Quantitative Security Metrics for Secure VM Migration Protocols

By

Tayyaba Zeb
2011-NUST-MS-CCS-36

Thesis Supervisor

Dr. Abdul Ghafoor
Department of Computing

A thesis submitted in partial fulfilment of the requirements for the degree
of Masters in Computer and Communication Security (MS CCS)

In

Department of Computing (DoC)
School of Electrical Engineering & Computer Science (SEECS)
National University of Sciences & Technology (NUST),
Islamabad, Pakistan
(March, 2015)

Approval

It is certified that the contents and form of the thesis entitled "**Quantitative Security Metrics for Secure VM Migration Protocols**" submitted by **Tayyaba Zeb** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Abdul Ghafoor**

Signature: ________________________

Date: ________________________

Committee Member 1: Dr. Awais Shibli

Signature:
____

Date: ________________________

Committee Member 2: Dr. Ali Haider

Signature:
____

Date: ________________________

Committee Member 3: Dr Zahid Anwar

Signature:
____

Date: ________________________

Dedication

To The Martyred Students of APSACS, Peshawar.

And their Mothers.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at National University of Sciences & Technology (NUST) School of Electrical Engineering & Computer Science (SEECS) or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Tayyaba Zeb

Signature: ______________________________

Acknowledgement

This thesis would not have been possible without the continuous support of my supervisor **Dr. Abdul Ghafoor**. His guidance, motivation and mentorship by far had been the most encouraging factors to complete my research work. I also thank my committee members, friends and family for their guidance and support.

Table of Contents

List of Abbreviations

AES	Advanced Encryption Standard
CBM	Cost Benefit Measure
Cent OS	Community Enterprise Operating System
DY Attack Model	Dolev-Yao Attack Model
ECDH	Elliptic Curve Diffie-Hellmann
FISMA	Federal Information Security Management Act
FIPS	Federal Information Processing Standard
GPRA	Government Performance and Results Act
IAAS	Infrastructure As A Service
KCI	Key Compromise Impersonation
KVM	Kernel based Virtual Machine
NISynch	Non-Injective Synchronisation
NIST	National Institute of Standards and Technology
PKI	Public Key Infrastructure
SHA	Secure Hash Algorithm
SPDL	Security Protocol Description Language
VM	Virtual Machine
VMM	Virtual Machine Monitor

List of Figures

List of Tables

Abstract

Lord Kelvin once said that "if you cannot measure it, you cannot improve it". Quantitative security metrics has been a challenging area so far. Defining enterprise level security metrics has been listed as one of the hard problems in the Infosec Research Council’s hard problems list. Almost all the efforts in defining absolute security metrics for the enterprise security have not been proved fruitful. However, with the maturity of the security industry, there has been a continuous emphasis from the regulatory bodies on establishing measureable security metrics. Security metrics help to make functional and business decisions for improving the performance and cost of the security controls. This thesis proposes a relative security metric model that derives three quantitative security metrics of efficiency, effectiveness and cost/benefit measure of security controls. Similarly, virtualization technologies are rapidly changing the landscape of the computing world. Devising security metrics for virtualized environment is even more challenging. This thesis took the secure virtual machine migration process as case study and applied the relative security metric model for measuring the efficiency, effectiveness and cost/benefit measure of the secure VM migration protocols. As secure VM migration is an evolving area and no standard protocol is available specifically for secure VM migration, therefore, this thesis first proposes a lightweight secure VM migration protocol and then applied the proposed relative security metric model in order to compute the security performance of the proposed protocol.

Chapter 1: Introduction

1. Introduction

In this chapter explanation of basic concepts and overview of the research work carried on during the thesis phase is elaborated. This chapter starts off with brief introduction of basic terms such as security metrics, virtualization, hardware virtualization and process of virtual machine migration. Afterwards the need for development of security metrics in order to perform security evaluation of security sub systems/protocols is explained. Later sections provide the motivation for carrying out the research in the field of quantitative metrics and virtual machine migration. In the end of this chapter, a brief yet comprehensive discussion on major contribution of the conducted research and scope of the work is discussed. The chapter ends with description of thesis organization.

1.1. An Overview of Security Metrics

Investments in the field of information security have increased significantly since last decade but it has become a matter of concern that still there exists no particulars ways which exactly measure the information security performance. A widely accepted management principle states that if you cannot measure it then you surely cannot manage it. In this regard, designing security metrics to measure information security performance as recommended by NIST IR 7564[35] can be of great help. SANS guide to security metrics states that threat, vulnerability and asset value are the critical elements that can be incorporated into security metrics. Efficient and effective security metrics are the computable functions which produce meaningful quantifiable information which shows trends in efficiency of security sub system and aids in decision making. Security metrics can measure efficiency and effectiveness of security controls and indicate the level to which security objectives are being met by security controls. Developing absolute security metrics is hard problem because field of information security has many unknown values such as unknown number of adversaries and weaknesses a system can have. However developing relative security metrics is a manageable task which compares two or more attributes of the security sub system to measure their efficiency and effectiveness. NIST SP 800-55 Rev. 1[36] has categorized metrics into implementation metrics, impact metrics and effectiveness or efficiency metrics. The first category looks for progress in implementing security controls related policies and procedures. Impact metric delivers the information security program's impact on organization mission. The efficiency and effectiveness metrics measure the results and effects of security controls on the system's security performance. Truly useful metrics are those which provide the extent to which security goals are being met by security controls and drive actions to improve overall performance of security programs [37].

1.2. An Overview of Virtualization

Virtualization technology is used to create a virtual version of computing resource such as operating system virtualization, hardware virtualization, storage virtualization, network virtualization etc. This idea was first introduced in 1960's where IBM's main frame computers which were then under-utilized used virtualization to share the hardware resources among multiple users to increase efficiency of both consumers and expensive hardware resources. At that time this idea was a breakthrough because it made possible for organization and even individuals to use an expensive machine without actually buying it. Later, with the passage of time and introduction of low priced computers and pc's use of this idea got restricted. Virtualization again got a boom in 1990's for the very same reason. A single server machine has such a huge capacity that it was hardly possible for most of the workloads to fully utilize it. In order to fully utilize the computing resource virtualization was used as a tool i.e. to partition a single piece of server machine into multiple small virtual servers. Now days, with the evolution of cloud computing, data centres use hardware virtualization to share the big data centre's hardware resources among multiple tenants in a transparent way.

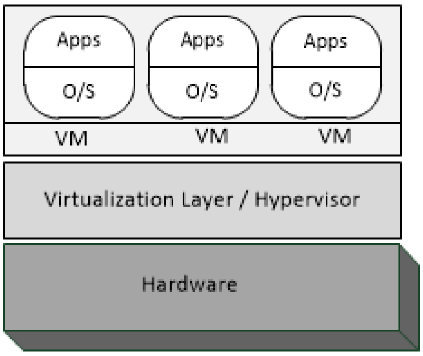


Figure 1: Layered Architecture of Virtualization

Enterprises acquire cloud services for their bulk data storage or running their systems on cloud instead of building data centres with expensive servers of their own thus reducing capital and operating cost. There are many advantages of virtualization. First it provides isolated execution environment for multiple operating systems to co-exist i.e. a system running Microsoft windows as host operating system may run Linux, Ubuntu, CentOS or any other operating system of user's choice on a guest virtual machine. Secondly, it aids in server consolidation i.e. instead of deploying separate expensive server machine in a data centre, one may run multiple server such as web server and file server on same server machine using virtual machines. Therefore virtualization results in increased hardware utilization and decreased capital and operating cost. Figure 1 shows the layered architecture where virtualization layer (virtualization software) runs directly on the hardware. Such virtualization software which runs directly on hardware is also stated as Type1 or bare metal hypervisor. Bare metal hypervisors are the far most used virtualization software in cloud computing environment and big data centres. Moreover multiple virtual machines are run and managed by virtualization layer. Each virtual machine has its own operating system and applications running in complete isolation from other VM's.

1.3. Hardware Virtualization and Virtual Machines

Hardware virtualization used by data centres refers to creation of virtual machines which run on a shared hardware with an operating system. Instead of making physical hardware characteristics visible to the user an abstract view of underlying hardware is shown to the user of virtual machine. A virtual machine (VM) is a software abstraction of the underlying hardware in which an operating system is installed and it runs virtually like a real machine with its own virtual hardware. In hardware virtualization, the machine which runs virtual machines on it is called host machine. Host machine may run one or more virtual machines on it called guest virtual machines. The virtualization software which runs on host machine is called hypervisor or virtual machine monitor. A hypervisor is responsible for creating virtual machines and their management like virtual machine's CPU resource allocation, VM memory and VM storage allocation etc. It manages the hardware resource sharing among multiple underlying guest VMs. Virtualization technology provides complete isolated execution environment to the virtual machines running on it.

1.4. Cloud Computing and Virtual Machine Migration

Cloud computing technology provides the computing resources such as applications, software, servers and network to the consumers over the internet. Cloud data servers use virtualization technology to provide an isolated execution environment to its consumers on a shared set of hardware resources giving them an illusion that they have a dedicated set of resources. In IAAS (Infrastructure as a service) model the cloud provider provides services to its consumers through provisioning of virtual machine. The virtualization software or hypervisor is responsible for creation, deletion, resource management and working of these virtual machines in cloud. Hypervisors support some state of the art feature such as virtual machine migration.

VM migration is the process of transferring the complete operating system that runs inside a VM along with applications running on it, from one physical location to other. Virtual machine (VM) migration is an administrative tool supported by many virtualization software or Virtual Machine Monitors (VMMs). For example XEN [17], VMware [18], KVM [19], Hyper-V etc. provide flexible migration and management of VMs. VM migration can be of many types including offline or cold, suspended and live VM migration. Live VM migration includes the transfer of VM's operating system and applications running on it from one physical location to other physical location while it is executing. During Live migration, applications running on being migrated VM might face varying downtime during final synchronization. In offline migration, VM is shut down or stopped at source, then sent over the network and resumed at destination. An abstract level view of VM migration is given in figure 2. In the figure, a VM (VM3) is being migrated from source domain to the destination domain. Possible reason could be whatsoever e.g. load balancing, power management or disaster recovery etc.

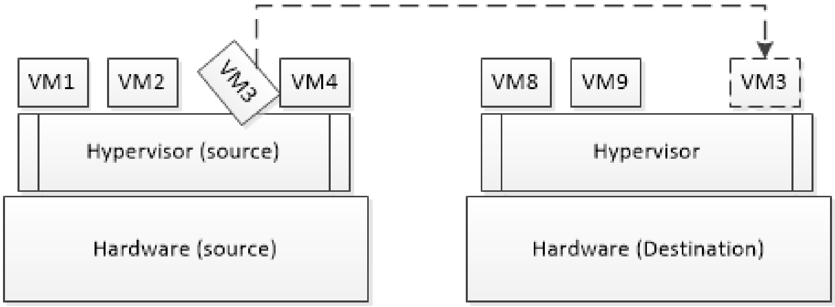


Figure 2: Virtual Machine Migration

Migration of VMs is a useful tool in data centres and cloud environments in which a virtual machine is migrated from one physical location to another for the sake of load balancing or in a scenario where a hardware failure is imminent i.e. a VM can be migrated from a server which is susceptible to some hardware failure to a stable server with free resources, so that even if the server stops functioning the services running on the VM do not face downtime. Similarly a data centre's admin may use the process of virtual machine migration for hardware maintenance. For example if he wants to bring down a server machine 'server 1' for maintenance running some virtual machines, he migrates the VMs running on that 'server 1' to some other 'server 2', so that VMs running on 'server 1' do not face a down time during maintenance. Furthermore, VM migration has a key role in load balancing scenarios of datacentres. In data centre's peak service hours a server machine may face a down time, in such scenario one or more VMs running on that heavy load server can be migrated to some other server with relatively lesser load. Hence, service of VM migration aids in load balancing, elastic scaling, fault tolerance, disaster recovery and easier hardware maintenance [13,14].

1.5. Requirement of Security Metrics

A system is stated as secure if no actual adversary can exploit it [33]. Generally system is made secure by deploying some security subsystem that serves to protect the system. This security subsystem is the integral part of a secure system. Organizations in general and security designers in particular often remain concerned about how much their system is secure. Information security performance measurements is also gaining interest due to number of regulatory requirements e.g. Government Performance and Results Act (GPRA) and Federal Information Security Management Act (FISMA) in particular require to measure information security performance [36]. These factors are pushing security designers to develop metrics for assessing the security of the system. However, developing enterprise level security metrics is a complex process and is listed as one of the hard problems in the Infosec Research Council's Hard Problem List 2005. It has been the main reason that why enterprise level meaningful security measures could not be developed. However, some security metrics have been successfully developed to measure the security of the specific attributes of components of security subsystem [40].

1.6. Research Methodology

Research is the process of systematic study of subject associated data sources and processes in order to conclude new facts and conclusions. Generally there are two approaches for carrying out the scientific research i.e. inductive research methodology and deductive research methodology. Inductive research methodology works from more specific to broader view of the problem and is also known as bottom up approach. Deductive research methodology works by narrowing down from general to specific view also known as top down approach. In this research work the deductive research methodology is used for carrying out the research work. A hypothesis was made about secure migration of virtual machine which was examined by performing extensive literature review on the problem. Observations resulted in design of a protocol for secure VM migration and quantitative security metrics for its evaluation. The designed protocol was verified using formal verification method. The protocol was further evaluated through devised security metrics against different adversary models.

1.7. Scope and Motivation

According to NIST Interagency Report, Directions in Security Metrics Research 7564 [35], evaluation of security effectiveness is usually performed through reasoning rather than direct measure of system's components however security metrics provide a systematic way to measure performance of security controls. Therefore this research work proposed a relative quantitative security metric model which is used to devise three security metrics such as efficiency, effectiveness, and cost benefit measure for evaluation of security controls of designed protocol. Security evaluation results are obtained via real values obtained from applying attack models i.e. Dolev-Yao Attack Model & KCI Attack Model on proposed protocol and using these values as inputs to devised metrics. The target of assessment for devised security metrics is the proposed secure VM migration protocol. VM migration and security metrics for information security both are emerging technologies this thesis works around these two domains.

Area of Secure migration of virtual machine is recently capturing attention. Many of the hypervisors which support this process do not include a comprehensive solution for securing this process [13][20]. Businesses are increasingly acquiring cloud services using IAAS (Infrastructure as a Service) service delivery model by provisioning of virtual machines. In order to satisfy the concerns of enterprises acquiring cloud services and providing them with flexibility of migrating their virtual machines securely, it has become crucial to develop some uniform security scheme along with a negotiation protocol that deals with security issues of virtual machine migration in cloud environment. As VM migration involves sending critical infrastructural information over network, therefore, VM migration involves number of security challenges. For example unencrypted traffic may result in exposing machine states, secret keys and passphrases [20]. Similarly unauthorized VM migration may result in VM to be migrated to a platform under the control of attacker. Moreover, lack of mutual authentication may also result in same kind of attacks i.e. man in the middle attack. In this regard, we designed a protocol for secure virtual machine migration that preserves confidentiality, authenticity and integrity of virtual machine before, during and after transit; both on source and destination platform. This proposed protocol was made target of assessment for devised security metrics and efficiency, effectiveness and cost benefit measure of the security controls was measured using devised metrics.

As shown in figure 3, the scope of thesis includes the detailed study on security aspects of virtual machine migration and finding out a set of security requirements which must be met in order to securely migrate a virtual machine to its destination domain. Later a detailed architecture is proposed for secure migration of VM. Afterward, the process of formal verification of the designed protocol is performed.

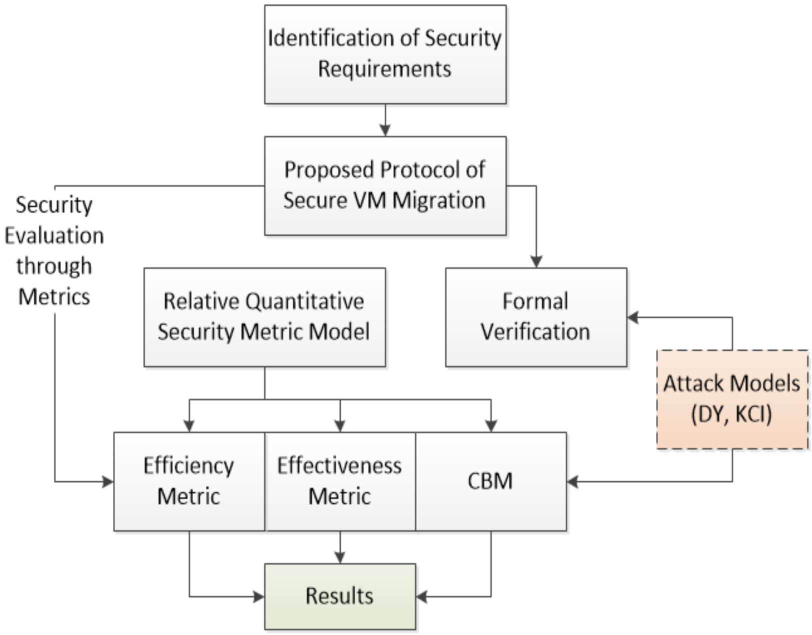

Identification of Security Requirements
Proposed Protocol of Secure VM Migration
Security Evaluation through Metrics
Relative Quantitative Security Metric Model
Formal Verification
Attack Models (DY, KCI)
Efficiency Metric
Effectiveness Metric
CBM
Results

Figure 3: Thesis Scope

In literature no work has been done so far which measures the security performance of VM migration protocols. In this regard a relative quantitative security metric model has been proposed which is used to further derive three metrics for measuring efficiency, effectiveness and cost benefit measure of VM migration protocols. The results are given for three different security states of protocol against two different attack models.

1.8. Problem Statement

Information security regulations and standard guidelines require devising quantitative security metrics for measuring the performance of security controls. However no such security metrics are present which measure efficiency, effectiveness or cost of security protocols. Like all other security areas, there is a need to develop security metrics for secure VM migration protocols also as no such security metrics exist already.

1.9. Research Contribution

A detailed study on process of virtual machine migration was carried on and many weaknesses along with potential attacks were identified. In the light of identified threats and weaknesses a detailed design and architecture is proposed for secure migration of virtual machine. A conference paper with title “A Secure Architecture for Inter-Cloud Virtual Machine Migration” authors Tayyaba Zeb et al., [46] is published in 10th International Conference on Security and Privacy in Communication Networks (SecureComm 2014). In this conference paper a secure architecture for inter cloud VM migration is proposed. The proposed approach provides the authenticated and authorized migration of virtual machine from source cloud domain to destination cloud domain. Both source and destination domains mutually authenticate each other and validate migration request. This helped to avoid unintended migration of VM to some malicious destination under the control of attacker. Similarly this also helped to avoid unintended malicious VM potentially with rogue applications to be received on a legitimate destination. The mutual authentication of source and destination cloud domain is based on Federal Information Processing Standard, FIPS PUB 196 i.e. Authentication Using public key cryptography. The domains must have acquired X.509 certificate from trusted Certificate Authority. Confidentiality and integrity of VM data is achieved by applying Advanced Encryption Standard (AES) and SHA-512 respectively. The scheme presented in this paper also provides the non-repudiation service. Each of the domains presents the signed ticket containing digitally signed request/response with the domain's private key.

