VM). A combined performance and security metric (CPSM) is formulated for illustration purposes. To enhance its computation, two CPSMs are used [11], and then parameterisation and numerical experimentation were carried out with results presented and interpreted, which shows the impact of security on the performance as anticipated.

The rest of the paper is organised as follows: Section 1.1 provides an overview of information system security. Section 2 presents selected literature on different trade-off models, while security and performance metrics are discussed in sections 2.1 and 2.2, respectively. In addition, the impact of security on performance was discussed in section 2.3. Section 3 presents a proposed combined performance and security trade-off model of virtualisation in OpenStack CCP. Section 4 presents an overview of implementation using the Möbius package. Numerical experiments, including parameterisation, results and interpretations, are presented in Section 5. Finally, the paper's concluding remarks are highlighted in section 6.

1.1 Information Systems Security Overview

The advancements in information systems have led to the emergence of devices capable of storing and transmitting vast amounts of data globally. From portable tools like laptops and mobile phones to wearables, the sheer volume of data they store equates to millions of computational storage units, making the entire sector more susceptible to threats. As the

landscape of computer networks, including corporate and cloud data centres, continues to evolve, ensuring data security, its integrity and privacy, and restricting its access to only authorized personnel becomes increasingly complex. Since cloud systems are designed to share resources with diverse users simultaneously, it's crucial to embed security measures at every level. This not only safeguards the primary aspects of data security but also ensures adherence to service level agreements (SLAs). Any breach of these SLAs might lead to significant business repercussions and potential legal disputes. Security experts in many organizations face a myriad of challenges in this context.

System security or computer security is an umbrella term used to describe any form of mechanisms used to protect the information system from theft, damage, or access to the content. Computer and information security on the other hand can be described as the protection of an information system for the purpose of achieving its intended purpose of preserving the confidentiality, availability and integrity of the system resources such as hardware, middleware, software, communication systems, information and data as well as services among others [2, 5]. Confidentiality, Integrity and Availability (CIA) are the three main widely used building blocks of computer and information security.

Cloud computing security, on the other hand, concerns with all the components highlighted above with additional constraints due to clouds distinctive properties such as virtualise service layered architecture, multitenancy and scalability, among others. The abstracted service layers (IaaS, PaaS, and SaaS) provide additional security vulnerabilities and prone to both internal (between subscribed cloud users) and external (from outside the CCP) security incidents. For example, as the cloud is built on virtualisation technology at the IaaS layer, security vulnerabilities may be associated with hardware, storage, hypervisor and VMs. Any flaws or faults in the complicated code that builds the hypervisor that varies from platform to platform could compromise the isolation between VMs hosted by the same physical machine (PM). VM snapshots, instance provisioning, live migration and other distinctive features pose threats and prone to attacks, such as the denial of service (DoS) attack resulting in data disclosure and integrity compromise. However, various strategies are applied to deter, prevent and react to any (potential) attack by adversaries [4, 15, 16].

Several security mechanisms are employed in order to minimise the impact of any security incidence and thereby, help to mitigate the adverse effects of a security attack. Some of these security mechanisms include digital signatures, hashing as well as virtualisation and data security [4].

Securing CCP through either controlling user access or masking the data through encryption helps in preserving its integrity. However, these mechanisms add overhead in system resources, thereby affecting the QoS defined in the SLA such as performance and security. As the process consumes system resource (such as CPU time), the effect, especially with stronger algorithms that are usually implemented using a longer encryption key on system performance, could result in significant breach of the SLA. In order to address this issue, a compromise may be required, which could improve one aspect and give away the other. An optimal trade-off will lead to a balance by ensuring not only a secure service but also an acceptable level of performance.

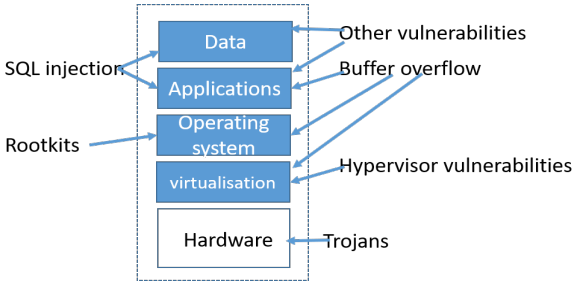

Fig 1: Typical cloud vulnerabilities [14]

In order to attain the highlighted level of optimal performance vs security trade-offs, an analysis is required that will examine the approaches of trade-off and its application in a complex environment such as the cloud. The analysis should first identify as appropriate the metrics used in evaluating computing system performance and security and then model and evaluate the optimal trade-offs between these two QoS components that are vital to SLA of virtualisation in CCPs.

2 Related Trade-off Models

Performance and security analysis in computing system has been in the literature long ago with mainly analysis of each one in isolation or as part of the QoS components. In addition, investigations to identify the relation between the two, such as how one affects the other has also been studied mostly using quantitative approaches. For example, Cho et al. [17, 18] studied wireless group communication between entities to identify and analyse a trade-off between the performance properties of the intrusion detection system (IDS) and its security. The study showed how the frequency of executing IDS can affect the performance of the group communication system and how reducing it improves performance. The quantitative analysis utilises SPN for modelling and mean time to security failure (analogous to mean time to failure in dependability and reliability analysis [19, 20]) for security metric, while service response time was used as the performance metric.