1.10. Thesis Organization

Rest of the thesis is organized as follows. Chapter two provides a literature survey regarding virtualization, virtualization software, forms of virtual machine migration and security metrics. It specifically discusses the security issues related to hypervisor, virtual machine migration and existing solutions for secure migration of virtual machines. The chapter concludes with limitations in existing techniques identified in the light of literature review and need of security metrics for security controls evaluation. Third chapter is about proposed architecture. The chapter initially describes the security requirements for virtual machine migration. Later it explores the different components of the designed system and proposed protocol for secure migration of virtual machine. Chapter four explains that how the process of formal verification for the proposed protocol is carried out while chapter five describes proposed security metrics model for evaluation of designed protocol. Chapter six concludes this work by giving conclusion and future directions.

Summary

This chapter provides an introduction to security metrics, virtualization and explains the basics of virtual machine migration. It then describes the scope of thesis work and motivation to carry out research in the field of secure virtual machine migration and its evaluation through quantitative metrics. It also explains the need of security metrics for evaluation and measurement of security controls. Furthermore it elaborates the problem statement and research contribution. At the end of this chapter thesis organization is described.

Chapter 2: Literature Review

Literature Review

This chapter includes the literature work that has already been done in the area of security metrics and secure migration of virtual machine. The related work section is divided into five categories. Initially security metrics for evaluation of security sub systems and its different types are discussed. Second section describes the process of VM migration its forms and functionality issues including VM migration within data centres and across data centres. Later a detailed study of attacks on VM migration process along with security approaches used by community for securing VM migration is provided. The chapter concludes with limitations of existing techniques.

2.1. Security Metrics

According to NIST Special Publication 800-55 Rev.1 [36], information security programs must measure their performance through security metrics. A security metric is a computable function that tells about the security extent of the system. The aim of the system evaluation through metrics is to have a measureable level of confidence about how much the system is secure. According to the system under discussion three security metrics which match the security specifications are selected from literature [33] to find out the system's security level and to reach to the desired security goal. These metrics are as under:

- Weakness based Metric
- Threat based Metric
- Protection based Metric

Weakness based Metrics: In weakness based metric the weaknesses of the system are identified and then system is assigned a security level/number on the basis of rectified weaknesses. Weaknesses that weakness based metrics takes as input may be divided into i) known weaknesses and ii) unknown weaknesses. Since unknown weaknesses set consists of unknown term and thus are not measurable so known weakness is the only term that can possibly be included in this metrics.

Threat based Metrics: A threat based metric takes estimates of possible threats to the system from an active adversary. It takes into account following two features, i) Adversary capabilities and ii) known weaknesses of the system that an intruder can exploit. To quantify the adversary capability there come many attack models which can be applied on the system to verify the security level of the system such as Dolev-Yao model [25], CK model, eCK[34] model and Actor Key compromise model etc. The attack model defines capabilities and restrictions an attacker has. Secondly, there are exploitable weaknesses that an attacker can exploit to cause damage to the system. These attack models also help in finding out those weaknesses of the system which might not be obvious but an attacker can exploit to do the damage to the system.

Protection based Metrics: Protection based metrics provides a relative sense that how much our system is protected. Classically, protection based metrics can be based on quantitative measure such as cost to implement the protection system or on qualitative measure such as the certification level or ranking assigned by some external body. This research work proposed two new protection based metrics that are efficiency of the security control and the effectiveness of putting that security control in the system.

2.1.1. Metrics for Security Sub Systems/Protocols

Security metrics for security evaluation of security sub systems is an emerging domain and research community is putting emphasis on it. Vaughn et al., [45] has given a classification of different information assurance/security metrics. The author has stated metrics which measure the security of technical objects, processes and protocols as TTOA metrics (Technical Target of Assessment). The authors have further categorized TTOA metrics into metrics for strength assessment and metrics for weakness assessment. The first category i.e. metrics for strength assessment takes into account two scenario where security is assessed with and without presence of adversary. Moreover, metrics for weakness assessment take into account measures such as threats, risks and vulnerabilities.

M. S. Ahmed et al., [44] have given quantitative security metrics based solution on the subject of network security. The authors have proposed security metrics framework for measuring network security. The authors make use of three measures such as existing vulnerabilities measure, historic vulnerabilities from National Vulnerability Database (NVD) and probabilistic vulnerability measure to evaluate network security. Attack propagation metric is devised using vulnerabilities measures to assess that how an attacker can cause an attack to propagate in the network exploiting services vulnerabilities. As a result of proposed metrics different security policies can be compared in order to evaluate which policy provides better network security. However the approach adopts apparently complex scenario for measures while security metric guides recommend that the measures should be readily available. Moreover the approach also assigns a protection level of one to firewalls and between 0-1 to IDS (Intrusion Detection System) qualitatively as opposed to quantitative measures mentioned in title.

Lemay et al., [46] provides a model based approach for generation of security metrics intended for a particular system. The approach takes adversary information, system information and desired security metrics as input. The model is implemented as a tool which generates attack execution graph using system information and attackers profile showing the possible ways an attacker can execute an attack on system. The proposed model gives the security level of a system against particular adversary. The proposed model however assumes that a system is already aware of types of adversaries and their capabilities.

H. Wang et al., [49] has proposed security metrics for software systems. The measures for formulated metrics are taken from CVE (Common vulnerability Exposures) and CVSS (Common Vulnerability Score System). The metric takes product of weakness severity and its associated risk as input. As a single weakness may result in multiple vulnerabilities so severity of weakness is calculated as sum of severity of all vulnerabilities associated to that weakness. Severity of vulnerabilities is taken using scores from CVSS while risk is calculated as ratio of frequency of occurrence of that vulnerability to a span of time taken in months. The results are taken against three different web browser applications. The author has provided the conclusion with quantitative measures that more the number of vulnerabilities a software application has the lesser security it provides.

In literature a number of approaches are being proposed for secure VM migration and its performance metrics in terms of migration delay, response time, downtime etc., however, no work has been done on measuring the security performance of security controls. For example, William et al. [8] provides the cost of virtual machine migration in cloud. The authors have performed series of experiment with workload of modern internet applications on a set of virtual machines. They showed that process of VM migration with a workload of overly subscribed application with around 600 users exhibits a downtime of around three seconds. The objective of paper was limited to find out the performance metrics i.e. down time experienced by virtual machine during live migration.

2.1.2. Metrics Formulation for Security Measurement

It is reported that more than ninety per cent of security incidents are result of flaws at design and coding [49]. Therefore, performing the assessment of the system security at early level i.e. design stage helps in reducing exploitable future vulnerabilities. Quantitative metrics are the tools to measure and assess the fulfilment of security objectives in a precise manner. In literature most of the work has been done on qualitative metrics however they tend to be less precise as they only assign a ranking of bad, good or average to the system which can be misleading sometimes. A lot of emphasis has been put together by standard and regulatory bodies to measure the security performance of security sub systems using security metrics. For example, NIST Direction in Security Metrics Research [35][36] recommends devising quantitative security metrics for measuring the efficiency and effectiveness of security controls. However no metrics has been formulated so far which measure the efficiency and effectiveness of communication and security protocols specifically for secure VM migration. As with the advent of new applications and their changing requirements, new protocols are also being proposed to secure their communication. Therefore, it has become necessary to formulate security metrics which measure the extent to which security objectives of these protocols are being met. A number of approaches are proposed for measuring the security using quantitative metrics in the field of network security [44], application security [49] and code inspection [50], however, to the best of our knowledge currently no metrics are there which quantitatively measure the security performance of the security protocols and specifically secure VM migration protocols. A well understood classification, taxonomy and nomenclature is present in literature for security metrics [39][45] such as technical vs. management level metrics, component vs. enterprise level metrics, efficiency and effectiveness metrics, however, little attention is paid on actual formulation of these metrics. Hence our contribution in this regard is the formulation of these quantitative metrics and further evaluation of secure VM migration protocol using formulated metrics. The efficiency metric formulated in this research work provides the efficiency of secure VM migration protocols in terms of their resilience against number of attempted attack. Furthermore lack of proper quantitative security metrics for effectiveness makes it harder to perform the comparison of different alternative security solutions. The relative quantitative security metric model presented in chapter 5 allows to relatively compare the effectiveness of two alternative protocols with similar security goals and objectives. However approach presented in this work compares the two different states of the same protocol to check their relative efficiency, effectiveness and cost benefit measure.

2.2. Virtual Machine Migration

Virtual machine migration involves sending complete operating system and application running on it from one physical location to other. This process seems complicated however in the presence of a hypervisor or virtual machine monitor things are not that convoluted. A process of virtual machine migration consists of two forms: i) Live VM migration and ii) offline VM migration. Live VM migration involves sending a VM along with its operating system and applications running on it from one physical location to other when its executing. While in offline VM migration a VM is paused at source machine, sent over the network and resumed at destination machine after its complete transmission.

2.2.1. VM Migration within Data centre

VM migration is supported by most of the current hypervisors including XEN, Hyper V, KVM and VMware within data centre over Local Area Network (LAN). Live VM migration involves sending three main physical resources of the migrating VM [1]. They include memory, disk and network. In scenario of LAN (local area network) or within a data centre local disk storage migration is not required because data centres usually use (NAS) Network Attached Storage and all VM’s use this disk storage as single point of storage for disk images. Second main issue in live VM migration is the transfer of active network connection. In case of LAN or a single data centre it is done through sending an unsolicited ARP reply to the network about the new location of migrated VM's IP [3]. As a consequence, live VM migration within a cloud's data centre mainly consist of continuously copying memory state of migrating VM from source to destination server machine[2]. VM memory migration is divided into three phases [1] which include i) Push Phase, ii) Stop and copy phase, and iii) Pull phase.

In push phase, the migrating VM starts sending its memory pages on destination machine while continuing its running on source domain. After a certain time it stops its execution for a very short duration, copies its pages on destination and starts execution on destination domain. This phase is therefore called stop and copy phase. In final pull phase if the VM executing at destination machine faces a page fault i.e. a page access which is not copied yet, the VM copies this page across the network from source domain. Although the process seems complicated but hypervisor implement Post copy or pre copy algorithms to migrate VMs and makes it easier for admin to perform the task. Both of the above mentioned algorithms (Post copy and pre copy) use only two phases from pull, stop and copy, push phases in their implementation. A brief preview of post copy and pre copy is given as follows.

Pre copy algorithm uses push phase along with stop and copy phase for VM migration. It is iterative in nature such that it copies memory pages iteratively to the destination using push phase. This iterative nature is due to dirty pages i.e. the pages that get modified very frequently. The push phase is followed by stop and copy phase in which VM is stopped for as little time as possible. Post copy algorithm uses stop and copy phase first. It stops the VM at source, sends as many memory pages as possible, resumes the VM at destination. After starting execution at destination phase, migrated VM starts fetching the memory pages from source machine. Both approaches have their own advantages and disadvantages. In pre copy dirty pages get to be sent repetitively while in post copy memory pages are at most transferred once. Many hypervisors like XEN, VMWare, KVM etc. use these algorithms for VM migration. XEN for example uses Pre copy for VM migration [4].

2.2.2. VM Migration across Data Centres

Current open source hypervisor implementations provide VM migration within a single data centre and limit its scope to LAN environment only [6]. The migration of virtual machine over (WAN) Wide Area Network among distant data centres or whether it is inter-cloud VM migration, remains a desirable feature. A cloud provider may require it in order to cut down the cost by combining multiple smaller sites into single large data centre [3], thus moving the VMs over WAN flexibly. A cloud user i.e. an enterprise may require it if it finds cost benefit with some other cloud provider. The research community is putting a lot of focus on extension of VM migration over WAN [3,5,6]

Across data centre VM migration involves sending memory, disk state and redirecting network connections to the destination location. Memory pages are sent using same procedure as described in previous section but disk state transfer and network redirection are also not trivial. It is because transferring a VM's disk state over WAN may require tens of Giga bytes of data to be transferred on low bandwidth and high latency links. Secondly In live VM migration active network connection needs to be redirected such that applications running on VM do not loose active connections and VM is transferred in a way transparent to applications.

Franco et al. [5], discusses the analysis of the processes that allow the live migration of VMs over long-haul networks. The author also explains how VMs can be migrated across geographical distances transparently to applications. Live migration of VM is done over dedicated light paths of 1Gbps capacity among distant sites like Amsterdam and San Diego. An agent named (VM traffic controller) VMTC is responsible for maintaining connectivity with VM's destination domain. VMTC communicates with an entity 'AAA' (Authentication, Authorization, and Accounting) for authorization in order to use the dedicated light path. The AAA entity grants VMTC with a token to start a set up request for available light path. VMTC when receives the confirmation of available light path starts VM migration command. In order to provide seamless connectivity to the applications running on the VM to be migrated, VMTC configures an IP tunnel. The authors state that a long haul migration occurs across multiple domains (source, destination and intermediate domains) with a limited trust but does not provide any solution for trust establishment. Secondly, solution requires 1Gbps (a guaranteed service) capacity link which is not suitable or possible in most of circumstances. Moreover, the authors discuss security in term of resource allocation. Resources are granted to those claimers who present a security token to avoid resource theft. AAA entity (Authentication, Authorization, and Accounting) is responsible for authorizing the use of light path. In the form of AAA entity, paper provides Access control feature for accessing dedicated light path.

Timothy et al. [3], presents the problems that are associated with live migration of virtual machine over wide area network. Author describes the key difference between LAN based VM migration and WAN based VM migration stating that VM migration on LAN only requires moving memory state because disk state is shared on network attached storage therefore, it evades moving disk state. Whereas WAN migration becomes challenging because it requires moving both memory as well as disk state. Secondly in LAN VM migration remains transparent to the running applications with active connections because IP address remains unchanged. On contrary WAN based VM migration require coordination with network routers in order to keep network connection of migrating VM alive. The paper proposed an architecture named CloudNet which implements a set of optimizations which reduce the cost of sending VM memory and storage over low bandwidth and high latency links. The CloudNet platform uses DRBD a synchronous replication mode to reduce the volume of disk storage to be migrated. The use of a synchronous replication reduces the effects on performance in bulk data transfer. The CloudNet platform uses a set of optimization techniques to optimize the memory transfer such as smart stop and copy minimizes the unnecessary iterations. Using page delta optimizations, if a page needs to be resent only the 32 bit index to that page and page delta value is sent again instead of sending whole 4 kb page. The problem of redirecting active connections of applications while migrating a virtual machine from an enterprise to a cloud is discussed in [3]. The CloudNet platform developed by authors uses VPLS (Virtual Private LAN Services) which bridges the VLANs at Cloud and the Enterprise; enabling open network connections to be seamlessly redirected to the VM's new location.

The author made changes in Xen hypervisor to solve performance issues in migration over WAN, however, it does not consider the software vulnerabilities that may arise due to change in module such as memory safety violation i.e. buffer overflow. The optimization technique and algorithm although helped to reduce the bandwidth issue and reduced pause time of VM during migration, but it increases the CPU overhead due to excessive processing such as taking hash of each page to be sent. The authors use layer 2 VPN's for security perspective thus protecting transmission channel. Samer et al. [7] provide optimization techniques i.e. data de-duplication for a group of migrating VMs. The author discusses the challenges of migrating a set of virtual machines stated as VMFlock between the data centres and across clouds. The said scenario may be needed when an application setup running on more than one virtual machine requires to be migrated. To accelerate the process of migrating flock of VMs the optimizations such as data de-duplication is performed to minimize the duplication of data sent over the network. The author proposes the deployment of designed approach as a virtual appliance in cloud.

Pierre et al. [6] provides approach for VM migration over WAN considering the fact that multiple VMs running identical operating systems have considerable amount of identical data. Shrinker, the approach used by [6], leverages this identical data to achieve transmission efficiency between data centres over WAN. The approach uses content based addressing to identify the pages that are already present on destination, thus reducing the number of pages sent over WAN. The approach is implemented in KVM hypervisor.

2.3. Related Security Issues

Cloud computing offers services to their customers using three service delivery models named as SAAS (Software as a Service), PAAS (Platform as a service) and IAAS (Infrastructure as a Service). In Infrastructure as a service model the cloud provider offers complete infrastructure to its consumer including applications, servers and storage. In IAAS delivery model the cloud provider provides services through provisioning of virtual machines. CloudStack and OpenStack are two open source cloud platforms with IAAS delivery models. CloudStack is open source cloud operating system product for deployment and management of (IAAS) Infrastructure cloud. It supports variety of hypervisors such as KVM, XEN etc. Both of the mentioned hypervisors expose VM data during virtual machine migration. However, CloudStack itself provides encryption for network traffic which is enabled by setting one of its global encryption parameter as ’true’ thus providing confidentiality to VM migration traffic indirectly. One more constraint by CloudStack is that it restricts VM migration to local area network only. Moreover, there are many approaches proposed in literature which focus on providing secure cloud architecture, however, we keep our focus on study of those approaches that are related to virtualization or hypervisor security and then narrow it down to VM migration security.

2.3.1. Security Issues of Hypervisors and Virtual Machines

Hypervisor – the virtualization software is a key component in cloud infrastructure. In cloud service delivery models, the IAAS (Infrastructure as a Service) delivery model provides services to the customers through provisioning of virtual machines. Hypervisor is responsible for creation and management of virtual machines. There are two basic types of hypervisors:

- Type I hypervisor
- Type II hypervisor

As shown in figure 4, Type I hypervisor are those which run directly on raw hardware. i.e. the virtualization software is directly installed on hardware. They are also called bare metal hypervisor e.g. XEN, Hyper V etc. Type II hypervisors are those hypervisors which are installed on a host operating system therefore, called hosted hypervisor e.g. oracle box and VMware. The hypervisor or virtualization software runs as a process in Type II hypervisor. Type I hypervisors are mostly adopted in cloud environment because they give higher efficiency and performance as they can communicate directly with hardware.

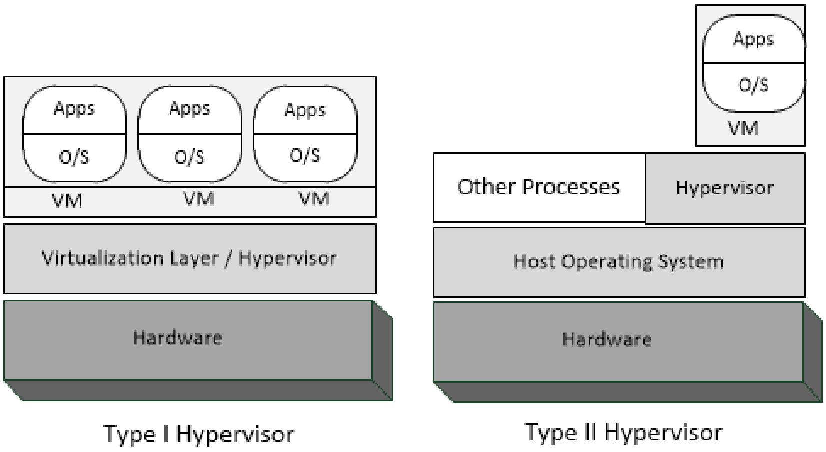


Figure 4: Type I and Type II Hypervisor

Hypervisor security is directly connected to the cloud security because if an attacker gets to compromise virtualization layer, he can get control over the cloud infrastructure e.g. migrate VMs to a destination under the control of attacker. As virtualization has its own benefits like server consolidation, load balancing, power management, elastic scaling etc. at the same time it becomes vulnerable to (VMBRs) VM based Rootkits like Subvirt [9] and malware like Blue pill[10]. SubVirt was virtual machine based rootkit that used to infect system invisibly. It changes the system's boot sequence to avoid its detection by operating system. Similarly blue pill malware was a hypervisor based root kit which made an ultra-thin layer of hypervisor on targeted operating system and get control over it. In order to cope with such security issues in virtualization environment many security solutions are proposed by research community. Some of them are given as under:

Advanced cloud protection system (ACPS) provided by [11] is integrated into virtualization software (virtual machine monitor) to monitor the integrity of guest VMs. In ACPS, the host-side database (Checksum DB) contains computed checksums for selected critical host infrastructure and guest's kernel code, data, and files. Integrity of cloud components is checked via logging and periodic checksum verification of executable files and libraries. The system responds to the warning according to the set policy. ACPS works well against network probing, root kits, weak passwords etc. As it uses IP tables for detection mechanism it cannot detect distributed denial of service attacks. It provides integrity of VMs from possible malware, viruses and rootkits. The paper provides a good approach to provide security with the help of virtual machine monitor. The approach resembles intrusion detection system functionality. The modules are implemented at host level but the author does not specify that how to cater if a VMM itself gets compromised by an attacker.

An exclusive aspect of increasing VM security in cloud environment is presented in [12]. A combined approach that checks for software updates and scans virtual machines for known security vulnerabilities is presented. Update checker is capable of checking software update requirement in both active and dormant VMs. Online penetration suite finds any possible vulnerability in software packages and generates results in the form of a report. Vulnerability scanners used by penetration suite are although third party products but the two modules presented in the paper help in increasing security by alarming against any un-patched or vulnerable software. The paper provides an organized approach towards keeping the VM up to date and avoids retaining any un-patched software. Although auto update utility in MS Windows provides the similar functionality but the paper provides extra feature such as vulnerability assessment and a proper report generation even in the machines which are dormant and not so active for a long period. Providing basic security was not the scope of paper so it discusses only vulnerability assessment.

2.3.2. Security Issues specific to Virtual Machine Migration

Virtual machine migration is an administration tool supported by many Virtual Machine Monitors (VMMs) i.e. XEN, VMware, providing flexible movement and management of VMs. In distributed computing environment such as cloud computing, VM migration allows transfer of complete operating system and applications running on it thus aids in load balancing, elastic scaling, fault tolerance, disaster recovery and easier hardware maintenance[13][15]. As VM migration involves sending critical infrastructural information on network, therefore VM migration involves number of security challenges. The area of secure VM migration is recently getting attention. In literature, a few solutions are proposed regarding different aspects of security issues related to the VM migration process but first a slight discussion on how vulnerable a process of migration is and possible attacks in literature on VM migration are given as follows.

Attacks on data and control plane of migrating VM are categorized and implemented by Jon Oberheide et al. [13]. Authors demonstrated that integrity of data can easily be harmed during migration. Attacks on VM are categorized in three categories [13] i.e. attacks on Data plane, Control plane and attacks on Migration Module. Data Plane is the communication channel on which VM migration data is transmitted while Control plane includes the communication mechanism employed by hypervisor to initiate the process of migration. The authors state that lack of security against control plan may result in attacks such as malicious outgoing/incoming migration requests. Moreover, large number of unsolicited VM migration requests by attacker may result in Denial of Service attack. Similarly false resource advertising (i.e. in Auto load balance enabled environment) is also a possible attack against control plane. In auto load balance enabled environment a VM automatically gets migrated to a host which has extra available resources. In such environment an attacker may take advantage and advertise false resources in order to get a victim VM migrated to the host under his control. Similarly on data plan passive sniffing and active memory manipulation is performed by author as proof of concept. The paper showed that XEN hypervisor entirely exposes migrating VMs data during transit. The author also indicated the loopholes of migration module in VMM that can cause Buffer/stack/heap overflow attacks. In order to avoid such attacks use of type-safe language is suggested for development of VMM. The authors have empirically demonstrated proof of concept by developing a tool, Xensploit and showed how these attacks can be applied on Xen and VMware. The author has described how certain functions in Xen source code can be exploited by attacker to cause heap/stack/buffer overflow. The author has also identified how easily integrity of data is harmed during migration however it does not specify any solutions for it which drew our major inspiration to devise a secure protocol for VM migration.

The author also demonstrated simple memory manipulations against the VM states of migrating VM during live VM migration hence proved the need of an immediate rectifications and proper security system for this critical process. In figure 4 a possible attack scenario on process of VM migration is given [13].

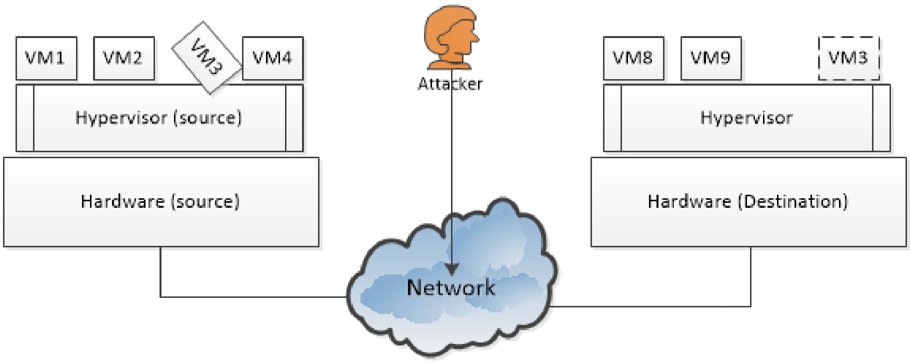


Figure 5: A possible Attack Scenario on VM Migration

As shown in figure 5, a VM is being migrated from source domain to destination, if VM data is not protected, an attacker may easily launch Man in the Middle attack. At a minimum, he can capture the network traffic resulting in disclosure of sensitive information to attacker.

2.4. Existing Security Solutions for VM Migration

Security issues in VM migration are being studied in recent couple of years. A few protocols are proposed by researchers for secure migration of VMs. Security issues regarding the protected processes running inside a VM are discussed in [14]. Authors enforce the security when VM is migrated over the network providing confidentiality and integrity of protected process. A VM cannot be straightforwardly migrated if there are protected processes inside. It is required that the protection strength is not lowered during and after the relocation of a VM. The paper introduces security modules such as migration data protection module that encrypt the protected process pages before sending it to control VM or on network. The system also provides integrity of protected process data and prevention from replay attacks by taking hash and checksums. They have implemented a prototype system called PALM (Protection Aegis for Live Migration of VMs), which is based on Xen VMM and measures the performance of PALM with server application benchmarks. The encryption applied to only protected processes should have been applied to all memory pages for confidentiality and security reasons but scope of paper is limited to protected processes only.

The author presents security and trust issues for the process of VM migration in [15] i.e. platform authenticity and confidentiality of VMs data during transit. For platform authenticity, authors introduced PTAA (Platform Trust Assurance Authority) which assigns trust levels to platforms based on their configurations and provides them with a trust credential 'trust-token' which they exchange before migration. PTAA gives the authenticity that VM is being migrated on a trust worthy platform. For confidentiality a VM is encrypted with a symmetric key before migration.

In this paper the author presents the idea of PTAA which is a traditional approach and is similar to certificate authority. The solution is based on TPM bind key which introduce hardware dependency and increased overhead due to PTAA (platform Trust Assurance Authority). The author initially presents the idea of VM migration audit by end user but does not discusses the solution in detail. Moreover, Trust Assurance Level (TAL) value assigned for a particular software configuration may be outdated or false after a software patch. As cloud is a big infrastructure its software and hardware configuration might change frequently, so after every update or change it could potentially require a new trust-token from third party. In this scenario, TAL value assigned to a particular software configuration may frequently be outdated or become false after a software patch.

A TPM based VM migration protocol using virtual TPM (Trusted Platform Module) is presented by Boris et al. [16]. Authors presented a hardware based protection system which provides information protection and software authenticity in private clouds. The solution creates a hierarchy of TPM keys that are migrated along with the migrating VM which might cause the protection level to degrade as TPM's security relies on its non-migrate-able keys. In both of the above mentioned approaches[15][16], the protocols work only if the infrastructure has TPM support, thus introducing the hardware dependency which is not suitable feature in all data centres with legacy hardware. Moreover the solution given by [16] works for suspended VM or in other words it supports offline VM migration only.

The process of live migration of virtual machine using KVM (Kernel based Virtual Machine) was carried out in [20]. The authors state that KVM and Xen expose entire machine state i.e. operating system kernel and application's state during the process of VM migration; however, they do not provide solution for it. Jyoti et al. provide a survey on security schemes proposed by research community for secure VM migration [21]. The mentioned schemes use different security solutions like firewalls, TPM (Trusted Platform Module) and virtual LAN etc. for securing VM migration. The author however concludes that no approach provides a comprehensive solution to the migrating VM issues. Furthermore, Chen et al. [22] provide a framework for VM migration over local area network stated as network security enhanced hypervisor. The solution is based on network security engine i.e. state full firewall. The module implemented by author named SCMA (Security Context Migration Agent) is responsible for migrating VM from source to destination after checking security context of each packet.

2.5. Limitations of Existing Techniques

Most of the existing work for VM migration is focused on following two areas. First area is the optimization techniques for reducing the redundant disk data in VM migration over WAN to achieve better transfer performance over low bandwidth and high latency links. And the other area is the approaches that deal with the transfer of the active network connections of VM over Wide Area Network (WAN). However, area of secure VM migration is recently getting attention. Most of the existing solutions for VM migration security are either TPM based and fail to work with legacy hardware, or they cater VM migration security issues individually. As mentioned in [13][20][22], XEN and KVM hypervisor expose the entire machine state during migration. A nearby attacker can easily capture the network traffic and perceive the internal states of migrating VM (keys, passphrases) etc. Even process of VM migration carried out using one of the security features such as encryption, provides confidentiality of data but its security may potentially fail if other security features are absent such as mutual authentication and data integrity. For example, lack of mutual authentication may cause VM to be migrated to a platform under the control of an attacker, even if VM was encrypted during transmission. Similarly lack of encryption may result in even more worst results.

Table 1: Security Features provided by Existing VM Migration Protocols

Security features/Sec. Approaches	CloudNet[3]	PTAA[15]	PALM[14]	VM-vTPM[16]	NSE-H[23]
VM Migration Confidentiality	Yes	Yes	Partial	Yes	No
Integrity of VM migration	Yes	No	Yes	Yes	No
Mutual Authentication	No	Yes	No	Yes	Yes
Non-Repudiation	No	No	No	No	No
Platform Trust	No	Yes	No	Yes	No
Identity Protection	No	No	No	No	No
Evaluation of Security Controls	No	No	No	No	No

The table 1 shows the security features provided by different VM migration protocols in literature. The major security features for VM migration, inferred through literature review are mentioned in the table 1. First feature mentioned is VM migration confidentiality i.e. VM data remains confidential in transit during VM migration. Second feature is VM's data integrity, the integrity checks are applied so that VM data during migration remains unchanged or at least any unwanted changes get notified so that data can be retransmitted. Third security feature is mutual authentication among source and destination domain of the migrating VM so that VM does not get migrated to a spoofed destination i.e. destination under the control of attacker. Fourth security feature is of non-repudiation i.e. the receiving node or destination of VM cannot deny that it has received the VM. Similarly sending node or source domain cannot deny over the fact that a VM is migrated by it. The second last security feature is of platform trust. This security feature is provided mostly by those approaches which use Trusted Platform Module (TPM). The TPM chip binds system configuration with its keys, and keys are retrieved only when the configurations are not altered, in this manner the owner of a domain cannot lie about the properties and configuration of the system. The last feature mentioned in the table is evaluation of security controls. In the field of information security it is a general understanding that security performance of the adopted security controls should be measured however in literature no such approach exist which performs the evaluation of the security controls.

In the table given above two of the approaches including PTAA[15] and VM-vTPM[16] provide platform trust through the use of TPM chip. The authors provide platform trust for hardware and software configuration through use of crypto processor. These approaches however introduce hardware dependency. The PALM[14] approach mentioned in the table partially provides confidentiality because it is applying encryption to protected processes running inside a VM only. Similarly NSE-H[23] covers mutual authentication property only. Moreover CloudNet[3] approach provides data confidentiality and integrity for VM live migration only.

Summary

This chapter starts off with a detailed description of different types of security metrics for evaluation of the security protocols. It then provides in depth understanding of how VM migration is performed within and across data centres. Furthermore it discusses the related security issues including security issues in virtual environment and the security issues specific to VM migration. The chapter concludes with discussion on limitations of existing security techniques.

Chapter 3: Proposed Architecture

Proposed Architecture

In this chapter the detailed design and architecture for secure virtual machine migration across inter cloud domains is described. Initially a set of security requirement are defined which must be met in order to securely migrate the VM from source cloud domain to destination cloud domain. Secondly the designed system's components are discussed and then detailed design and message exchange are described. The chapter ends with description of performance modelling for the proposed protocol.

3.1. Security Requirements for VM migration

After a deliberate review of literature and finding out the limitations of existing techniques, following security requirements were considered while designing our proposed solution:

- Mutual Authentication of source and destination domain
- Confidentiality of VM in transit
- Integrity of VM in transit
- Non-Repudiation for VM migration
- Cloud and User Identity Protection
- Logical error detection in Migration Protocol flow

In literature review chapter the security requirement for secure migration of virtual machine are analysed, and it is identified that lack of single security feature may arise many other vulnerabilities in the process of VM migration. The emphasis is put on complete secure architecture because a single security feature i.e. mere encryption might not help. For example, a virtual machine being migrated to destination with full encryption may fail to provide security if the VM get transmitted to some rogue domain under the control of attacker due to the lack of mutual authentication among participating domains. The approach presented in this thesis work attempts to cover all above mentioned security requirements as a single comprehensive solution.

3.1.1. Mutual Authentication of source and destination domains

In the process of virtual machine migration source domain is the one which is currently hosting the virtual machine and wants to migrate it while destination domain is the domain which is the intended receiver of the virtual machine. Mutual authentication of source and destination domain is necessary because lack of this property may result in many attacks possibility, Such as virtual machine being migrated to rogue destination. Secondly there arises clogging attack i.e. a malicious node may continuously initiate large number of migration requests, the destination domain if reserves some resource after receiving a request may suffer through un-utilised reserved resource and thus may not be able to entertain legitimate user request.

3.1.2. Confidentiality of VM Data in transit

Most of the hypervisors responsible for VM migration expose the VM data during transmission over network. An active intruder can easily intercept the migrating VM's traffic and disclose the data that may include secret information like secret keys, passphrases etc. The confidentiality property of migrating VM is required so that classified information of virtual machine is not disclosed to the adversary.

3.1.3. Integrity of VM Data in transit

Jon Oberheied et al. [13] performed memory manipulation for migrating virtual machine's data and showed how easily the integrity of data can be harmed. He showed that by inserting a test stream into running virtual machine during migration. Therefore integrity checks for VM migration are mandatory so that if some modification is detected, VM data could be retransmitted.

3.1.4. Non-Repudiation for VM migration

Non repudiation is the property which prohibits the sender or receiver of a transaction to refuse from it after having performed it. This property is helpful in VM migration so that receiver of VM could not refuse from having received the virtual machine. Similarly the sender of VM can also not refuse the sending of VM by him.

3.1.5. Identity Protection

In many security protocols it becomes a required feature that a possible intruder should not be able to know that who is communicating with whom. In secure migration of VM the identity protection of both user and participating domain should be provided so that possible intruder becomes totally blind from participants of on-going transaction.

Logical errors in the proposed protocol flow are identified and rectified during formal verification of the protocol described in chapter four.

3.2. Inter Cloud Virtual Machine Migration

Inter cloud virtual machine migration is performed over public network between two distant data centres. As attacker has extensive control over public network as compared to migration performed within data centres so inter cloud VM migration becomes more susceptible to threats. Inside a data centre, the VMs are migrated from one host to other using local area network within closed periphery thus comparatively less susceptible to attacks as compared to VM migrated over public network. The focus of proposed solution is to address the limitations of existing techniques which are identified in literature review and devise a comprehensive protocol for securely migrating the virtual machine as an authenticated and authorized process.

The approach presented in this thesis work provides a single comprehensive solution for secure VM migration to an authenticated and authorized environment by considering its authentication, confidentiality, data integrity, identity protection and non-repudiation. The proposed protocol initially performs the local authentication of user. The authorization servers on both of the source and destination domains mutually authenticate the domains through exchange of digitally signed tickets. The domains must have acquired X.509 certificate from trusted Certificate Authority. The mutual authentication of source and destination cloud domain is performed based on Federal Information Processing Standard, FIPS PUB 196 i.e. Authentication Using public key cryptography. A symmetric session key is generated on both ends using ECDH and VM data is encrypted during transmission using AES. Moreover, least possible inter domain message exchange for mutual authentication of domains make the protocol not only secure but efficient as well.

3.3. The System's Components

There are different components involved in secure migration of virtual machine from source cloud domain to destination cloud domain which are explained as under:

- Source cloud domain
- Destination cloud domain
- Virtual machine
- Certificate Authority
- Cloud Subscriber
- VMM/ Hypervisor

Source cloud domain is the cloud domain which is hosting the migrating VM while destination cloud domain is the domain which is intended receiver of the virtual machine. Both of the cloud domains before sending or receiving the virtual machine will negotiate on set of terms which are described in protocol description. A virtual machine is the main entity that is being migrated in this protocol from source domain to the destination domain. Certificate authority is an infrastructural level domain from which both of the participating domains acquire digital certificate. As cloud is a big domain it is assumed that both of the domains have already acquired certificates from certificate authority. A cloud subscriber is the entity who has obtained IAAS cloud services through provisioning of virtual machine from cloud service provider. The process of virtual machine migration is carried out by a VMM (Virtual Machine Monitor) also called hypervisor. A hypervisor is the main component in cloud responsible for creation, deletion and management of virtual machines. It is responsible for resource allocation among multiple underlying virtual machines sharing a same piece of hardware. A cloud domain may have support of many hypervisors including both commercial as well as open source. The hypervisors supported by CloudStack (cloud operating system) include XEN, KVM, VMware and Hyper-V. It is mandatory for both participating cloud domains to run same cloud operating systems i.e.(CloudStack or Openstack) or atleast same hypervisors. The hypervisor type and version for VM migration is also supposed to be identical.

The process of VM migration is initiated based on need of a cloud provider or on cloud subscriber request. A cloud provider requires to migrate virtual machine from its data centre to another data centre to increase its data centre's resources which may fall short in peak service hours, while a cloud subscriber may require it if he finds a cost benefit with other cloud provider.