In another study of performance and security, Alentina C. et al. [21, 22], use PerfCloud tool to represent a cloud-Grid architecture for the investigation of a trade-off between performance and security. The study implemented a grid architecture on top of a CCP and studied the overhead introduced by having an additional layer. The study also compares the performance overhead introduced by cloud services with different security levels. This study differs from others by using a measurement approach to record the response time of messages passing with and without security at the transport layer and at data layer (encryption) which shows how adding security affected performance and how to attain acceptable security with decent performance.

Zeng W et al. [10] studied a trade-off model for performance and security in secured networked control systems. The study employs encryption as security element and uses a quantitative security metric of key length to determine the strength of the system security and used it to assign value to the time it takes to crack (penetrate through) a system. The mathematical model implemented in differential equations considers a brute force to determine the time it takes to gain access. The trade-off was determined by computing the level of both security and performance needed to reach optimal trade-off. The study was implemented using discrete event simulation. Although the study was conducted on the secure DC motor, their findings reflect how high-performance encryption algorithms can provide secure environments and limits the performance overhead.

Another approach to determine the trade-off between performance and security is the analysis of block cipher encryption using the data encryption standard (DES) algorithms. This kind of encryption, specifically with block cipher requires a second level of verification by correcting all the errors generated during the first level process. The study investigated the effect of giving up the error correction stage of encryption in order to improve performance [23].

Mean Time To Security Failure (MTTSF) was one of the most used security metric (analogous to mean time to failure MTTF in reliability analysis [20]) in many other trade-off models. For example in the trade-off model of combined sensing, performance and security [24], robotic ad hoc network [11, 25, 26], and mobile CCP using timing attack case study [27]. All these models use quantitative approach through GSPN implemented using different tools and illustrates either the use of encryption key length or firewall through access control to adjust the security level in return to gain more performance. The performance uses either, mean response time, node or systems throughput, node or system utilisation, among others as the performance metrics.

Alternative technique of performance and security analysis is using a delay tolerant network (DTN) is presented in [28]. The study considers an anonymous routing protocol for DTN performance analysis aspect. A mathematical model 0was developed to describe the performance and security guarantee for onion-based anonymous routing in DTN. The nature of onion ring anonymous routing requires different encryption at each layer of the (onion) message. The work is mainly aimed at bringing out the complex nature of the various layers of the onion ring (anonymous) protocol and how it affects the performance of communication between group members through evaluating the number of message copies which dictates the performance metric [28].

Lastly, Wolter K. et al. [14], developed a detailed model to study the trade-off between performance and security, by first adopting and modifying the reliability model [29, 30] in an attempt to identify a metric for security modelling. The study then identifies a performance metric of utilisation and use the two QoS components for modelling the trade-off between them. A GSPN was employed, which resulted in generating some illustrative results that show how an increase in one can drastically affect the other and vice versa. The model assumed an abstract communication system and can also be used to forecast the potential revenue generation base on satisfying both parameters of security and performance. This paper adopted the approach proposed in [14] and similar studies to illustrate how the trade-off model of CCP can be implemented primarily during a VM provisioning request in CCP. Next section presents discussions of the performance and security metrics employed for the analysis.

2.1 Security Metrics

Quantifying security and defining its metrics has long been studied [20] mostly in the area of dependability and reliability analysis. Metrics assigned are mainly centred on the applicability of the security area. In their detailed survey on security metrics, Pendleton et al. [31] propose four main categories of system security metrics mainly centred on the measurement of the dynamics of security issues. The metrics are the severity of the threats, the power of defence mechanisms, situation awareness and the vulnerability levels. However, these groups of metrics are further expanded to show the low-level metrics used to measure the security level of the system. Another intuitive approach to quantify security is expressing it in the concept of dependability, which is represented by either of the two values of work or fails. This is analogous to secure or insecure and the transition between both states as a probability by a stochastic process. This paper assumes the system exist in either secure, insecure or recovering state as outlined by Wolter (c.f., [14]). Wolter's model-based their choice on the reliability analysis with parameters of mean Time-Between-Incidents (TBI), mean Time-To-Incident-Discovery (TTID), mean Time-To-Incident-Recovery (TTIR), and mean Time-Between-Detection-and-Recovery (TBDR). The corresponding security parameters are highlighted in Figure 2.

Fig 2: Security Metric by analogy with dependability metric [14]

2.2 Performance Metrics

Measuring performance and assigning value to gauge its impact is determined using different indicators depending on the circumstance and interest of the enterprise. At each performance evaluation, given criteria and metrics are chosen mainly from the list of system service. While performance is mainly about how fast a particular task is executed, it is also, about how slow some tasks are executed. The three metrics [32] relevant to the fast or slow execution time are productivity, responsiveness and utilisation of a given task. The task ranges from retrieving data from a disk drive, time taken to process data in memory and time it takes to transport result to the file, among others. In this context, the performance metrics applied mainly in CCP performance can be the end-to-end delay for transmitting the data, response time to user's request, how busy a system is being utilised, and the probability of request being lost among others. In this context of performance and security trade-off model, the performance metrics chosen are the request loss probability and utilisation. Moreover, the metrics can be categorised as external evaluating from outside the system and internal measuring inside the system.