Both cloud providers (source & destination) should have acquired the certificates from CA (Certificate Authority) which they exchange to mutually authenticate each other. After both domains authenticate each other, a symmetric master key is generated using ECDH (Elliptic Curve based Diffie Hellmann scheme). This master key is further used to generate one time session key to encrypt the virtual machine before migration. The use of ECDH improves system efficiency in term of speed and provides Perfect Forward Secrecy which inhibits VM's data compromise even if key is disclosed later. The integrity of VM during transit is ensured using SHA-512. After migration integrity verification is performed and final message contains acknowledgement.

3.4. Proposed Inter-Cloud VM Migration Architecture

As shown in Figure 6, in the proposed architecture, the process of inter-cloud virtual machine migration [46] consists of following steps:

Step-1: Acquire X.509 certificates: Source and destination cloud providers are required to have X.509 certificates from a trusted Certificate Authority.

Step-2: Request for VM migration process initiation: The process of VM migration can be initiated either by a cloud provider or by a cloud subscriber. A cloud provider may require migrating virtual machine from its data centre to another data centre for increasing its data centre's resources which may fall short in peak service hours. A cloud subscriber may require VM migration if he finds cost benefit with some other cloud provider.

Step-3: Authentication from local authentication server: After verifying the credentials presented by the migration client, the authentication server provides an authentication ticket to the migration client.

Step-4: Getting authorization ticket from local authorization server: The migration client presents the authentication ticket to the authorization server. After necessary verification, authorization server issues an authorization ticket to the migration client.

Step-5: Migration request to the destination cloud domain: The migration client sends the migration request to the destination cloud domain. This request contains the public key certification of the source cloud domain and the authorization ticket issued by the authorization server of the source cloud domain.

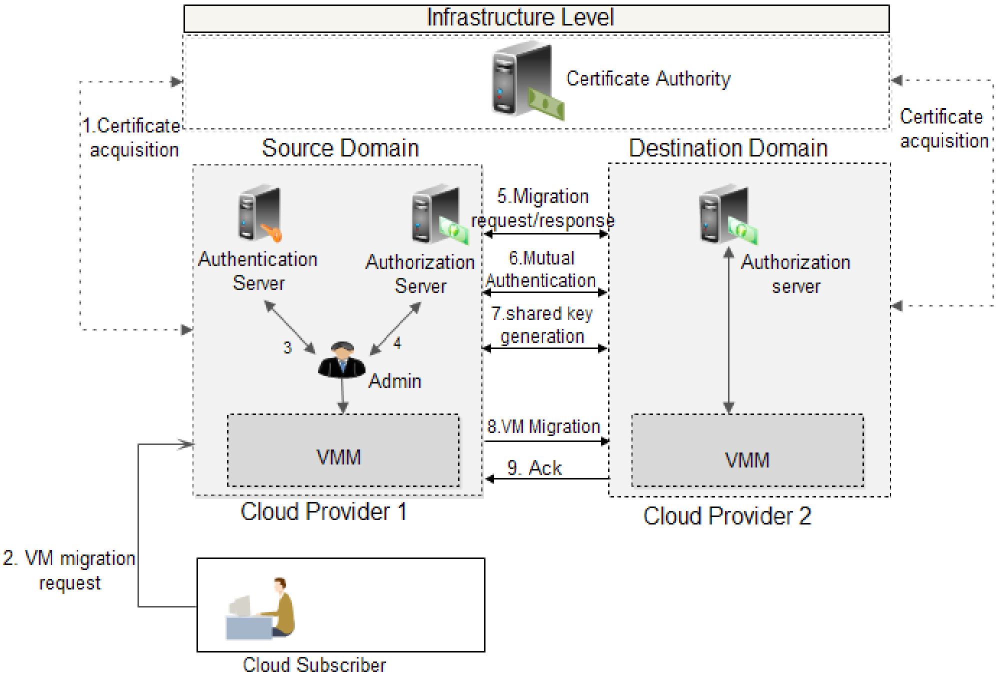


Figure 6: Proposed Architecture for Secure Migration of Virtual Machine

Step-6: Mutual Authentication: The authorization server in destination cloud domain verifies the public key certificate and authorization ticket for VM migration sent by the source domain. The authorization server in destination cloud domain verifies the rights of requesting domain for the migration request. After needful verification, the destination domain sends the positive reply for the migration request and also sends its own public key certificate. The source cloud domain verifies public key certificate of the destination cloud domain. This process provides the mutual authentication service for both source as well as destination cloud domains.

Step-7: Shared Key Generation: After both domains authenticate each other, a symmetric master key is generated using ECDH (Elliptic Curve Diffie-Hellmann Scheme). This master key is further used to generate session key to encrypt the virtual machine data before migration.

Step-8: VM Data Transfer: VM data is encrypted with the shared key using symmetric key algorithm e.g. AES and then this encrypted data is sent to the destination cloud domain. The integrity of VM data during transit is ensured using SHA-512 hash algorithm.

Step-9: Acknowledgement: Destination cloud domain performs the integrity verification and then sends back the acknowledgement message for successful transfer of virtual machine data. The process of VM data transfer and acknowledgement continues until all the VM data is successfully transferred to the destination cloud domain.

3.5. Detailed Message Exchange

Figure 7 shows the message exchange between different components of source and destination cloud domain for secure VM migration process. In the first step, the migration client is authenticated from local authentication server. The client sends authentication request message along with its user ID to the local authentication server in source domain. In response, the authentication server sends back the authentication reply message containing the user ID, *Authentication Ticket* and the shared key for secure communication between migration client and the authorization server. The communication between migration client, authentication server and authorization server is secured using shared key cryptography algorithm e.g. AES. SK_1 is shared key between migration client and the authentication server. SK_2 is the shared key between migration client and the authorization server and SK_3 is the shared key between authentication server and the authorization server. These keys can either be used as pre-shared keys or can be generated by the authentication server. Nonce is used to avoid the replay attacks.

$$Authentication\ Request = [UserID\ ||\ Nonce_1\ ]$$

$$Authentication\ Reply = [E_{SK1}(UserID||\ Nonce_1||SK_2)||(Autht\_Tkt)]$$

$$Authentication\ Ticket = [E_{SK3}(UserID\ ||\ Nonce_2\ )\ ]$$

The migration client forwards the migration request message along with authentication ticket. The authorization server decrypts the authentication ticket using shared key between authentication and authorization server i.e. SK_3. Ticket and message both contain nonce to avoid message replay attack. After verifying the authenticity of request, authorization server checks the access rights of the user. The authorization server further generates an *Authorization Ticket* containing Domain ID (DID), user ID, migration request and nonce signed with private key of source cloud domain. The message is encrypted with public key of destination cloud domain (pbB); therefore it remains confidential during transit. The destination domain decrypts this message using its private key; it also verifies the digital signature of source domain in the message. The destination's authorization server checks the rights for requesting domain and decides to proceed or abort. Furthermore, in case of positive response, the destination domain sends back the digitally signed encrypted acknowledgement to source domain.

$$Migration\ Request_{local} = [E_{SK2}(Mig\_Rqst||Dest\_DID||UserID||Nonce_3)||(Autht\_Tkt)]$$

$$Authorization\ Ticket = [E_{PrA}(Src\_DID||Dest\_DID||UserID||Nonce_4\ )\ ]$$

$$Migration\ Request_{remote} = [E_{pbB}(Mig\_Rqst||Src\_DID||Dest\_DID||UserID)||Authr\_Tkt||Cert_A]$$

$$Migration\ Response_{remote} = [E_{pbA}(Sign_{prB}(Dest\_DID||Ack||Nonce_5))||Cert_B]$$

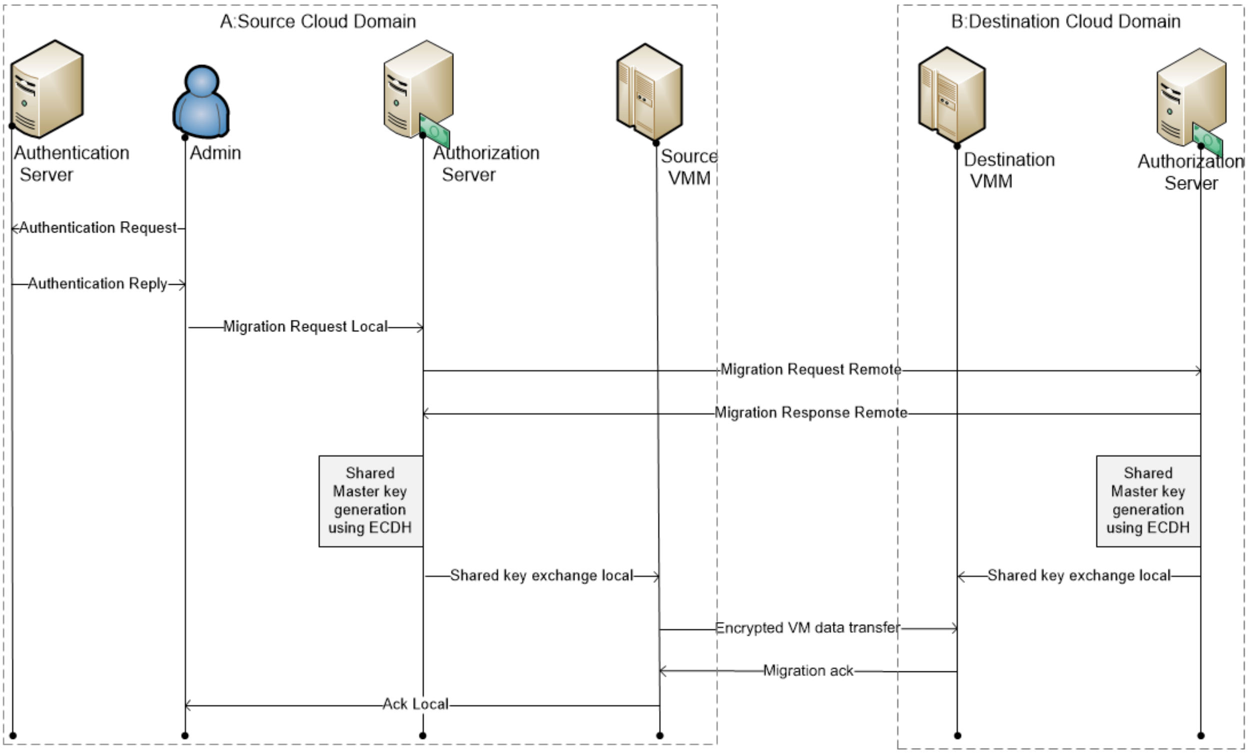


Figure 7: Message Exchange for Secure Migration of Virtual Machine

Both of the domains keep the digitally signed messages as a record thus providing the feature of non-repudiation to the system. The use of public key cryptography is not recommended for bulk data transfer e.g. VM data due to relatively slow encryption process. Therefore, a shared symmetric key is required which is used to encrypt the VM states during transit. Both source and destination domains generate shared key using Elliptic Curve Diffie-Hellman Scheme (ECDH). After generation of ECDH based shared key, the Virtual Machine Monitor (VMM) of source domain encrypts the VM states using that shared key (SK_V) and a SHA-512 hash of data is calculated and concatenated with the sent message. Destination cloud domain after successfully receiving the VM data sends back the acknowledgement messages.

$$VM\ Data\ Transfer = [E_{skv}(VM\_Data||Hash(VM\_data))]$$

$$Migration\ Ack = [E_{skv}(Ack)]$$

The use of ECDH is made due to performance and security edge that it has over simple Diffie-Hellman and other approaches for key generation. As the protocol exchanges least possible inter domain messages for mutual authentication of domains, thus we refer it as a secure and efficient protocol for VM migration.

3.6. Performance Modelling

As delay involved in migrating the virtual machine across data centres is the most important performance parameter therefore, this section models the delay involved in performing such virtual machine migration.

$$Delay = Local\ Message\ Exchange\ Delay + WAN\ Message\ Exchange\ Delay$$

$$Delay = n * \left(\frac{S_L}{B_L} + D_{PL} + D_{Proc}\right) + \ m * \left(\frac{S_W}{B_W} + D_{PW} + D_{Proc}\right)$$

Here,
n = Number of Local Control Messages Exchanged
S_L = Size of the Local Control Messages
B_L = Bandwidth on Local Link
DP_L = Propagation Delay in Local Network
D_{Proc} = Processing Delay that depends upon the cryptographic algorithms used
m = Number of Control Messages Exchanged over WAN
S_W = Size of the Control Messages Exchanged over WAN
B_W = Bandwidth on WAN Link
DP_W = Propagation Delay in WAN

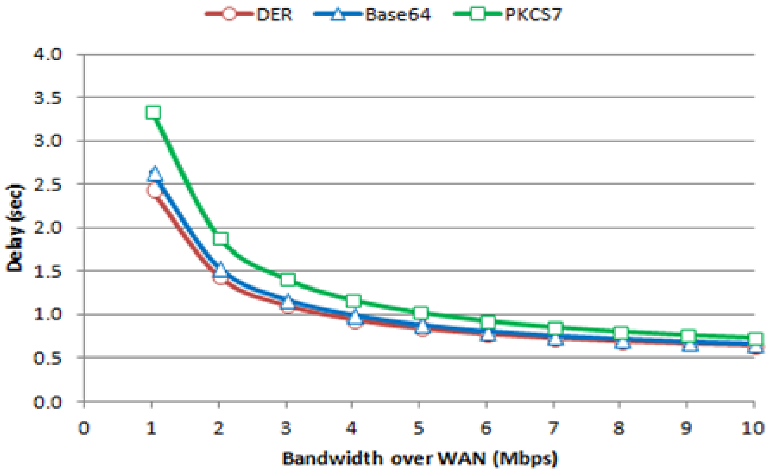


Figure 8: Delay for Migrating Virtual Machine with varying Bandwidth over WAN link

Figure 8 shows the effect of available bandwidth for WAN connectivity over migration delay. The graph is drawn for three different public key storage file formats i.e. DER, Base64 and PKCS7. The graph shows that increasing the WAN bandwidth decreases the migration delay. This trend is obvious; however, the notable thing is that when the bandwidth is increased greater than a certain limit, it gives no advantage towards decrease in migration delay.

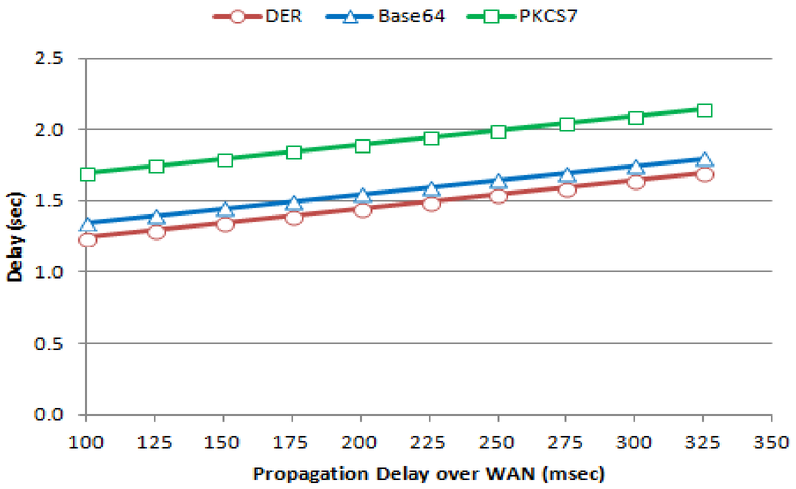


Figure 9: Delay for Migrating VM with varying Propagation Delay over WAN link

Figure 9 shows the effect of propagation delay between two datacentres locations over the migration delay. The graph shows that the propagation delay has linear affect over the migration delay i.e. with increased propagation delay the delay involved in migrating the virtual machine from one datacentre location to another datacentre location over the WAN will linearly increase. The factors that may affect the propagation delay include the available bandwidth, geographical distance between two datacentre locations, congestion over the WAN path, etc. Depending upon these mentioned parameters, propagation delay over the Internet usually varies between 100ms to 350ms and overall migration delay that is affected from this propagation delay varies only from 1 second to 2 seconds.

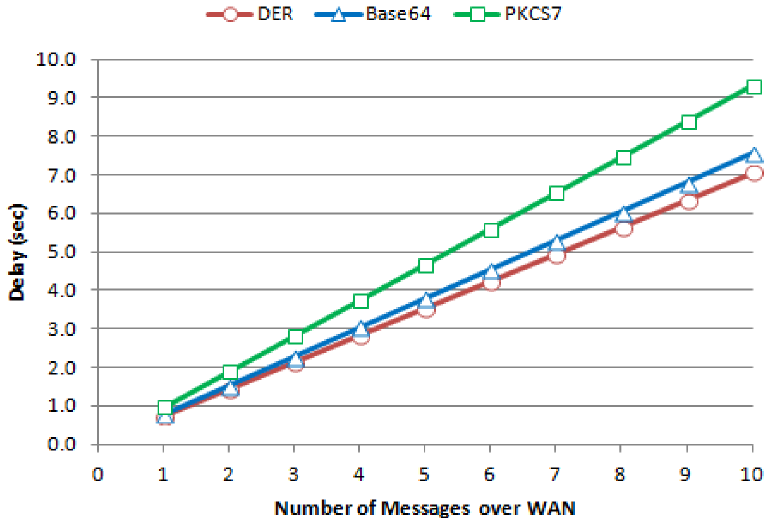


Figure 10: Delay for Migrating VM with varying Number of Control Messages over WAN link

Figure 10 shows the migration delay with the varying number of messages that are exchanged during the virtual machine migration. The number of messages depends upon two factors; one is the control messages exchanged by the migration protocol and other is the size of the virtual machine itself.

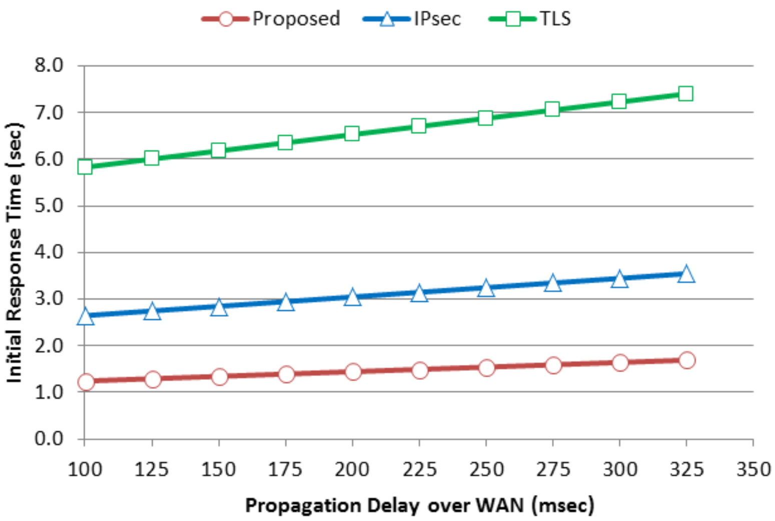


Figure 11: Comparison of Initial Response Time of the proposed architecture with IPsec and TLS

Figure 11 shows the comparison of the delay in terms of initial response time of the proposed architecture with the IPsec and TLS protocols. Initial response time is the delay involved in mutual authentication of the two cloud domains and the establishment of the shared master key. The proposed architecture exchanges two messages for this purpose whereas IPsec Internet Key Exchange Protocol (IKEv2) takes at least four control messages for this purpose [47]. Similarly Transport Layer Security Protocol (TLSv1.2) takes at least nine messages for this purpose including the Ack messages [48]. If let some of the Ack messages of TLS are piggybacked with the TLS Handshake messages even then TLS takes on average seven messages in order to complete the TLS mutual authentication and the generation of the shared key. In this respect, the overhead of the proposed architecture is less as compared to the IPsec and TLS.

Result of Figures 8, 9, 10, and 11 shows that out of number of factors e.g. available bandwidth, distance between two datacentre locations over the WAN, number of messages, the main factor that affects the migration delay is the number of messages exchanged. Although bandwidth and distance also affect the migration delay, however, their affect is considerably small as compared to the affect caused by the number of messages exchanged.

Summary

This chapter elaborates the proposed architecture for inter cloud virtual machine migration. In starting sections, the security requirements for the process of virtual machine migration are described, and then different components of the designed architecture are explained. Finally proposed architecture for inter cloud virtual machine migration is described along with protocol's performance modelling.

Chapter 4: Formal Verification

4. Formal verification

Formal verification is the act of verifying the correctness of a protocol with respect to some set properties. Formal verification is performed to verify if the security goals of the protocol are met or not. The process of formal verification applied to security protocol at initial stages aids in finding the protocol errors at early level and prohibits defective protocols from being standardized [28]. For example, an analysis of IPsec Internet key exchange protocol (IKEv1) was carried on by Perlman et al. [29] and many weaknesses in the standard were identified. Their suggestions were incorporated in IKEv2 standard. Formal analysis and verification therefore aids in finding out the logical weaknesses along with un-intended flaws in message construction. This process thus results in a protocol that is less vulnerable to zero-day attacks and more resistant towards old classic attacks. Moreover many state of the art security protocols are formally verified such as entity authentication protocols in the ISO/IEC 9798 Standard [30] and IKE key exchange protocols in the IPsec standard [31].

VM migration is a relatively new communication paradigm so for that reason several security protocols are being proposed for this process however it still lacks any standardized protocol. Formal verification of newly proposed protocol is mandatory for finding out logical and technical errors at early level. In this chapter the formal verification for proposed secure VM migration protocol is performed. This chapter begins with introduction of an attack model which elaborates the capabilities and limitations of an active compromising adversary with respect to the system. Later on, proposed security properties of the protocol are presented that need to be satisfied when the said attack model is applied to the protocol.

4.1. Attack Model

The security analysis and verification of designed protocol in the presence of an active adversary who is capable of compromising these properties is an effective notion. There are two attack models that are applied on the designed protocol. The first attack model applied on the proposed protocol is Dolev-Yao model [25]. This is a formal model to verify the properties of cryptography based security protocols. This model has two basic assumptions: i) network is under the control of attacker i.e. attacker can learn, intercept or spoof messages into the network ii) second assumption which was later called as ‘Perfect Cryptography Assumption’, states that an intruder is only limited by the constraints imposed through use of cryptographic scheme and cannot decrypt any messages unless he has the decryption keys. In this model conspiring agents or malicious insider/agent are those entities which conspire with the intruder and may provide him with some secret internal information. These abstractions are close to real time environment thus applying this attack model aids in finding out logical errors in protocol construction along with detection of various attacks that may possibly be launched.

The second attack model is KCI (Key Compromise Impersonation) attack model. It is reasonably advance and strong adversary model than DY attack model. In security protocols resilience against KCI attacks is a desired feature. It is stated as strong adversary model because it captures the resilience against key compromise impersonation attacks and provisions the scenario where an adversary can reveal session keys, random numbers and long term secret keys of participating nodes. KCI revolves around the property called Actor Key Compromise (AKC) which states that if an attacker compromises an entity A’s secret key due to whatsoever reason, A should still be able to securely communicate with other nodes depending upon the protocol used for communication [42] i.e. attacker must not be able to infer session key from compromised long term secret key. In case of successfully launched KCI attack, an adversary with secret key knowledge of A can impersonate as A to some other node B [43].

In addition to Dolev-Yao and KCI attack model, some additional compromise capabilities of a possible adversary are also included in process of formal verification which include i) Infer local state automatically ii) session key reveal. In the two above mentioned compromise capabilities the adversary tries to infer the local state of a message and also attempts to reveal session key.

4.2. Proposed Security Properties for protocol

In order to verify the correctness of proposed protocol, the process of formal verification is performed. Formal verification gives a proof that either the described security properties of the proposed scheme are met or not. First of all, a set of security properties is defined that describe the security goals of the designed architecture.

Property 1: Confidentiality This property is satisfied if the virtual machine data and other parameters used for secure session establishment such as nonce and session key remain confidential from a possible intruder. To verify that ‘property 1’ is met certain claims are set. Claim events are used to verify that intended security property is met by the protocol specifications. The confidentiality claims are as under:

claim (Src,Secret,nMRq);
claim(Src,Secret,nDt);

claim(Src,Secret,vmdata);
claim(Dest,Secret,nMRs);
claim(Dest,Secret,nDA);
The claim event with argument of “secret” if satisfied, proves that VM migration request/response (nMRq/nMRs), nonces (nDt,nDA) and VM data (vmdata) remain secret/confidential during transit from an active intruder. Above mentioned first three claims are for source domain (Src) while last two claims are for destination domain (Dest).

Property 2: Authentication This property is one of the most rudimentary objectives of security protocols. It can be divided into a hierarchy of authentication properties [26] which include aliveness, agreement and synchronisation. Aliveness property guarantees that the peer is alive and thus saves from possible Reflection attack. Agreement expresses that two participating agents agree on a set of terms, constants or variables.

claim(Src,Alive);
claim(Dest,Alive);
claim(Src, Commit, Dest, nMRq, nDt, nAT);
claim(Dest,Commit,Src,nMRs);
The first two claim events are to check aliveness of source and destination domain respectively. The remaining two claims are that source (Src) and Destination (Dest) are agreed on terms nDT, nAT, (nonce) and nMRs, nMRq (message request/ response currency). The last notion in the hierarchy of authentication is synchronization. This property is satisfied if symmetry or order of protocol execution is reserved that is all messages are sent or received in the same order as prescribed by the protocol. Synchronisation in its strong form is stated as injective synchronisation. *Injective synchronisation* property holds for an initiator or sending node if protocol synchronises and each run of source domain corresponds to unique run of destination domain [24]. Therefore synchronisation and injective synchronisation aids in avoiding attacks such as reflection, Pre-play and replay attack. Pre-play attacks are launched when an attacker predicts a message and inserts it into the system before an intended user actually creates it while replay attack involves re-sending the old captured message by attacker.
Property 3: Mutual Authentication The international standard for entity authentication ISO/IEC 9798-1:2010S [32] states that an entity is authenticated by proving its knowledge of a secret. Mutual authentication is satisfied if both sender and receiver authenticate each other as legitimate agents through proving the knowledge of secret (private key encryption). Mutual authentication property eradicates the root cause of many possible attacks such as Man in the middle and identity spoofing. This property is enforced through use of digitally signed tickets and nonce while sending a message both by source and destination domain.

send_1(Src,Dest, {srcDID, destDID, userID,nAT} sk(Src)}pk(Dest));

send_2(Dest,Src, {ackRq, nMRq, nMRs, {nMRs} sk(Dest) } pk(Src));

First ‘send_1’ event is from source to destination encrypted with private key of source domain (sk(Src)) while second ‘send_2’ event is a reply from destination to source encrypted with private key of destination domain (sk(Dest)). Encryption with private keys ensures that message is sent by the legitimate domain.

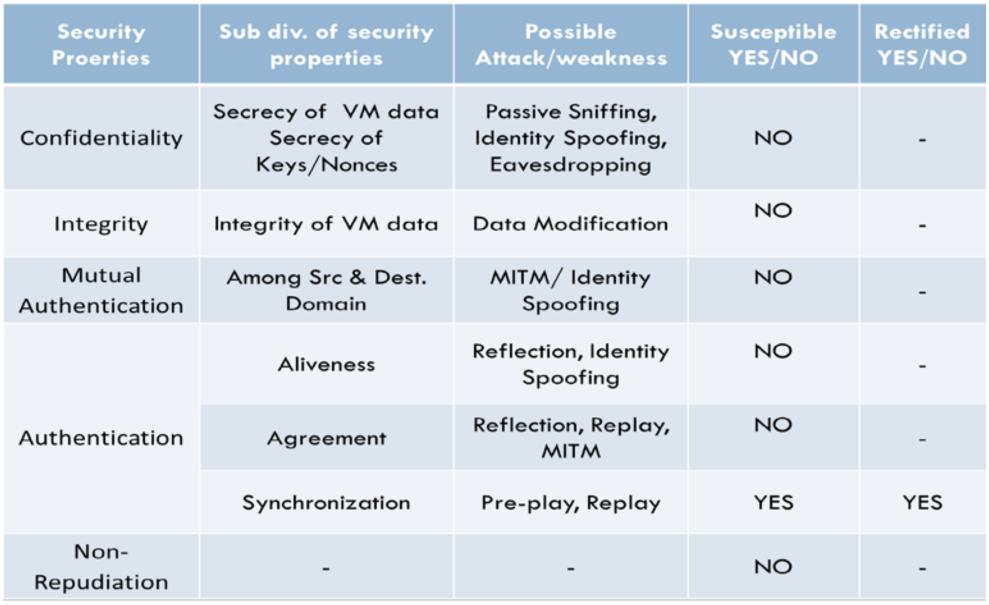
Property 4: Integrity This property is satisfied if the VM’s data integrity is not harmed during the process of migration. i.e. the data remains unchanged despite the efforts of possible intruder who tries to insert/learn the messages in transit. This property is enforced through appending a hash with every sent message preventing message stream modification.

4.3. Protocol Modelling and Analysis

The formal verification is performed for designed protocol using scyther-w32-Compromise-0.9.2 [27]. It is used for verification, falsification and a comprehensive analysis of security protocols. It is based on pattern refinement algorithm which aids in tracing all possible behaviours of the protocol, assist in analysis of classes of attacks and prove correctness for unbounded number of sessions. The protocol description is given in *spdl (Security Protocols Description Language)*. In modelling of security protocol, peers involved in protocol are defined in term of *Roles*. Roles are defined by sequence of events like send, receive and claim. The events may contain constants, freshly generated variables like nonce, session keys encrypted with/without private, public key pairs (Sk, Pk) and message encryption with symmetric key ‘k’ (E_K(M)). For the proposed secure VM migration protocol, two Roles are defined e.g. Source VM domain and Destination VM domain. In addition to Dolev-Yao attack Model, a few more adversary compromising capabilities are also added like Session key reveal and infer local state automatically.

The assessment of security properties in the presence of active adversary may reveal possible attacks and vulnerabilities. Table-2 is presented with security properties and possible weaknesses/attacks due to un-fulfilment of these security properties. The table shows that during formal analysis the protocol was either susceptible to the mentioned attacks or not and in case an attack was found possible was this attack possibility rectified or not.

Table 2: Security Properties and Sub Categories with Possible Weaknesses/Attacks

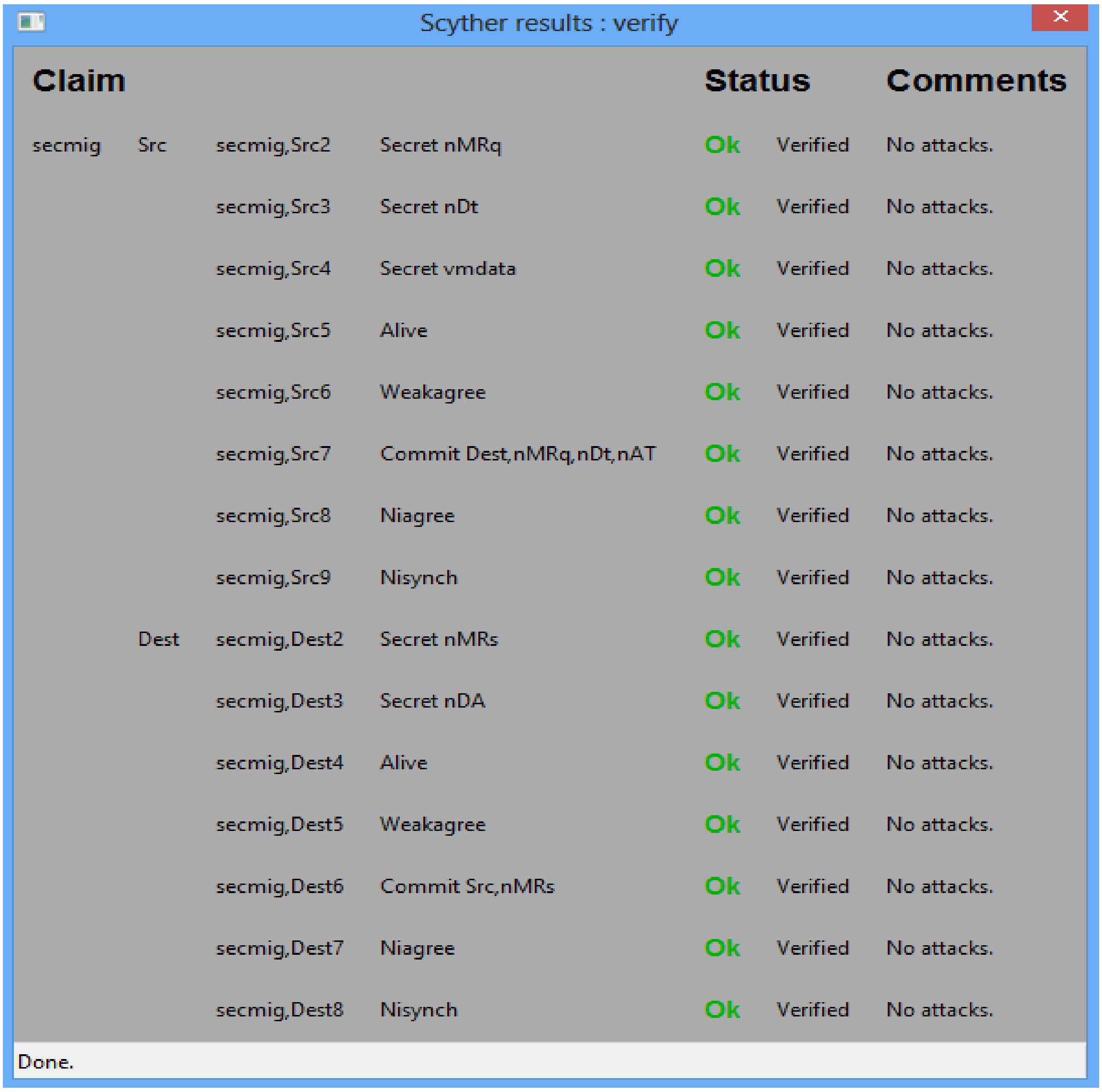


Security Proerties	Sub div. of security properties	Possible Attack/weakness	Susceptible YES/NO	Rectified YES/NO
Confidentiality	Secrecy of VM data Secrecy of Keys/Nonces	Passive Sniffing, Identity Spoofing, Eavesdropping	NO	-
Integrity	Integrity of VM data	Data Modification	NO	-
Mutual Authentication	Among Src & Dest. Domain	MITM/ Identity Spoofing	NO	-
Authentication	Aliveness	Reflection, Identity Spoofing	NO	-
	Agreement	Reflection, Replay, MITM	NO	-
	Synchronization	Pre-play, Replay	YES	YES
Non-Repudiation	-	-	NO	-

After applying attack models the authentication property was found susceptible to few attacks scenarios due to the reason that protocol was not injectively synchronising. It was possible because in attack model the attacker was able to change the symmetry of the messages exchanged, thus causing attacks like pre-play and replay to be effective. This property (synchronisation) was later enforced through introduction of a few terms which reserved the order of exchanged messages.

The first property mentioned in Table 2 is confidentiality of VM data and keys or nonce used for secure session establishment. Lack of confidentiality property may result in disclosure of migration data to adversary through passive/active sniffing and eavesdropping. This property was tested with security claim of secrecy and was proved to be met. Similarly Integrity of VM data and other messages is imposed through concatenation of hash with every sent message thus avoiding data modification. Authentication and non-repudiation is obtained using public key infrastructure i.e digital signature. Also the hierarchy of authentication properties including aliveness, agreement and synchronisation are verified thus resulting protection against reflection, identity spoofing, pre-play, replay and Man-in-the-Middle attack. Reflection attack is the kind of attack in which an attacker simply sends the message captured from sender, back to him, such that sender cannot identify it as a message actually belonging to itself.

Figure 12 given below shows the verification results for secure VM migration protocol named as ’secmig’. The security claims include Secret, Alive, Weakagree, Niagree, Nisynch and commit. Secret claim verifies the confidentiality property while alive, agree, Niagree and Nisynch make a hierarchy of authentication and verify mutual authentication and data authentication properties of the protocol. Weakagree and Niagree are weak forms of authentication as compared to Nisynch. Moreover Nisynch claim verifies that protocol is synchronising i.e. the symmetry of exchanged messages is reserved and each run of source domain (Src) corresponds to a unique run of destination domain (Dest). As mentioned before source and destination are the declared Roles in specification and each run corresponds to execution of a particular role.



Claim				Status		Comments
secmig	Src	secmig,Src2	Secret nMRq	Ok	Verified	No attacks.
		secmig,Src3	Secret nDt	Ok	Verified	No attacks.
		secmig,Src4	Secret vmdata	Ok	Verified	No attacks.
		secmig,Src5	Alive	Ok	Verified	No attacks.
		secmig,Src6	Weakagree	Ok	Verified	No attacks.
		secmig,Src7	Commit Dest,nMRq,nDt,nAT	Ok	Verified	No attacks.
		secmig,Src8	Niagree	Ok	Verified	No attacks.
		secmig,Src9	Nisynch	Ok	Verified	No attacks.
	Dest	secmig,Dest2	Secret nMRs	Ok	Verified	No attacks.
		secmig,Dest3	Secret nDA	Ok	Verified	No attacks.
		secmig,Dest4	Alive	Ok	Verified	No attacks.
		secmig,Dest5	Weakagree	Ok	Verified	No attacks.
		secmig,Dest6	Commit Src,nMRs	Ok	Verified	No attacks.
		secmig,Dest7	Niagree	Ok	Verified	No attacks.
		secmig,Dest8	Nisynch	Ok	Verified	No attacks.

Figure 12: Protocol Verification Results

Summary

This chapter provides the process of formal verification for the proposed VM migration protocol. It starts with explanation of formal verification and its significance. It then gives details of the attack model applied on the protocol clarifying the adversary capabilities. Moreover proposed security properties and their verification process are explained. The chapter concludes with protocol modelling and security analysis for the proposed VM migration protocol.

Chapter 5: Quantitative Security Metrics

5. Quantitative Security Metrics

A security metric is a computable function that tells us the extent to which a system is secure or not. This chapter provides a relative quantitative security metric model which is used to develop quantitative security metrics such as efficiency, effectiveness and CBM (Cost Benefit Measure) for evaluation of the proposed protocol.