2.2.1. External

Mean System Response time: This is the average time taken from the total time it takes from the minute a user request is sent to the time a reply is received. This include, time at senders device, the time it takes on transit, and the time it takes to process the request and send it back to the user [32, 33].

Mean System Throughput: Is the average measure of the number of messages (the rate at which a user request is serviced) are transmitted at a given time between two entities (nodes) [32].

2.2.2. Internal

Mean System Utilisation (ρ): The utilisation of a node is measured as the average fraction of time a given device is busy processing a request before forwarding it to the next node or send it back to the requesting user [33]. The other fraction is when the device is idle and the percentage of busy plus the percentage of idle must be equal to 100.

Request Loss Probability: A request loss probability describes the fraction of messages (request) that was dropped as a result of either buffer capacity is full or the processing (encryption) time is longer than usual which result in message time to live expirations.

2.3 Impact of Security on Performance

In general, the impact is in two folds, that is encryption & decryption and authentication & authorisation:

a) **Encryption and Decryption**: Algorithms that implement the encryption and the security protocol cost the computing power time and effort in encrypting and/or decrypting the message from plain text to cipher text. The longer the encryption key, the harder to break (i.e. more secure), as well as longer to process. The computational effort spent in encrypting degrades the performance. In other words, the longer the encryption key, the higher the security of encrypted messages.

Authentication and Authorisation: This is one of the levels of checking access credentials (e.g. username and password) supplied by the user during the quest to gain access to the resource against the credentials in the system, and when matched, access is granted. The time it takes to authenticate depends on the network speed, query engine power and location of the database. Many additional layers of authentication exist, ranging from checking sender's browser certificate to the most widely used two-level authentication. The two-level authentication sends verification message (text messages, email, and through an app) back to the user to reaffirm that he/she is the one trying to gain access. The message contains some one-time-password that the user will enter and when matched, will gain access. Although the two factor (level) authentication adds a layer of security, it resulted in a long time to gain access to the resource.

3 Combined Performance and Security Model CCP

The following illustrative model was based on the schematic architecture of the OpenStack CCP from our earlier work [13]. The model was enhanced for the combined performance and security trade-off model (CPSM). The scenarios assumed is for receiving, processing and returning user request for the provisioning of VM instance. However, this model is only concerned with the security processing component that encrypts the message before sending it back to the user. The Output node is responsible for packaging and sending the result back to the requesting user with the successful creation of VM or otherwise. The following diagram (c.f., Figure 3) shows the additional component from the figure that contains both the encryption and the decryption part. Meanwhile, this model is only implemented into the encryption component, which encrypts and transmit the message back to the requesting user. Future work will consider the whole schematic architecture.

This model depicts the architecture and method proposed in [14] and apply it into the OpenStack CCP architecture. This is through zooming the Output node and splitting it into two sub-nodes that is an Encrypt and Transmit sub-nodes (c.f., Figure 3). The model also considers the source of the message from the previous node and forwarding it to the next node onward.

As described in the literature, one of the mechanisms for securing an information system, including CCP, is through access control and encryption of the data. On encrypting data, the longer the encryption key, the more secure the system is, and the more processing cycle is required to encrypt or decrypt it.

Figure 4 shows the CPSM using a GSPN with named places and transitions. In this context, the performance model component (c.f., Figure 4 left) can be evaluated by obtaining the performance metric such as the utilisation ρ_c. Security part (c.f., Figure 4 right) likewise, can be evaluated using the probability of system in Secure state P_c.

In this context, the CPSM is considered as an optimisation problem by computing two CPSMs with maximisation and minimisation metrics. These are CPSM-Maximum (CPSM1), and CPSM-Minimum (CPSM2) [11]. These two CPSMs are defined as follows:

CPSM1 is defined as the sum of the probability of the system in a 'Secure' place plus the utilisation at the Transmit transition. In order to optimise the encryption time, the CPSM1 is maximised, namely:

$$max\ CPSM1 = max\ \{P(Secure) + \rho(Transmit)\} \quad (1)$$

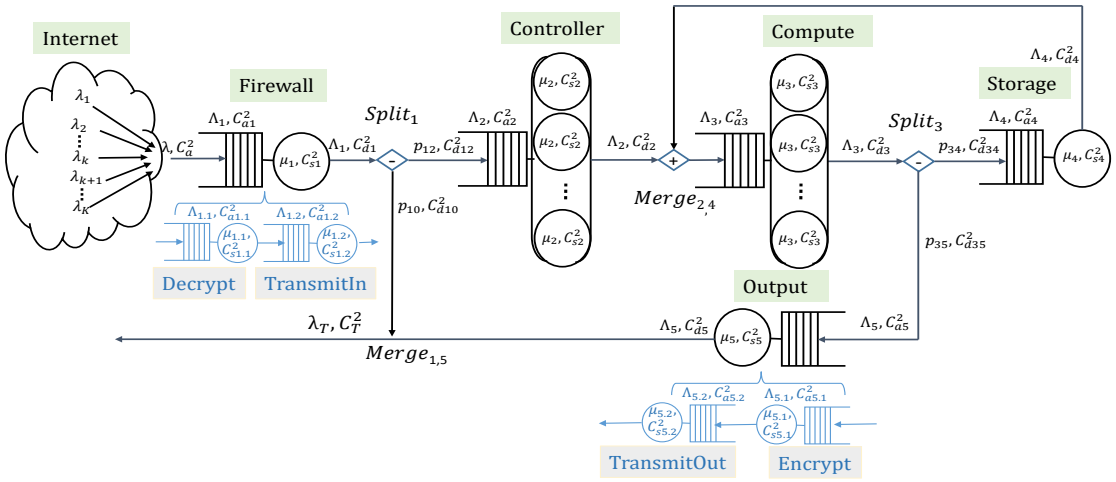

Fig 4: Schematic diagram for an open GE-type QNM of OpenStack VM provisioning requests with propose decryption and encryption extension [13].

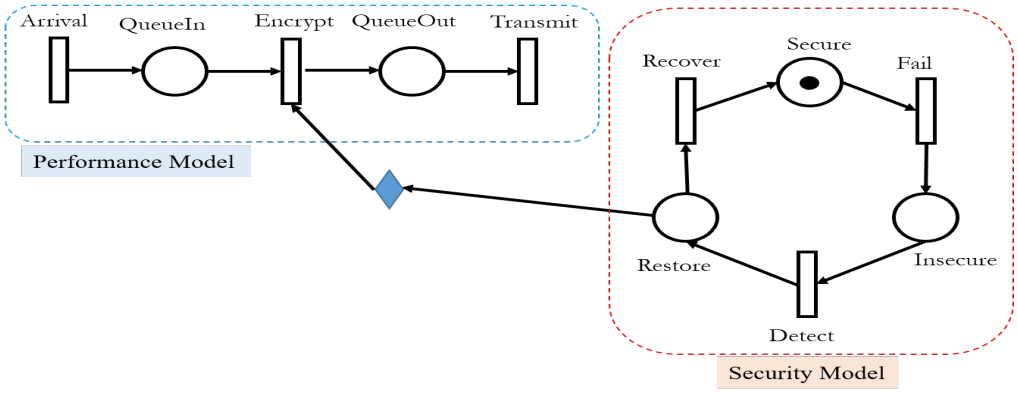

Figure 3: Combined performance and security trade-off analysis model [39]

Similarly, CPSM2 is define as the sum of the probability of the system in an 'Insecure' place plus the request loss probability (RLP) at QueueIn place. The QueueIn place is considered as a finite capacity buffer, and whenever the security detection control model (SDCM) fires frequently, the buffer can fill up, and the request that cannot be accommodated in the QueueIn will be dropped. This will logically increase the utilisation and throughput at the Transmit transition. In order to compliment the max optimisation, the CPSM2 is minimised, namely

$$min\ CPSM2 = min\ \{P(Insecure) + RLP\} \quad (2)$$

By maximising the CPSM1, as well as minimising the CPSM2 is expected to point to the same place and hence as expected, will provide the optimal trade-off between the two QoS components.

3.1 Performance Model Component

The performance model component is denoted by three nodes represented by the GSPN notations as Arrival, Encrypt and Transmit transitions. The arrival signifies the packaged reply that is going to be sent back to the requesting user. The response represented by Arrival is placed in the QueueIn and subsequently fired into the Encrypt transition. As long as the inhibitor arc from the security sub-model is not activated, it will process (encrypt) the message and forward it to the QueueOut which eventually fire and forwards it to the Transmit transition for sending it to the next node. The Encrypt transition will continue to fire until when a message is sent from the security component telling it otherwise (e.g. to freeze the operation because of detecting an attack).

3.2 Security Model Component

Security detection control model (SDCM) is the model detailing the scenario of security handling when an attack happens. For illustration purpose, this research considered stages and the changes that happen, which triggers the firing of the token in the model from one place to the other. The model is presented using the GSPN, which comprises of three places (Secure, Insecure and Restore) as well as the transitions (Fail, Detect, and Recover). The initial step happens when the token is in a secure state and moving from one place to another through respective transitions. The model starts with a service time that is exponentially distributed, with a mean service rate of μ_F, μ_D and μ_R representing Fail, Detect and Recover transitions, respectively. A token moves from one input place to the next output place at the end of each transition.

It is assumed that the system starts with a 'Secure' state (c.f., Figure 5 – Step 1), which signifies that it is enabling the Fail transition. The mean firing rate μ_F, μ_D and μ_R represents Fail, Detect and Recover rates, respectively. When a security breach happens, the 'Fail' transition is fired, and the security state changes from 'Secure' to 'Insecure'. In between the two successive security breaches is the mean inter-fail time, when a new breach happens, the 'Fail' transition fires with a mean rate μ_F. A token is then taken from the 'Secure' place into the 'Insecure' state, which enables the 'Detect' transition (c.f., Figure 5 – Step 2).