5.1. Measurements and Metrics

Process of data collection, its analysis and reporting can be termed as measurement [36]. The measures in information security are necessary inputs to quantitative metrics for measuring efficiency and effectiveness of security goals. As quantitative metrics provide quantifiable information therefore they indicate trends that aids decision making. This is the main reason that quantitative metrics are more desirable as compared to qualitative metric which merely provide a rating in term of good, bad or average. Even though absolute numbers are more helpful and desirable, however percentages or averages are most common while designing metrics [36].

Benefits of using Metrics: Security metrics have certain benefits including:

- Increased accountability
- Compliance check
- Aids in improving security subsystem performance and cost
- Provide quantifiable information for decision making while resource allocation

An information security program of an organization matures with passage of time thus has more measures to evaluate security performance. A high level mature information security program makes use of efficiency, effectiveness and business impact measures. As shown in figure 13, security controls adopted to protect a security sub system provide it with security strength against possible threats and enables it to meet business objectives effectively. Security controls helps to provide undisturbed business operation while number of disturbances to business operations and successful attempted attacks by adversaries are the measures that can be utilised as inputs to security metrics which in turn show the efficiency, effectiveness and cost benefit analysis of security controls.

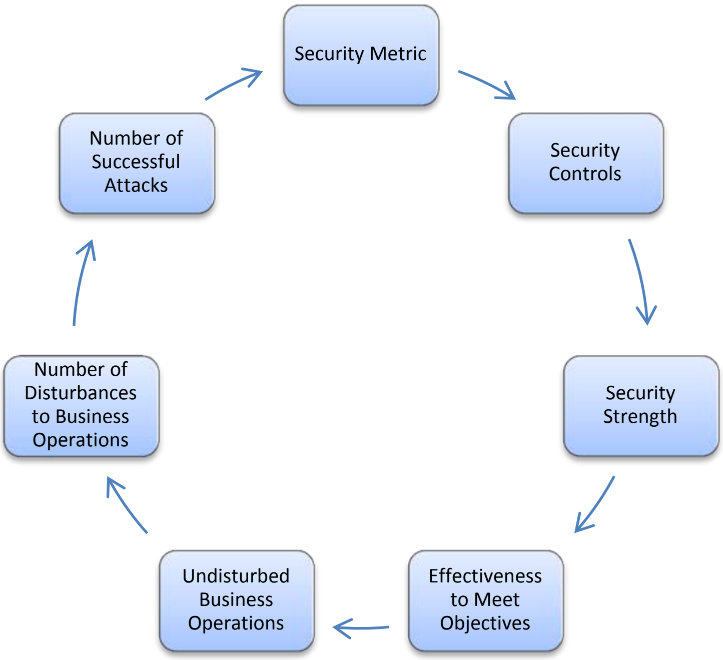

Security Metric
Security Controls
Security Strength
Effectiveness to Meet Objectives
Undisturbed Business Operations
Number of Disturbances to Business Operations
Number of Successful Attacks

Figure 13: Security Controls, Effectiveness and relationship with Business Objectives

Measurements deliver single point in time sight of certain specific factors. Measurements are mostly generated by counting the terms while metrics are generated through performing analysis of data. An example of measurement is total number of vulnerabilities detected on a web server by some vulnerability scanner [37], whilst metrics are generated by relating two or more measurements. One example of metrics would be change in the number of malwares detected by antivirus software in 2015 as compared to previous year in 2014. Security metrics should tell about the state or degree of safety relative to a reference point and what to do to avoid possible threat. Security objectives are the most important factors in information security as in the absence of security objectives it is not possible to develop useful metrics [38].

5.2. Effective Security Metrics

Effective metrics can be stated as SMART, i.e. specific, measurable, attainable, repeatable, and time-dependent. Metrics should also provide the extent to which security goals are met and should drive possible actions in order to improve the security system [37]. Security metrics can be obtained at different levels within an organization including business level to systems and service level [39]. Security metrics can be quantitative or qualitative, absolute or relative, and direct or indirect [39].

Qualitative Metric: Qualitative metrics are based on the quality of some attribute of the system.

Quantitative Metric: Quantitative metrics are the results that can be presented as numbers.

Absolute Metric: Absolute metrics use numeric values to represent the value of the measure. Information security domain is full of many unknown quantities such as unknown system weaknesses and unknown number of adversaries and their capabilities. As unknown attributes cannot be measured deterministically, therefore, it is hard to develop absolute security metrics [33].

Relative Metric: Making comparisons of different attributes of a system is a basic process for deriving relative metrics. Comparisons are helpful to establish metrics for the attributes of the systems that have no direct metric [42]. Relative metrics are normally represented in terms of percentages or comparisons.

Direct Metric: Direct metrics are based on the values that can be measured independent of the other parameters of the system. They can be measured directly.

Indirect Metric: Indirect metrics are based on the values that cannot be measured directly. They depend on the values of other parameters.

5.3. System and Security Subsystem

Let a system S with security subsystem C. If we change or enhance the security subsystem to mitigate some weakness or to provide some new security services then the changed security subsystem is presented by C'≠C. As C or C' are integral part of the system S so, with changed C' the system in new state will be termed as S'≠S. S' is considered completely new system with new set of weaknesses. When we apply some security control C it mitigates some weaknesses. However, there would still be some weaknesses that would have not been mitigated by the control C [33]. From the point of the view of the some viewer V, some of the weaknesses will be known to viewer V and some will be left unknown to V.

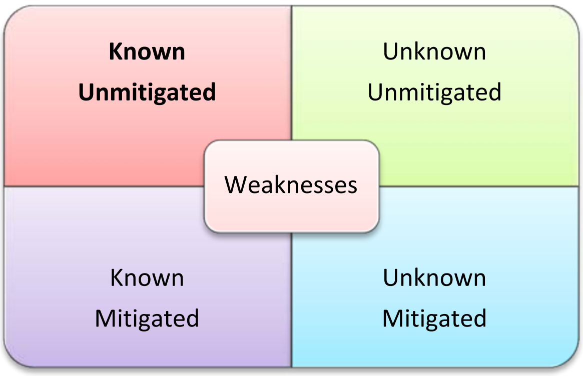


Figure 14: Categories of Weaknesses

Now, set of exploitable weaknesses (Wx) by an adversary is the intersection of set of weaknesses known to that adversary (Wk) and the set of unmitigated weaknesses (Wu) presented as [33]:

$$W_x = \{W_k\} \cap \{W_u\}$$

The fundamental goal of the security program is to reduce the set of exploitable weakness to zero.

$$W_x = \{\, \emptyset \,\}$$

In absolute terms, this security goal is not achievable, because we don’t know the number of adversaries and their capabilities [33]. Therefore, we propose a relative security metric model that can be used to quantitatively assess the relative efficiency, effectiveness and cost/benefit measure of the security subsystem.

5.4. Proposed Relative Security Metric Model

This model gives the relative, quantitative security metrics of efficiency, effectiveness and CBM (Cost Benefit Measure) of the security protocols against the attempted attacks. In order to state that how much system B is secure, we cannot give any absolute value. However, using the proposed model, relative security measure can be obtained to state that how much system B is secure as compared to the system A. However, for comparing two systems following conditions must be true:

1) Two systems in comparison are of same type
2) Attack instrument is same
3) Context is same

More often system A and B represent two different states of the same system i.e. system A may represent the previous state of the system with old security controls and system B may represent the new state of the same system with improved security controls.

Attack instrument is an instrument that is used to test the security strength of the security subsystem. Attack instrument does not enhance the security of the system rather; it is used to measure the efficiency and effectiveness of the security control. Different attack instruments can be used at different stages of the life cycle of the security controls e.g:

i) Design Stage
ii) Implementation Stage
iii) Deployment Stage

Attack instruments used at the design stage and the implementation stage represent the lab testing. Example of attack instrument that can be used at design stage is formal adversary models and the example of the attack instrument that can be used at implementation stage is penetration testing tools. Attack instruments used at the deployment stage represent the field testing with some real life adversaries.

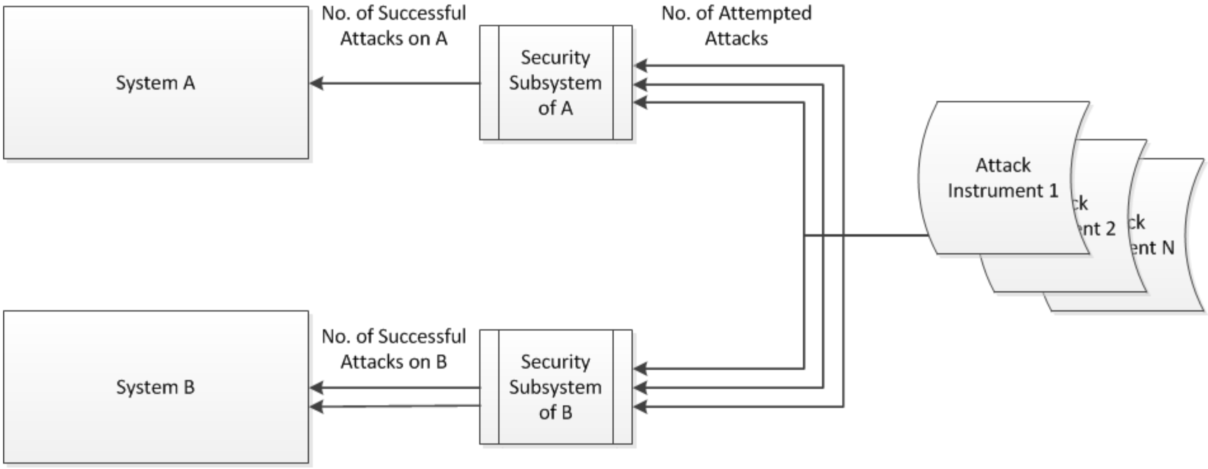


Figure 15: Model for measuring Efficiency and Effectiveness of Security Subsystem/protocols

Figure 15 presents the relative security metric model used for measuring efficiency, effectiveness and cost benefit measure of the security controls. Both systems A and B have their corresponding security sub systems which are designed to provide protection to system A and B respectively. An attack instrument is used to apply attacks on the respective security subsystems in order to test their security strength. When an attack instrument is applied to the security sub system it can block number of attacks whereas one or more attacks can be successful in bypassing the security sub system. This proposed relative security metric is used to derive three security metrics for measuring efficiency, effectiveness and cost benefit measure of security controls. These metrics are given as follows:

5.4.1 Efficiency Metric

Efficiency generally describes the extent to which a factor is well used for the intended purpose. In general, efficiency is a measurable concept, quantitatively determined by the ratio of output to input. This research work proposed the efficiency (η) of the security protocol as its resiliency to the attempted attacks. Relative security metric model is proposed to measure the efficiency of the proposed security protocol.

Let number of all attempted attacks is sum of the number of attacks blocked by the security subsystem and number of attacks successfully bypassing the security subsystem. Then efficiency of the security protocol is the ratio of number of blocked attacks by the number of all attempted attacks.

For number of attempted attacks > 0, following formula is used for calculating the efficiency of the security control of proposed protocol:

$$Efficiency\ (\eta) = \lim_{0\to 100}\left(\frac{No.of\ Blocked\ Attacks}{No.of\ Attempted\ Attacks} \times 100\right)$$

$$For\ \eta \implies \begin{Bmatrix} \eta = 100\% & ; & System\ is\ secure \\ \eta < 100\% & ; & System\ is\ potentially\ insecure \end{Bmatrix}$$

Here, impact of all successful attacks is taken as equal. However, in reality the impact of different security attacks is not always equal. In most cases, it cannot be measured in absolute sense. They are normally ranked with respect to their severity level i.e. low, medium, high. There is no consensus on how many number of low impact attacks are equal to a medium or high impact attacks. If such equivalence would have been existed then a weighted average efficiency formula would have been more appropriate and realistic.

$$No.of\ Attempted\ Attacks = No.of\ Blocked\ Attacks + No.of\ Successful\ Attacks$$

$$No.of\ Blocked\ Attacks\ \leq No.of\ Attempted\ Attacks$$

As total number all attempted attacks is relatively harder to calculate directly therefore, it is estimated by the sum of number of blocked attacks and number of successful attacks. Number of blocked attacks can be estimated from the alerts generated by the security subsystem such as firewall, IDS and antivirus programs. Similarly, number of attacks successfully bypassing the security subsystem can be estimated from sum of anomalies detected from the systems, reported by the users and sometimes claimed by the attackers. In this research work, total number of attempted attacks possible on designed protocol is determined by applying two different adversary models on it.

5.4.2. Effectiveness Metric

Effectiveness (ε) is often a relative, non-quantitative concept, mainly concerned with achieving objectives. Effectiveness is normally referred to as the capability of producing a desired result. In medicine, effectiveness relates to how well a treatment works in practice, as opposed to efficacy, which measures how well it works in clinical trials or laboratory studies. Same terminology can be used with the proposed model at different stages of the life cycle of the security system.

$$Effectiveness\ (\varepsilon) = \frac{\eta_{new} - \eta_{old}}{\eta_{old}}$$

Here, η_{new} = Efficiency of the proposed protocol with improved security control in place

η_{old} = Efficiency of the proposed protocol with previous security controls

In case when $\eta_{old} = 100\%$, there is no need of investing in η_{new}. Similarly when $\eta_{old} = 0$, then effectiveness formula will produce infinite value. Therefore, for the calculation of effectiveness (ε) following bound should exist:

$$0 < \eta_{old} < 100$$

$$0 \leq \eta_{new} \leq 100$$

With these ranges, effectiveness (ε) can be expressed as follows:

$$Effectiveness\ (\varepsilon) = \frac{\eta_{new} - \eta_{old}}{\eta_{old}}$$

$$For\ \varepsilon \Rightarrow \begin{Bmatrix} \varepsilon < 0 & ; & Improved\ Control\ has\ negative\ impact\ on\ security\ of\ protocol \\ \varepsilon = 0 & ; & Improved\ Control\ has\ no\ impact\ on\ the\ security\ of\ protocol \\ \varepsilon > 0 & ; & Improved\ Control\ has\ increased\ the\ security\ of\ protocol \end{Bmatrix}$$

5.4.3. Cost/Benefit Measure (CBM)

Cost/benefit measure indicates that how much cost is incurred for providing the security service. Here cost is taken in general e.g. cost of implementation, resources required, overhead, etc. In this work, we took number of encryptions used to implement the security protocol. Similarly, benefit is taken as efficiency with which a security protocol blocks the attempted attacks. In this sense, cost/benefit measure is the ratio of number of encryptions used by the security protocol to the efficiency of the protocol against particular attack instrument. For the sake of simplicity, we took symmetric encryptions, public key encryptions and cryptographic hashes with equal weight.

$$No.of\ Encryptions = \sum Symmetric\ Enc. + \sum Public\ Key\ Enc. + \sum Crypto\ Hashes$$

$$CBM = \frac{No.of\ Encryptions}{Efficiency\ '\eta'}$$

Efficiency ‘η’, Effectiveness ‘ε’ and CBM are Quantitative, Relative and Indirect measures in nature as shown in table 3. Quantitative in the terms that we calculate these metrics and assign numeric values to them. All three metrics are relative because they do not provide absolute sense. As we don’t have complete knowledge of weaknesses and number of adversaries and their capabilities therefore, absolute security metric can’t be measured. Efficiency depends upon the number of attacks attempted which can be computed only from the known and available attack instruments that is the incomplete set. Effectiveness is relative because it is the ratio of two efficiency values. Similarly, CBM is relative because it is the ratio of cost to the efficiency measure. All three metrics are indirect. Efficiency is derived from the ratio of the blocked attacks to the number of all the attempted attacks and Effectiveness is derived from the efficiency values.

Table 3: Summary of Nature of Metrics used in the Proposed Security Metric Model

S. No.	*Metric*	*Quantitative*	*Relative*	*Indirect*
1	Efficiency	Yes	Yes	Yes
2	Effectiveness	Yes	Yes	Yes
3	Cost/Benefit Measure	Yes	Yes	Yes

5.5. Secure Virtual Machine Migration Protocol as Target of Assessment

A security protocol is a collection of one or more security controls intended for providing protection to the system. However aggregation of multiple security controls in the name of security may sometimes result in inherently less secure system therefore measuring the extent to which security protocols are meeting their security objectives is a crucial factor. The proposed relative security metric model needs two components for performing evaluation. One is the security subsystem as Target of Assessment (ToA) and other is the attack instrument. We took secure virtual machine migration protocol as the target of assessment [46] and took a formal security verification tool scyther-w32-compromise-0.9.2 [20] as the attack instrument. We modelled three version of the secure VM migration protocol in scyther using SPDL (Security Protocol Description Language). These versions or states are termed as the i) migration protocol with no security, ii) migration protocol with initial level of security, and iii) migration protocol with improved security.

5.5.1 VM Migration Protocol Version 1: With No Security

Virtual machine migration protocol with no security control represents the protocol state when no security is provided to the virtual machine migration process. We used this state in order to identify the number of threats to which migration protocol is vulnerable. In this state migration protocol exchanges two initial control messages of migration request message and the migration response message. After a positive response message, the virtual machine migration data is sent and in response to this, the recipient sends back the acknowledgement. This process continues until all the data is transmitted at the destination end.

5.5.2. VM Migration Protocol Version 2: With Initial Security Protocol State

Secure virtual machine migration protocol presented in [46] and as described in chapter 3 is taken as the migration protocol with initial security protocol state. The proposed secure VM migration protocol provides the security services of mutual authentication of two cloud domains, confidentiality of the

VM data, integrity of VM data, non-repudiation and identity protection. The content of the messages exchanged is given in Table 4. We call it the VM migration protocol with initial security protocol state because during designed protocol analysis and formal verification, we found few unintended logical errors in protocol flow that may possibly result in number of security attacks.

Table 4: Message Contents of Secure VM Migration Protocol as Initial Security Protocol State

No.	Direction	Message Type	Message Contents
1.	$A \rightarrow B$:	$Mig\ Req\ Msg,$	$\{E_{pbB}(Mig\ Req, Src\ DID, Dest\ DID, UserID, Authr\ Tkt, Cert_A)\ \}$
			$Authr\ Tkt = \{E_{prA}(Src\ DID, Dest\ DID, UserID, Nonce_1)\}$
2.	$B \rightarrow A$:	$Mig\ Resp\ Msg,$	$\{E_{pbA}(Sign_{prB}(Dest\ DID, Ack, Nonce_2)\ ), Cert_B\}$
3.	$A \rightarrow B$:	$Data\ Msg,$	$\{E_{sk}(VM\ Data, Hash(VM\ Data))\}$
4.	$B \rightarrow A$:	$Data\ Ack\ Msg,$	$\{E_{sk}(VM\ Data\ Ack)\}$

Where,

Mig Req Msg = Migration Request Message
Mig Resp Msg = Migration Response Message
Mig Req = Migration Request
Dest DID = Destination Domain Id
Src DID = Source Domain Id
Authr Tkt = Authorization Ticket

E_{prA}= Encrypted with Private key of A
E_{pbA}= Encrypted with Public key of A
E_{pbB}= Encrypted with Public key of B
$Sign_{prB}$= Digitally Signed with Private key of B
$Cert_{A/B}$ = Certificate of Domain A or B
E_{sk}= Encrypted with shared symmetric key between A & B

5.5.3. VM Migration Protocol Version 3: With Improved Security Protocol State

The vulnerabilities of the protocol identified within initial security protocol state were mitigated by the modification in the protocol design. This modified security VM migration protocol is stated as improved security protocol. These modifications include the addition of the service of freshness of VM data, enforcement of order/symmetry of the messages exchanged and signature in the data and acknowledgement messages as shown in Table 5. The lacking of these attributes or parameters was resulting in introduction of new vulnerabilities in the protocol design which could be exploited by attacker in future in order to launch an attack.