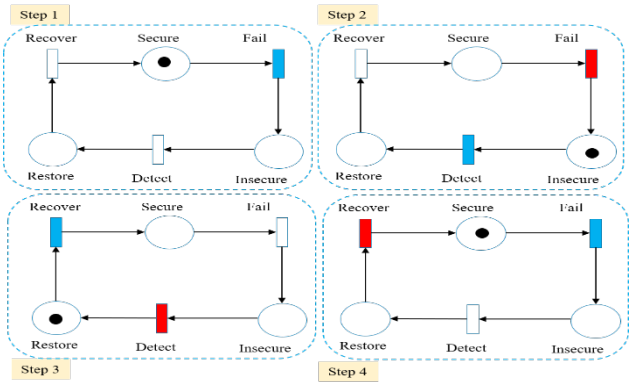

Fig 5: Security detection control model (SDCM) for CPSM

Once the system notices a security incident, the Detect transition will fire, and the state will change from 'Insecure' to 'Restore'. At the end of the time between two successive detects (inter-detect time), the security breach is detected, the 'Detect' transition fires with a mean rate μ_D and the token is removed from 'Restore' place, thereby enabling the 'Recover' transition (c.f., Figure 5 – Step 3). Then, the system is recovered, which allow the 'Recover' transition to fire making the token move from 'Restore' state back to the 'Secure' state (c.f., Figure 5 – Step 4). At this point, the time between two successive recoveries is given by μ_R.

At the end of one cycle, the time a token spent at one place is recorded in the period under observation T. Hence, the probability of the time spent by token is cumulatively determined with the observational period T. In addition, the probability that the SDCM is at each state (i.e., 'Secure', 'Insecure', and 'Restore') during period T can be determined. Thus, at the end of T, the aggregate time spent at each state is divided by T to obtain the proportion of time that the CCP is found at random period to be in 'Secure', 'Insecure', and 'Restore' places.

4 CPSM Implementation using Möbius Petri Net Tool

The CPSM was implemented using a Möbius package. The package was configured, and input data were supplied using the illustrative data from Table 1. The simulation was executed and results recorded.

4.1 Möbius Petri Net Overview

Möbius is a software tool developed at the University of Illinois for modelling and analysis of the behaviour of complex systems. Initially developed for the modelling and evaluation of system dependability, availability, reliability and performance of computing and communication systems, its usage expanded more. Its usage includes discrete event simulation, performance modelling and evaluation, biochemical and gene sequencing analysis.

Modelling with Möbius allows for the combinations of different modelling approaches. For example, it enables the Replica/Join composition technique via its SPN equivalent objects known as Stochastic Activity Networks (SAN). This feature allows the implementation of a number of formalisms such as GSPN, and PEPA. SANs features enable a compact representation of systems with the ability to use stochastic processes such as PN which gives it more acceptance compare to other packages [34]. Möbius has extensible objects that allow designing, parameterising and specifying distributions, as well as executing the model.

Modelling can be achieved using an Atomic Formalism feature. SAN models, in particular, are stochastic extensions of Petri Net. In this context, the SAN formalism was employed, and relevant objects were used. The objects of the SAN model include four primitive objects analogous to the tuples of the GSPN (c.f., Figure 6). These are places, activities, input gate, and output gates corresponding to the places, transitions, input places and output places, respectively.

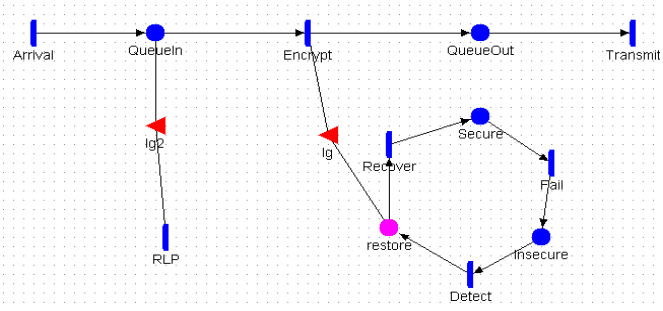

Fig 6: Implementation of CPSM using Mobius SAN

5 Numerical Experimentations and Result

This section discusses the parameterisation and experimentation to illustrate the trade-off model. It presents the results and interpretation showing how security (longer key length) affects the performance of CCP and how optimal trade-off is attained. This demonstrates an acceptable level of compromise between these two QoS components for an acceptable level of SLA. The model assumes the longer the security key, the longer it takes to encrypt the message, which will affect the performance of the CCP. An optimal trade-off (obtained by optimal key length) may be determined by assigning suitable metrics for the CPSM.

For illustration purpose, max (CPSM1) and min (CPSM2) are determined. CPSM1 is the value for the probability of the system in a Secure state and the utilisation of the Transmit transition that is P(Secure) + ρ(Transmit). Likewise, the min (CPSM2) is the sum of probability of system in Insecure state and the RLP of the Encrypt transition that is P(Insecure) + RLP(Encrypt). The rationale behind augmenting these metrics is to find the optimal point on the curve for both CPSM1 and CPSM2. If the optimal value points to the same encryption time, then the model is evaluating the optimal combined performance and security as expected. Next section provides details of parameters used and the experimentations with a discussion of the results.