Table 5: Message Contents of Secure VM Migration Protocol as Improved Security Protocol State

No.	Direction	Message Type	Message Contents
1.	$A \rightarrow B$:	$Mig\ Req\ Msg,$	$\{E_{pbB}(Mig\ Req, Src\ DID, Dest\ DID, UserID, Authr\ Tkt, Cert_A)\ \}$
			$Authr\ Tkt = \{E_{prA}(Src\ DID, Dest\ DID, UserID, Nonce_1)\}$
2.	$B \rightarrow A$:	$Mig\ Resp\ Msg,$	$\{E_{pbA}(Sign_{prB}(Dest\ DID, Ack, Nonce_2)\ ), Cert_B\}$

3.	$A \rightarrow B$:	$Data\ Msg,$	$\{E_{sk}(VM\ Data, Nonce_3, Nonce_2, E_{prA}(Hash(VM\ Data))\ )\}$
4.	$B \rightarrow A$:	$Data\ Ack\ Msg,$	$\{E_{sk}(VM\ Data\ Ack, Nonce_4, Nonce_3)\}$

5.6. Attack Models:

Formal security verification tool scyther-w32-compromise-0.9.2 [20] is used as the attack instrument. Three version of the secure VM migration protocol were modelled in Scyther using SPDL (Security Protocol Description Language) and then two attack models i.e. DY attack model and KCI attacks model are applied on the protocol. These attack models served as the attack instruments used to measure the values of the proposed security metrics.

5.6.1. DY Attack Model

The first attack model applied on the proposed protocol is Dolev-Yao model [25]. This is a formal model to verify the properties of cryptography based security protocols. This model has two basic assumptions: i) network is under the control of attacker i.e. attacker can learn, intercept or spoof messages into the network ii) second assumption which was later called as 'Perfect Cryptography Assumption', states that an intruder is only limited by the constraints imposed through use of cryptographic scheme and cannot decrypt any messages unless he has the decryption keys. In this model conspiring agents or malicious insider/agent are those entities which conspire with the intruder and may provide him with some secret internal information. These abstractions are close to real time environment thus applying this attack model aids in finding out logical errors in protocol construction along with detection of various attacks that may possibly be launched.

5.6.2. KCI Attack Model

The second attack model is KCI (Key Compromise Impersonation) attack model. It is reasonably advance and strong adversary model than DY attack model. In security protocols resilience against KCI attacks is a desired feature. It is stated as strong adversary model because it captures the resilience against key compromise impersonation attacks and provisions the scenario where an adversary can reveal session keys, random numbers and long term secret keys of participating nodes. KCI revolves around the property called Actor Key Compromise (AKC) which states that if an attacker compromises an entity A's secret key due to whatsoever reason, A should still be able to securely communicate with other nodes depending upon the protocol used for communication [42] i.e. attacker must not be able to infer session key from compromised long term secret key. In case of successfully launched KCI attack, an adversary with secret key knowledge of A can impersonate as A to some other node B [43].

5.7. Results

Figure 16 represents the number of successful attacks launched by two different attack instruments against different states of the security protocol. No control represents the state of the system when no security is available. Initial security state represents the state of the protocol when initial level of security measures are applied. Improved security stste represents the state of the protocol when improvements are made to the security protocol in order to thwart some specific attacks. We implemented the proposed secure virtual machine migration protocol in Scyther formal verification tool and used DY attack model and KCI attack model as sample attack instrument for our security metric model. Total twenty two security claims are taken as reference. These security claims served as objectives of the security protocol.

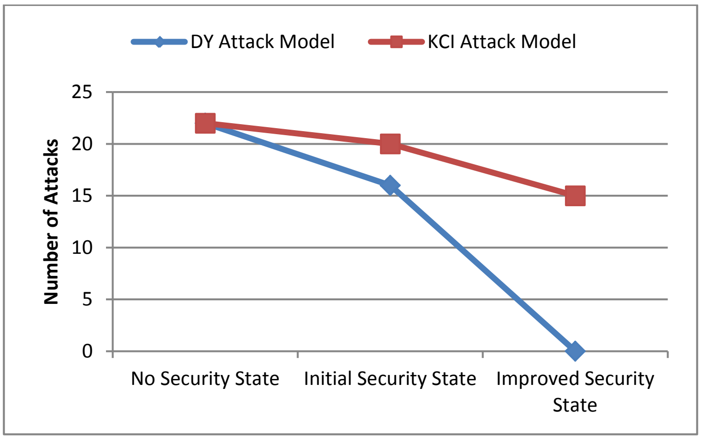


Figure 16: Number of Attacks using DY and KCI Attack Models on different Protocol States

Figure 16 represents that in protocol version 1 with no security in place, all the claims are compromised thus indicating that all the attacks launched by using DY and KCI attack models remained successful. With initial level security state in place (protocol version 2), some of the attacks are blocked while numbers of attacks were still successful. With improved security state (protocol version 3), all the attacks launched using the DY attack model are blocked, while some of the attacks launched by KCI model are still successful.

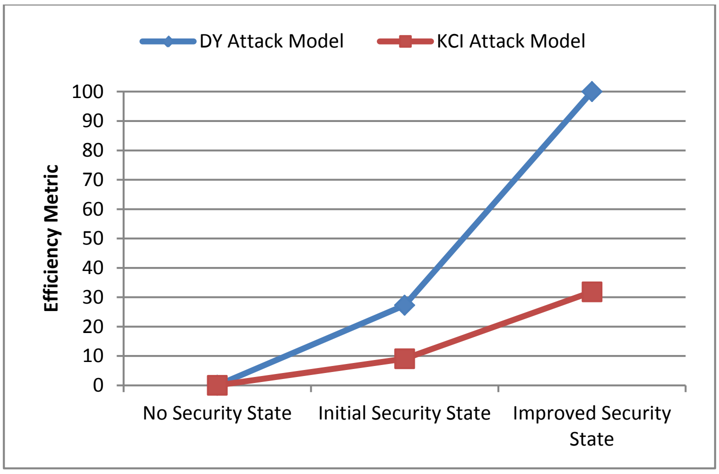


Figure 17: Efficiency of different protocol security states against DY and KCI Attack Models

Figure 17 represents the efficiency of the security protocol states against the DY and KCI attack models. As expected, with no security in place (protocol version 1), the efficiency of the security protocol is 0%. Whereas, with improved security protocol in place (protocol version 3), the efficiency of the security protocol is 100% against the DY attack model. However, efficiency is 32% against the KCI attack model.

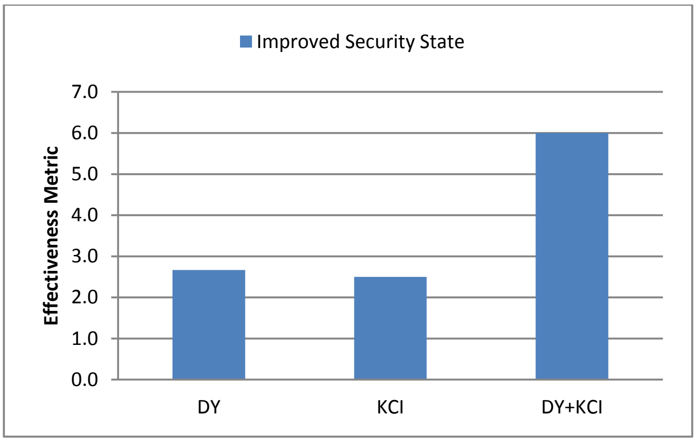


Figure 18: Relative Effectiveness Measure of Improved Security protocol state w.r.t. Initial Security protocol state against different Attack Models

Figure 18 represents the relative effectiveness of the improved security protocol with respect to the initial security protocol version against DY and KCI attack models. Positive values of effectiveness measure show that security of the protocol is increased with the improved version of the security protocol.

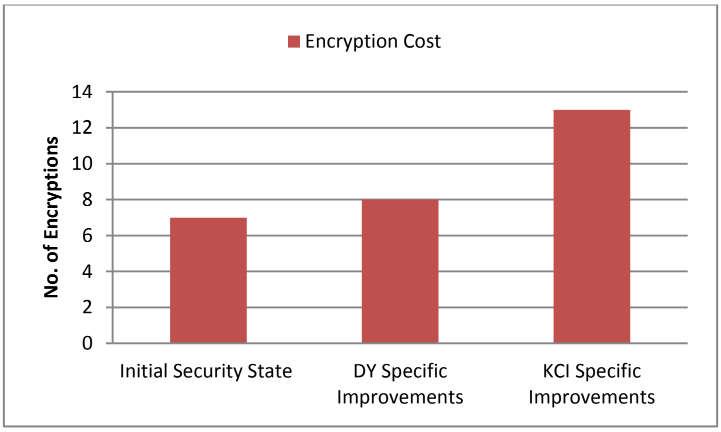


Figure 19: Cost in terms of Number of Encryptions used in different Security protocol states

Figure 19 shows the total number of encryptions used for implementing the different security protocol versions. These numbers of encryptions are used for calculating the cost/benefit measure of different security protocol versions.

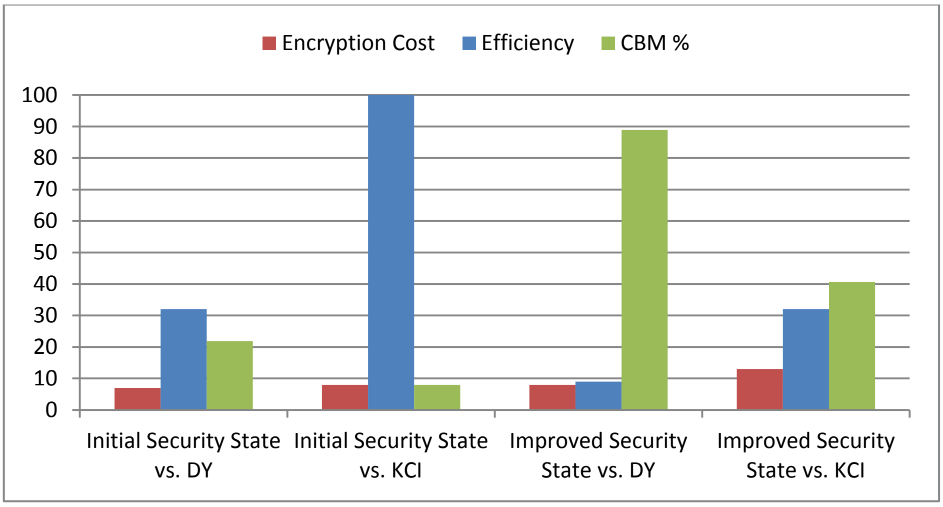


Figure 20: Cost/Benefit Measure (CBM) of different Security Protocol Versions

Figure 20 represents the cost/benefit measure of different security protocol states against DY and KCI attack models. Figure shows that CBM is high for the security protocol state provided to counter the attacks of the KCI attack model. Low value of CBM is desirable whereas high value indicates that cost of implementing the security protocol is higher as compared to the benefit gain of the security protocol against the attack model. In the figure, CBM is plotted as percentage value in order to make it comparatively visible on the chart.

Summary

This chapter initially differentiates among measures and metric. It then explains the proposed relative quantitative security metric model used for formulating security metrics. In later sections all the three *formulated metrics and their calculation methods are explained. The last section provides the results that are obtained through applying devised security metrics on proposed VM migration protocol. The results are provided for three different states of secure VM migration protocol against two different adversary models.*

Chapter 6: Conclusion and Future Directions

6. Conclusion and Future Directions

This chapter provides the conclusion of presented thesis work and highlights the possible future directions.

6.1. Conclusion

This section delivers the overall work flow carried on in this research work. A detailed literature review was carried on and security requirements for secure VM migration were identified. In the light of those security requirements a protocol for secure VM migration was proposed. Afterword, formal verification of the designed protocol was performed using scyther-w32-Compromise-0.9.2. In order to evaluate the security control's performance a relative quantitative security metric model was proposed. Three security metrics are devised in order to evaluate the efficiency, effectiveness and cost benefit measure of the security controls of proposed protocol. Efficiency is calculated in term of resilience of security controls against number of attempted attacks applied on them using two different attack/adversary models. Effectiveness is a relative metric taking ratio of difference of new and old efficiency to old efficiency. Moreover CBM metric is ratio of cost to benefit where cost is taken as total number of encryptions, hashes required to implement the security controls to the efficiency gain through the security controls. The measures are taken by applying different adversary models on the proposed protocol. The results are given for three different security control states of proposed VM migration protocol against different adversary models. Results showed that with improved security controls applied to VM migration protocol the efficiency and effectiveness was increased against both DY and KCI attack models. However, cost benefit measure value for KCI (40.6%) showed that cost for implementing security controls against KCI attack model was more than the achieved efficiency (32%) whereas against DY attack model, CBM value was lesser (8%) showing that cost for implementing the security controls was smaller as compared to gain in efficiency (100%).

6.2. Future Directions

We describe three important future directions that can be the further extension of the presented work. These future research directions are given as follows:

- In future the proposed security metric model can be used for analysis and security evaluation of more security protocols, products and standards.
- We aim to evaluate the existing security solutions for secure VM migration using the security metric model and devised security metrics proposed in this research work.
- As presented in NIST publication 800-55, Security metrics area is less explored so in this regard, we intend to define more security metrics for security analysis and evaluation of variety of security sub systems.

Summary

This chapter briefly concludes all the research work described in previous five chapters of this thesis document including literature review, proposed secure VM migration protocol, its formal verification and security metrics devised for the security evaluation of VM migration protocols. This chapter ends with a description of future directions that can be used to extend this work.

References

[1] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, A. Wareld, “Live Migration of Virtual Machines," In Proceedings of 2nd Conference on Networked Systems Design & Implementation, Vol 2, pp. 273-286, USENIX Association, (2005).

[2] D. Botero, "A Brief Tutorial on Live Virtual Machine Migration from a Security Perspective", Technical Report, University of Princeton, USA (2011).

[3] T. Wood, P. Shenoy, K. Ramakrishnan, J. Merwe, “CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines,” In Proceedings of the 7th ACM International Conference on Virtual Execution Environments, SIGPLAN/SIGOPS, Vol. 46(7), pp. 121-132, NY, USA, ACM, (2011).

[4] P. Pisa, N. Fernandes, H. Carvalho, M. Moreira, M. Campista, L. Costa, and O. Duarte, “Openflow and Xen-based Virtual Network Migration," In Proceedings of International Federation for Information Processing, Vol. 327, pp. 170-181, Springer, (2010).

[5] F. Travostino, et al. “Seamless Live Migration of Virtual Machines over the MAN/WAN,” Future Generation Computer Systems, Vol. 22(8), pp. 901-907, Elsevier, (2006).

[6] P. Riteau, C. Morin, T. Priol, "Shrinker: Efficient Wide-Area Live Virtual Machine Migration using Distributed Content-Based Addressing," In Proceedings of *Euro-Par 2011 Parallel Processing*, pp. 21-27, Berlin Heidelberg, Springer, (2011).

[7] S. Al-Kiswany, D. Subhraveti, P. Sarkar, M. Ripeanu, “VMFlock: Virtual Machine Co-Migration for the Cloud,” In Proceedings of 20th International Symposium on High Performance Distributed Computing, pp. 159-170, ACM, (2011).

[8] W. Voorsluys, J. Broberg, S. Venugopal, R. Buyya. "Cost of Virtual Machine Live Migration in Clouds: A performance evaluation," Cloud Computing, pp. 254-265. Springer, (2009).

[9] S.T. King et al., “SubVirt: Implementing Malware with Virtual Machines,” In Proceedings of IEEE Symposium on Security and Privacy (SP 06), pp. 314-327, IEEE, (2006).

[10] M. Price, "The paradox of security in virtual environments," Computer, Vol. 41(11), pp. 22-28, IEEE, (2008).

[11] F. Lombardi, R. DiPietro, “Secure Virtualization for Cloud Computing,” Journal of Network and Computer Applications, Vol. 34(4), pp. 1113-1122, Elsevier, (2011).

[12] R. Schwarzkopf et al., “Increasing Virtual Machine Security in Cloud Environments,” Journal of Cloud Computing: Advances, Systems and Applications, Vol. 1(1), pp. 1-12, Springer, (2012).

[13] J. Oberheide, E. Cooke, F. Jahanian, “Empirical Exploitation of Live Virtual Machine Migration,” In Proceedings of BlackHat DC Convention,USA, (2008).

[14] Zhang, F., Huang, Y., Wang, H. “PALM: Security Preserving VM Live Migration for Systems with VMM-enforced Protection,” In Proceedings of 3rd Asia-Pacific Trusted Infrastructure Technologies Conference, pp. 9-18, IEEE, (2008).

[15] M. Aslam, C. Gehrmann, M. Bjorkman, “Security and Trust Preserving VM Migrations in Public Clouds,” In Proceedings of IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, (TrustCom), pp. 869-876, IEEE, (2012).

[16] B. Danev, et al. "Enabling Secure VM-vTPM Migration in Private Clouds," In Proceedings of 27th Annual Computer Security Applications Conference (ACSAC)," pp. 187-196, ACM, (2011).

[17] The Xen Project, www.xenproject.org, (Accessed on 11-12-2013).

[18] VMware Virtualization for Desktop & Server, Application, Public & Hybrid Clouds, www.vmware.com, (Accessed on 11-12-2013).

[19] Kernel based Virtual Machine, www.linux-kvm.org, (Accessed on 11-12-2013).

[20] Y. Devi, P. Aruna, D. Sudha, "Security in Virtual Machine Live Migration for KVM," In Proceedings of International Conference on Process Automation, Control and Computing (PACC), pp. 1-6, IEEE, (2011).

[21] J. Shetty, M. R. Anala, G. Shobha, "A Survey on Techniques of Secure Live Migration of Virtual Machine," International Journal of Computer Applications, Vol. 39(12), pp. 34-39, (2012).

[22] C. Xianqin, G. Xiaopeng, W. Han, W. Sumei, L. Xiang. "Application- Transparent Live Migration for Virtual Macshine on Network Security Enhanced Hypervisor," In Proceeding of China Communication, Vol. 8(3), pp. 32-42, (2011).

[23] NIST Guide to Security for full Virtualization, Special Publication 800-125, (2011).

[24] Cremers, Cas JF, Sjouke Mauw, and Erik P. de Vink. "Injective synchronisation: an Extension of the Authentication Hierarchy," Theoretical Computer Science, Vol. 367(1), pp. 139-161, Elsevier, (2006).

[25] D. Yao, A. C. Yao "On the Security of Public Key Protocols," In Proceedings of IEEE Transaction on Information Theory, Vol. 29(2), pp. 198–208, IEEE, (1983).

[26] G. Lowe, "A hierarchy of authentication specifications," In Proceedings of Computer Security Foundation Workshop, pp. 31–44, New York, USA, IEEE, (1983).

[27] C. Cremers, "The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols," Computer Aided Verification, pp. 414-418, Springer, (2008).

[28] D. Basin, C. Cremers, C. Meadows. "Model Checking Security Protocols," Handbook of Model Checking, (2011).

[29] R.J Perlman, C Kaufman, "Analysis of the IPSec Key Exchange standard," In Proceedings of 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 150-156, Cambridge, MA, USA, IEEE, (2001).

[30] D. Basin, C. Cremers, and S. Meier, "Provably Repairing the ISO/IEC 9798 Standard for Entity Authentication," In Proceedings of First International Conference, Held as Part of the European Joint Conferences on Theory and Practice of Software, Lecture Notes in Computer Science, Vol. 7215, pp. 129–148, Springer, (2012).

[31] C. Cremers, "Key exchange in IPsec revisited: formal analysis of IKEv1 and IKEv2," In Proceedings of the 16th European Conference on Research in Computer Security, pp. 315–334, Berlin, Heidelberg, Springer-Verlag, (2011).

[32] International Organization for Standardization, Geneve, Switzerland. ISO/IEC 9798-1:2010, Information technology-Security techniques-Entity Authentication-Part 1: General, Third edition, (2010).

[33] M. Torgerson, "Security Metrics for Communication Systems," In Proceedings of 12th International Command and Control Research and Technology Symposium (ICCRTS), Newport, Rhode Island, (2007)

[34] B.A. LaMacchia, K. Lauter, A. Mityagin. "Stronger Security of Authenticated Key

Exchange," In Proceedings of ProvSec 07, Lecture Notes on Computer Science, Vol. 4784, pp. 1–16, Springer, (2007).

[35] W. Jansen, "Directions in Security Metrics Research," NIST Interagency Report 7564, April (2009).

[36] Elizabeth Chew, et al., "Performance Measurement Guide for Information Security," NIST Special Publication 800-55 Rev. 1, (2008).

[37] A Guide to Effective Security Metrics, Version 1.0, (2012).

[38] W. K. Brotby, "Security Metrics Overview", Accessed on 24-12-2014, http://www.infosectoday.com/Articles/Security_Metrics_Overview.htm.

[39] R.M. Savola, "A Security Metrics Taxonomization Model for Software-Intensive Systems," Journal for Information Processing Systems, Vol. (5)4, pp. 197-206, (2009).

[40] Department of Homeland Security, "A Roadmap for Cyber Security Research," http://www.cyber.st.dhs.gov/docs/DHSCybersecurityRoadmap.pdf, (2009).

[41] T. L. Saaty, "Relative Measurement and Its Generalization in Decision Making," In Proceedings of Rev. Real Academia de Ciencias: Serie A. Mathematicas (RACSAM), Vol. 102(2), pp. 251–318, Springer, (2008).

[42] D. Basin, C. Cremers, M. Horvat. "Actor Key Compromise: Consequences and Countermeasures," In Proceedings of 27th. IEEE Computer Security Foundations Symposium (CSF), pp. 244-258, IEEE, (2014).

[43] M. C. Gorantla, et al. "Modeling Key Compromise Impersonation Attacks on Group Key Exchange Protocols," In Proceedings of Transactions on Information and System Security (TISSEC) Vol. (14)4, ACM, (2011).

[44] M. S. Ahmed, E. Al-Shaer, L. Khan, "A Novel Quantitative Approach for Measuring Network Security," In Proceedings of 27th Conference on Computer Communication, INFOCOM, IEEE, (2008).

[45] R. B. Vaughn, R. Henning, A. Siraj, "Information Assurance Measures and Metrics-State of Practice and Proposed Taxonomy," In Proceedings of the 36th Annual Hawaii International Conference, pp. 10-19, IEEE, (2003).

[46] T. Zeb, A. Ghafoor, A. Shibli, M. Yousaf, "A Secure Architecture for Inter-Cloud Virtual Machine Migration," In 10th International Conference on Security and Privacy in Communication Networks (SecureComm), Springer, (2014).

[47] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)," IETF RFC-5996, (2010).

[48] T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," IETF RFC-5246, (2008).

[49] J. A. Wang, H. Wang, M. Guo, M. Xia, "Security Metrics for Software Systems," In Proceedings of 47th Annual Southeast Regional Conference, Vol.(47)1, pp. 6-20, ACM, (2009).

[50] I. Chowdhury, B. Chan, and M. Zulkernine, "Security Metrics for Source Code Structures," In Proceedings of the Fourth International Workshop on Software Engineering for Secure Systems, pp. 57-64, ACM, (2008).

[51] B. Alshammari, C. Fidge, D. Corney, "Security Metrics for Object-Oriented Class Designs," In Proceedings of 9th International Conference on Quality Software, pp. 11-20, IEEE, (2009).

Appendix

Appendix

Scyther Code

Secure VM Migration Protocol ver1: No Security Controls

```
/*

 * The description of secure migration of vm protocol.  No Security Control in place.

 * All Type Attacks:  Automatic F6, Attack Instrument DY=KCI=After PFS = 22, After Correct
wPFS=36

 */

usertype ID, Data, Type;

protocol secmig(Src,Dest)

{

role Src

        {
//data sent

fresh srcDID, destDID, userID: ID;                    // Source Domain ID

fresh migReq: Type;                                   // Message Type - Migration Request

fresh vmdata: Data;                                   // vm data
//data received
var ackRq, ackDt: Type;
send_1(Src,Dest,  migReq, srcDID, destDID, userID)  ; // migration request remote

recv_2(Dest,Src, ackRq, destDID );                    //claim(Src, Running, Dest, nMRs);
send_3(Src, Dest, vmdata);
recv_4(Dest, Src, ackDt);
claim(Src, Secret, vmdata);                           // secrecy of vmdata
claim(Src, Alive);

claim(Src, Weakagree);                                //claim(Src, Commit, Dest,  nAT);

claim(Src, Niagree);                                  // non-injective agreement

claim(Src, Nisynch);                                  // non-injective synchronization

} ////////////////////////////////////////// End Role Src

/////////// Role - Desitnation
role Dest

{
```

```
//data sent

fresh ackRq, ackDt: Type;
// data received
var migReq: Type;                                      // Message Type - Migration Request
var srcDID, destDID, userID: ID;                       // Source Domain ID
var vmdata: Data;
recv_1(Src,Dest,  migReq, srcDID, destDID, userID );   // migration request remote - received
send_2(Dest,Src, ackRq, destDID );                     // migration response remote – sent

recv_3(Src,Dest, vmdata);
//claim(Dest,Running,Src,  nAT);
send_4(Dest, Src, ackDt);
//claim(Dest,Commit,Src,nMRs);
//claim(Dest,Secret,nMRs);

claim(Dest,Alive);

claim(Dest,Weakagree);

claim(Dest,Niagree);

claim(Dest,Nisynch);

} /////////////////////////////////////////// End Role Dest

} /////////////////////////////////////////// End protocol secmig
```

Secure VM Migration Protocol ver2: Initial Security Controls

```
/*
 * The description of secure migration of vm protocol.  SecureComm 2014, Initial Controls
 * Automatic F6: Attack Instrument DY=15, KCI=21, After PFS = 24, After Correct wPFS=22,
Session Key Reveal=24
 */
usertype ID, Data, Type;
hashfunction H;
protocol secmig(Src,Dest)

{

role Src

{

//data sent

fresh nAT:Nonce;                         // Currency of Authenticaion Ticket
//              fresh nAT, nDT:Nonce;    // Currency of Authenticaion Ticket
fresh srcDID, destDID, userID: ID;       // Source Domain ID
fresh migReq: Type;                      // Message Type - Migration Request
fresh vmdata: Data;                      // vm data
//data received

var nMRs:Nonce;                          // Currency of Migration Response Message
//              var nMRs, nAD:Nonce;     // Currency of Migration Response Message
```

```
var ackRq, ackDt: Type;
send_1(Src,Dest, {migReq, srcDID, destDID, userID} pk(Dest),  {(srcDID, destDID, userID, nAT)}
sk(Src)  );                                                              // migration request
remote

//send_1(Src,Dest, { {H(srcDID, destDID, userID, nAT)} sk(Src) } pk(Dest)   );  // migration request
remote
recv_2(Dest,Src, { {ackRq, destDID, nMRs} sk(Dest) } pk(Src) );

claim(Src, Running, Dest, nMRs);
send_3(Src, Dest, { vmdata, H(vmdata) }k(Src,Dest) );
//send_3(Src, Dest, {vmdata, nDT, nMRs, {H(vmdata, nDT)}sk(Src)} k(Src,Dest));
//send_3(Src, Dest, {{H(vmdata, nDT, nMRs)}sk(Src)} k(Src,Dest));
recv_4(Dest, Src, { ackDt }k(Src,Dest) );
claim(Src, Secret, vmdata);                     // secrecy of vmdata
claim(Src, Alive);

claim(Src, Weakagree);
claim(Src, Commit, Dest,  nAT);

claim(Src, Niagree);                            // non-injective agreement

claim(Src, Nisynch);                            // non-injective synchronization

} //////////////////////////////////////////// End Role Src

/////////// Role - Desitnation
role Dest

{
//data sent

fresh nMRs:Nonce;                               // Currency of Migration Response Message


//fresh nMRs, nAD:Nonce;                        // Currency of Migration Response Message
fresh ackRq, ackDt: Type;
// data received
var nAT:Nonce;                                  // Currency of Authenticaion Ticket
//var nAT, nDT:Nonce;                           // Currency of Authenticaion Ticket
var migReq: Type;                                       // Message Type - Migration Request
var srcDID, destDID, userID: ID;                                // Source Domain ID
var vmdata: Data;
recv_1(Src,Dest,  {migReq, srcDID, destDID, userID} pk(Dest),  {(srcDID, destDID, userID, nAT)}
sk(Src)  );                                             // migration request remote - received


//recv_1(Src,Dest,  { {H(srcDID, destDID, userID, nAT)} sk(Src) } pk(Dest)   );  // migration request
remote - received
send_2(Dest,Src, { {ackRq, destDID, nMRs} sk(Dest) } pk(Src) );
                                                        // migration response remote - sent

recv_3(Src,Dest, { vmdata, H(vmdata) }k(Src,Dest) );
claim(Dest,Running,Src,  nAT);
send_4(Dest, Src, { ackDt }k(Src,Dest) );

//send_4(Dest, Src, {ackDt, nAD, nDT, {H(ackDt, nAD)}sk(Dest)} k(Src,Dest));
```

```
//send_4(Dest, Src, {{H(ackDt, nAD, nDT)}sk(Dest)} k(Src,Dest));
claim(Dest,Commit,Src,nMRs);
claim(Dest,Secret,nMRs);

claim(Dest,Alive);

claim(Dest,Weakagree);

claim(Dest,Niagree);

claim(Dest,Nisynch);

} //////////////////////////////////////////// End Role Dest
} //////////////////////////////////////////// End protocol secmig
```

Secure VM Migration Protocol ver3: Improved Security Controls

```
/*
 * The description of secure migration of vm protocol.  Improved Security Controls
 * Automatic F6: Attack Instrument DY=15, KCI=21, After PFS = 24, After Correct wPFS=22,
Session Key Reveal=24

 */

usertype ID, Data, Type;
hashfunction H;
protocol secmig(Src,Dest)

{

role Src

{
//data sent

//fresh nAT:Nonce;                  // Currency of Authenticaion Ticket
fresh nAT, nDT:Nonce;               // Currency of Authenticaion Ticket
fresh srcDID, destDID, userID: ID;                  // Source Domain ID
fresh migReq: Type;                 // Message Type - Migration Request
fresh vmdata: Data;                 // vm data
//data received


//var nMRs:Nonce;                   // Currency of Migration Response Message
var nMRs, nAD:Nonce;                // Currency of Migration Response Message
var ackRq, ackDt: Type;
send_1(Src,Dest, {migReq, srcDID, destDID, userID,  {(srcDID, destDID, userID, nAT)} sk(Src) }
pk(Dest)  );                                    // migration request remote

//send_1(Src,Dest, { {H(srcDID, destDID, userID, nAT)} sk(Src) } pk(Dest)   );  // migration request
remote
recv_2(Dest,Src, { {ackRq, destDID, nMRs} sk(Dest) } pk(Src) );

claim(Src, Running, Dest, nMRs);
send_3(Src, Dest, {{ vmdata, nDT, nMRs, H(vmdata) }k(Src,Dest)}pk(Dest) );
```

```
//send_3(Src, Dest, {vmdata, nDT, nMRs, {H(vmdata, nDT)}sk(Src)} k(Src,Dest));


//send_3(Src, Dest, {{H(vmdata, nDT, nMRs)}sk(Src)} k(Src,Dest));
recv_4(Dest, Src, { ackDt, nAD, nDT }k(Src,Dest) );
claim(Src, Secret, vmdata);                     // secrecy of vmdata
claim(Src, Alive);

claim(Src, Weakagree);
claim(Src, Commit, Dest,  nAT);

claim(Src, Niagree);                            // non-injective agreement

claim(Src, Nisynch);                            // non-injective synchronization

} ///////////////////////////////////////////// End Role Src

/////////// Role - Desitnation
role Dest

{
//data sent

//fresh nMRs:Nonce;                             // Currency of Migration Response Message
fresh nMRs, nAD:Nonce;                          // Currency of Migration Response Message
fresh ackRq, ackDt: Type;
// data received

//var nAT:Nonce;                                // Currency of Authenticaion Ticket
var nAT, nDT:Nonce;                             // Currency of Authenticaion Ticket
var migReq: Type;                               // Message Type - Migration Request
var srcDID, destDID, userID: ID;                        // Source Domain ID
var vmdata: Data;
recv_1(Src,Dest,  {migReq, srcDID, destDID, userID,  {(srcDID, destDID, userID, nAT)} sk(Src) }
pk(Dest) );                                             // migration request remote - received


//recv_1(Src,Dest,  { {H(srcDID, destDID, userID, nAT)} sk(Src) } pk(Dest)   );  // migration request
remote - received
send_2(Dest,Src, { {ackRq, destDID, nMRs} sk(Dest) } pk(Src) );
                                                        // migration response remote - sent

recv_3(Src,Dest, {{ vmdata, nDT, nMRs, H(vmdata) }k(Src,Dest)}pk(Dest) );
claim(Dest,Running,Src,  nAT);
send_4(Dest, Src, { ackDt, nAD, nDT}k(Src,Dest) );
//send_4(Dest, Src, {ackDt, nAD, nDT, {H(ackDt, nAD)}sk(Dest)} k(Src,Dest));
//send_4(Dest, Src, {{H(ackDt, nAD, nDT)}sk(Dest)} k(Src,Dest));
claim(Dest,Commit,Src,nMRs);
claim(Dest,Secret,nMRs);

claim(Dest,Alive);

claim(Dest,Weakagree);

claim(Dest,Niagree);

claim(Dest,Nisynch);
```

```
} //////////////////////////////////////////// End Role Dest

} //////////////////////////////////////////// End protocol secmig

/*
 * The description of secure migration of vm protocol.  Improved Security Controls - KCI Support

 * Automatic F6: Attack Instrument DY=15, KCI=21, After PFS = 24, After Correct wPFS=22,
Session Key Reveal=24

 */
usertype ID, Data, Type;
hashfunction H;
protocol secmig(Src,Dest)

{

role Src

{
//data sent

fresh nAT, nDT:Nonce;             // Currency of Authenticaion Ticket
fresh srcDID, destDID, userID: ID;                // Source Domain ID
fresh migReq: Type;               // Message Type - Migration Request
fresh vmdata: Data;               // vm data
//data received
var nMRs, nAD:Nonce;              // Currency of Migration Response Message
var ackRq, ackDt: Type;
send_1(Src,Dest, {migReq, srcDID, destDID, userID,  nAT, {H(srcDID, destDID, userID, nAT)}
sk(Src) } pk(Dest)  );                          // migration request remote

recv_2(Dest,Src, { ackRq, nMRs, { H(ackRq, nMRs, nAT)} sk(Dest) } pk(Src) );

claim(Src, Running, Dest, nMRs);

//send_3(Src, Dest, { vmdata, nDT, { H(vmdata, nMRs) }sk(Src) }k(Src,Dest) );

// kci attacks 17
send_3(Src, Dest, { { vmdata, nDT, { H(vmdata, nMRs) }sk(Src) }k(Src,Dest) }pk(Dest) );

// kci attacks 15

//recv_4(Dest, Src, { { { ackDt, nAD, { H(ackDt, nDT) }sk(Dest) }k(Src,Dest) }pk(Src) }k(Src,Dest)
);
recv_4(Dest, Src, { { ackDt, nAD, { H(ackDt, nDT) }sk(Dest) }k(Src,Dest) }pk(Src) );
claim(Src, Secret, vmdata);                       // secrecy of vmdata
claim(Src, Alive);

claim(Src, Weakagree);
claim(Src, Commit, Dest,  nAT);

claim(Src, Niagree);                              // non-injective agreement

claim(Src, Nisynch);                              // non-injective synchronization

} //////////////////////////////////////////// End Role Src
```

```
/////////// Role - Desitnation
role Dest

{
//data sent

fresh nMRs, nAD:Nonce;                    // Currency of Migration Response Message
fresh ackRq, ackDt: Type;
// data received
var nAT, nDT:Nonce;                       // Currency of Authenticaion Ticket
var migReq: Type;                         // Message Type - Migration Request
var srcDID, destDID, userID: ID;          // Source Domain ID
var vmdata: Data;
recv_1(Src,Dest,  {migReq, srcDID, destDID, userID,  nAT, {H(srcDID, destDID, userID, nAT)}
sk(Src) } pk(Dest) );      // migration request remote - received
send_2(Dest,Src, { ackRq, nMRs, { H(ackRq, nMRs, nAT) }sk(Dest) }pk(Src) );
                                                  // migration response remote - sent

//recv_3(Src,Dest, { vmdata, nDT, { H(vmdata, nMRs) }sk(Src) }k(Src,Dest));
recv_3(Src,Dest, {{ vmdata, nDT, { H(vmdata, nMRs) }sk(Src) }k(Src,Dest)}pk(Dest) );
claim(Dest,Running,Src,  nAT);
send_4(Dest, Src, { { ackDt, nAD, { H(ackDt, nDT) }sk(Dest) }k(Src,Dest) }pk(Src) );

//send_4(Dest, Src, { { { ackDt, nAD, { H(ackDt, nDT) }sk(Dest) }k(Src,Dest) }pk(Src) }k(Src,Dest)
);

//send_4(Dest, Src, { { ackDt, nAD, { H(ackDt, nDT) }sk(Dest) }pk(Src) }k(Src,Dest) );
claim(Dest,Commit,Src,nMRs);
claim(Dest,Secret,nMRs);

claim(Dest,Alive);

claim(Dest,Weakagree);

claim(Dest,Niagree);

claim(Dest,Nisynch);

} //////////////////////////////////////////////// End Role Dest
} //////////////////////////////////////////////// End protocol secmig
```