5.1 Parameterisation and Experiments

The CMPS trade-off model is implemented by choosing and assigning numerical values and executing the representative experiment for the identification of optimal trade-off between performance and security. The architecture operates when a message arrives for encryption and then transmitting back to the requesting user. As most of the modelling approach, this work chooses a firing delay using both exponential (M-type) distribution and bursty (GE-type) distribution. GE-type was determined by H_2 with a higher value of tuning parameter (k). The parameters (c.f., Table 1) that handle the processing (encrypting) and transmitting the message will determine the performance measure while the security aspect is determined by the values assigned to the SDCM transitions. Specifically, the Encrypt and Fail vary, as the research is interested in the impact of security on performance. The study also assumes that the longer the encryption key, the more secure the system is and will take a longer time to process, thereby adding more performance overhead.

For illustration purpose, the parameters used are presented in Table 1 and were carefully chosen to show the impact of security on performance. The performance metrics are the utilisation (ρ) of the Transmit transition and the request drop probability at the Encrypt transition, while the security metrics are represented by the probabilities of a request being in Secure or Insecure states. As probability values are less than 1, the values for the performance used were also closer to the security in order to see the apparent effect of security on performance. This model's problem mainly affects the Encrypt and Fail transitions as the time required encrypting the message and the time taken during a security breach to break the encryption key. These values were inputted into the Möbius tool in order to obtain analytical results and to visualise the impact of security on performance as well as determining the optimal trade-off.

Table 1: Assigned values for the parameters of the Trade-off Model

Parameters	Value (time)	Comments
Arrival	0.125	`Mean Inter-Arrival' time
Encrypt	0.01 to 0.34 by 0.01	'Mean Encryption' time
Transmit	0.1	'Mean Transmission' time
Fail	1.25, 2.5, 5, 10, …, 1510	'Mean Security Inter-Fail' time (after 10, rises by 50)
Detect	12	'Mean Inter-Detect' time
Recover	36	'Mean Inter-Recover' time
Finite capacity	10	Finite capacity of QueueIn and QueueOut places

The experiment was simulated with 100 runs in order to collect the result and subsequent interpretation. The execution has less than a 5% confidence level, and the results were collected and plotted in the next section.

5.2 Results and Interpretations

The M-type arrival process of the max(CPSM1) = p(Secure) + Utilisation(Transmit) (c.f., (1)), is presented in Figure 7. It shows the utilisation of the Transmit transition. The red curve stands for the utilisation. While the blue curve represents the security metric, which is the probability of the system in a secure state. As the red curve raises and continues to move towards the maximum value of 1, it signifies the system will continue to encrypt the message before forwarding it to the next transition because it is in a secure state. When encryption time is short, the message waiting time is short, and that indicates that there is a high possibility that an attack can compromise it. In addition, when the encryption time is high, and the process takes a longer time, the messages will saturate the buffer of the encryption node (Encrypt transition). Hence, an increase in overall processing time, which in turn affects the utilisation and throughput of the Transmit transition. In addition, for an encryption key length, the security level is maintained until when the key length is increased, which will affect the next node (Transmit transition) performance attributes such as utilisation and utilisation.

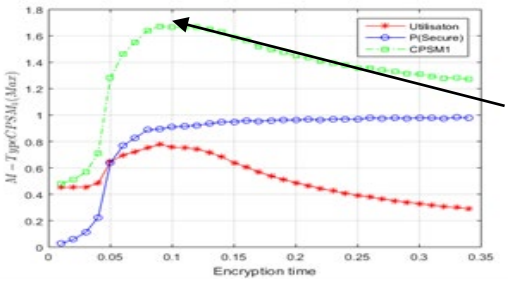

Fig 7: M-type CPSM1

Therefore, the short encryption key lengths make the utilisation of the transmitting node (Transmit transition) starts at a low level due to the small security key length. This will result in the firing of the 'Fail' transition and eventual recurrent enabling of the inhibitor arc (represented by diamond shape), thereby freezing the encryption process and denying the movement of a message to the Transmit transition more often. Likewise, when encryption key length increases, the firing of the 'Fail' transition and subsequent enabling of the inhibitor arc reduces, hence increasing the throughput and utilisation of the Encrypt transition.

Conversely, when then encryption key length is more than the threshold, the Encrypt transition will become a bottleneck, which results in a slow rate of departure and declining the throughput and utilisation of the Transmit node. The optimal value of the encryption time in this scenario max (CPSM1) is around 0.09 seconds.

Figure 8 presents a min (CPSM2) with the performance metric RLP in 'QueueIn' place of the performance component as well as the probability of the CCP in 'Insecure' state, i.e. min (CPSM2) = RLP + p(Insecure) (c.f., (2)). Increase in key length led to rising encryption time, which leads to the build-up of messages in the 'QueueIn' place, and results in loss of messages affecting the performance (c.f., Figure 8). Therefore, the probability of an encryption key being compromised must be reduced, leading to the platform moving into the recovery mode. Hence, whenever the encryption is quick enough, and the corresponding key length is secure, then the encryption process is said to be at its optimum.

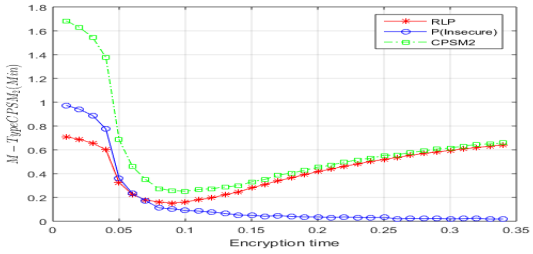

Fig 8: M-Type CPSM2

5.2.1 Impact of Traffic Burstiness on CPSM

The max (CPSM1) does not affect the CCP when the arrival process bursts using a family of H_2 distribution or even GE-type distribution. Figure 9 presents the max (CPSM1) for different values processed at all the transitions in both performance and security sub-models with varying level of traffic burstiness. The figure clearly shows that when the value of tuning parameter k is less, it signifies that the interarrival times results in declining the rate of transfer of request to the 'QueueIn' place. As the traffic rate increases, the messages drop rate will increase, thereby affecting the throughput and utilisation at the Transmit transition, which will cause the max (CPMS1) curves to shift down appropriately.

The increase in traffic burstiness, according to the computed result, did not affect the optimal value for both max (CPSM1) and min (CPSM2). More importantly, the optimal value at both max (CPSM1) and min (CPSM2) are at the same place (i.e. 0.09 sec) as expected.

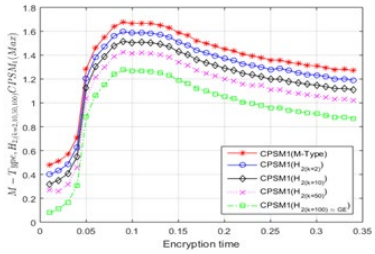

Fig 9: M-Type, $H_{2(k=2,10,50,100=GE)}$ CPSM1

Figure 10 depicts how increasing burstiness affects the request drop rate of traffic. Clearly, it shows that when arrival follows bursty flow at the QueueIn place, it results in a higher drop rate of messages entering the Encrypt transition.

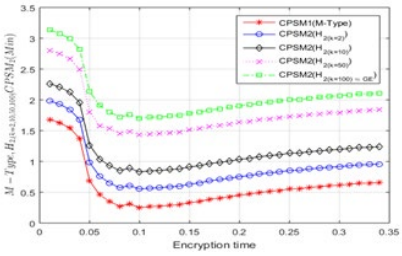

Fig 10: M-Type, $H_{2(k=2,10,50,100=GE)}$ CPSM2

6 Conclusions and Future Work

This paper focuses on the quantitative analysis of a generalised stochastic Petri net (GSPN) model for the 'optimal' performance and security trade-offs of an OpenStack IaaS CCP architecture with virtualisation using the M¨obius tool. Consequently, 'optimal' combined metrics of an OpenStack IaaS cloud computing platform (CCP) architecture were determined. Typical numerical experiments were conducted and appropriate interpretations were made.

The results will provide a useful insight into identifying the adverse impact of security on performance using the 'optimal' length of the encryption time. This will provide ample support to the architects and performance engineers during the tuning and capacity

planning of existing CCP architectures as well as the design of new ones to consider evaluating such 'optimal' trade-offs on the impact of security on performance under both normal and heavy traffic workloads.

Moreover, future investigations will involve applications of machine learning techniques for the prediction of future security incidents on CCPs and associated intrusion preventive measures, as appropriate.

ACKNOWLEDGMENT

The authors wish to thank D Kouvatsos for his contribution to the earlier version of this research work. We also wish to express their thanks to unknown reviewers for their critique, comment and feedback which helped improve the quality of this manuscript and constructive.

.

REFERENCES

[1] S. Bosworth and M. E. Kabay, *Computer security handbook*, 6th ed.: John Wiley & Sons, 2014.

[2] B. Guttman and E. A. Roback, *An introduction to computer security: the NIST handbook*: DIANE Publishing, 1995.

[3] B. R. Kandukuri and A. Rakshit, "Cloud security issues," in *2009 IEEE International Conference on Services Computing*, 2009, pp. 517-520.

[4] R. L. Krutz, R. D. Vines, and G. Brunette, *Cloud security: A comprehensive guide to secure cloud computing*: Wiley Indianapolis, 2010.

[5] W. Stallings, *Cryptography and network security: principles and practice*: Pearson Upper Saddle River, 2017.

[6] A. B. El-Sisi and H. M. Mousa, "Evaluation of encryption algorithms for privacy preserving association rules mining," *International Journal of Network Security,* vol. 14, pp. 289-296, 2012.

[7] M. Umaparvathi and D. K. Varughese, "Evaluation of symmetric encryption algorithms for MANETs," in *2010 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2010*, 2010, pp. 182-184.

[8] M. Xin, "A Mixed Encryption Algorithm Used in Internet of Things Security Transmission System," in *Proceedings - 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2015*, 2015, pp. 62-65.

[9] H. Zo, D. L. Nazareth, and H. K. Jain, "Security and performance in service-oriented applications: Trading off competing objectives," *Decision Support Systems,* vol. 50, pp. 336-346, 2010.

[10] W. Zeng and M.-Y. Chow, "A trade-off model for performance and security in secured networked control systems," in *2011 IEEE International Symposium on Industrial Electronics*, 2011, pp. 1997-2002.

[11] G. M. Miskeen, D. D. Kouvatsos, and E. Habibzadeh, "An exposition of performance-security trade-offs in ranets based on quantitative network models," *Wireless personal communications,* vol. 70, pp. 1121-1146, 2013.

[12] V. Zorkadis, "Security versus performance requirements in data communication systems," in *European Symposium on Research in Computer Security*, 1994, pp. 19-30.

[13] K. M. Maiyama, D. D. Kouvatsos, B. Mohammed, M. Kiran, and M. A. Kamala, "Performance Modelling and Analysis of an OpenStack IaaS Cloud Computing Platform," in *IEEE 5th International Conference on Future Internet of Things and Cloud*, Prague Chzeck Republic, 2017, pp. 198-205.

[14] K. Wolter and P. Reinecke, "Performance and security tradeoff," in *International School on Formal Methods for the Design of Computer, Communication and Software Systems*, 2010, pp. 135-167.

[15] J. R. Vacca, *Cloud computing security: foundations and challenges*: CRC Press, 2016.

[16] C. Modi, D. Patel, B. Borisaniya, A. Patel, and M. Rajarajan, "A survey on security issues and solutions at different layers of Cloud computing," *The journal of supercomputing,* vol. 63, pp. 561-592, 2013.

[17] J.-H. Cho and R. Chen, "On design tradeoffs between security and performance in wireless group communicating systems," in *1st IEEE ICNP Workshop on Secure Network Protocols, 2005.(NPSec).* 2005, pp. 13-18.

[18] J.-H. Cho, I.-R. Chen, and P.-G. Feng, "Performance Analysis of Dynamic Group Communication Systems with Intrusion Detection Integrated with Batch Rekeying in Mobile Ad Hoc Networks," pp. 644-649, 2008.

[19] B. B. Madan, K. Gogeva-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "Modeling and quantification of security attributes of software systems," in *Proceedings International Conference on Dependable Systems and Networks*, 2002, pp. 505-514.

[20] F. C. Freiling, "Introduction to security metrics," in *Dependability metrics*, ed: Springer, 2008, pp. 129-132.

[21] V. Casola, A. Cuomo, M. Rak, and U. Villano, "Security and performance trade-off in perfcloud," in *European Conference on Parallel Processing*, 2010, pp. 633-640.

[22] V. Casola, A. Cuomo, M. Rak, and U. Villano, "The CloudGrid approach: Security analysis and performance evaluation," *Future Generation Computer Systems,* vol. 29, pp. 387-401, 2013.

[23] S. Wei, J. Wang, R. Yin, and J. Yuan, "Trade-off between security and performance in block ciphered systems with erroneous ciphertexts," *IEEE Transactions on Information Forensics and Security,* vol. 8, pp. 636-645, 2013.

[24] C. Ejike and D. Kouvatsos, "Combined sensing, performance and security trade-offs in cognitive radio networks," in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, 2017, pp. 1-4.

[25] G. Miskeen, D. Kouvatsos, and M. Akhlaq, "Performance and security trade-off for routers in high speed networks," in *the 26th UK Performance Engineering Workshop (UKPEW 2010)*, 2010, pp. 119-128.

[26] M. J. Ikram and K. Ahmad, "Investigation of Performance-Security Tradeoff in Robotic Mobile Wireless Ad hoc Networks (RANETs) using Stochastic Petri Nets," *International Journal of Computer Applications,* vol. 62, 2013.

[27] T. Meng, "Security and Performance Tradeoff Analysis of Offloading Policies in Mobile Cloud Computing," 2017.

[28] K. Sakai, M.-T. Sun, W.-S. Ku, J. Wu, and F. S. Alanazi, "Performance and security analyses of onion-based anonymous routing for delay tolerant networks," *IEEE Transactions on*

Mobile Computing, vol. 16, pp. 3473-3487, 2017.
[29] B. Littlewood, S. Brocklehurst, N. Fenton, P. Mellor, S. Page, D. Wright*, et al.*, "Towards operational measures of computer security," *Journal of computer security,* vol. 2, pp. 211-229, 1993.
[30] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing,* vol. 1, pp. 11-33, 2004.
[31] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Computing Surveys (CSUR),* vol. 49, p. 62, 2017.
[32] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*: John Wiley & Sons, 1990.
[33] B. Gregg, *Systems Performance: Enterprise and the Cloud*: Pearson Education, 2013.
[34] S. Maza, "Stochastic activity networks for the modeling of repairable systems including diagnosis performance," in *8ème Conférence Internationale de Modélisation et Simulation, MOSIM'10*, 2010, p. CDROM.