International Journal of Networked and Distributed Computing, Vol. 3, No. 2 (April 2015), 99-108

Evaluation of A Secure Live Migration of Virtual Machines Using IPsec Implementation

Norshazrul Azman bin Sulaiman
Graduate School of Information Science
Kyoto Institute of Technology, Kyoto, Japan
Email: n-abs14@dsm.cis.kit.ac.jp

Hideo Masuda
Center for Information Science
Kyoto Institute of Technology, Kyoto, Japan
Email: h-masuda@kit.ac.jp

Abstract

Live migration enables administrators to perform online maintenance on host and virtual machines without interrupted service during fault occurrences. While migrating VMs between physical servers of different data centers over the network is considered, security is essential for securing the migration data which has not been considered in most cases up until now. This paper considers performing live migration under an IPsec implemented transmission channel to achieve a secured transmission of migration data between servers. IPsec is proven to be secure and can be implemented for live migration but due to the excessive amount of overhead produced by IPsec, proposed sophisticated tuning of MTU and MSS values resulted in a better migration performance.

Keyword: IPsec implementation, virtural machines, live migration

1. Introduction

Nowadays, practical application of virtualization technologies are becoming more and more common and with it, the trend for a more complex and larger scale of data center grows rapidly. As a result, cloud service administrators for example will not only be affixed to managing a single data center but also be needed to manage multiple data centers at the same time which includes linking and integrating of its resources. Thus, there will be demand for a distributed data center system that enables operation of multiple data centers as an integrated unit all simultaneously. A distributed data center benefits in various ways. When the systems and its data are widely distributed, auto-reconnection of available

data centers is possible. This benefits in providing continuity of service even during fault occurrences at one section from the currently connected data centers.

Moreover, procurement of needed resources not from a single installment of data center but also from multiple data centers simultaneously enables establishment of cloud services in a more efficient way. To reduce initial investment and operating costs for installation of server facilities such as data centers, virtualization of servers is becoming a trend. In order to provide continuity of services even during fault occurrences at data centers as mentioned in one of the benefits above, live migration of servers as one of the features of the virtualization technology is essential.

Live migration is defined as a process where dynamical transfer of running virtual machines from one physical server to another without interrupted running services and with little or zero downtime. By performing live migration, regardless of between servers of the same data center or between servers of widely distributed data centers, several vulnerabilities are disclosed where one of the major one is that the migration data is insecure due to the fact that migration data is transmitted along as clear text. This vulnerability allows attackers to sniff on the kernel memory, application state, and sensitive data such as passwords and keys during the migration. This security issue is of main concern if migration is performed between servers of different widely distributed data centers compared to migration of servers of the same data center. This is because there is a considerably low possibility for a sniffer to sniff on the migration data during a transmission of servers at a data center due to various security measurements applied at the data center. Therefore, migration of VMs between servers of the same data center emphasizes more on efficiency and performance instead of security so live migration data are transmitted as clear text.

Meanwhile, securing transmission channel between servers of different widely distributed data centers practically is not that easy and up until now, it is entrusted to the operators' conscience in performing the migration. Thus, when performing live migration of VMs across different data centers, instead of entrusting it completely to the operators' conscience, application of security measures is essential in protecting the migration data. Therefore, the need for a secured live migration between servers of different data centers surfaces to address and maintain trust establishment, confidentiality and integrity of the transmitted migration data.

In order to make the migration safe and secure, the migration data needs to be encrypted while authentication is also important as to authenticate and identify to whom the transmission is carried out. Besides that, for validation purposes, the other party receiving the transmission for the live migration to be executed needs to be authorized. Also, the integrity of the migration data needs to be intact after a completed transmission to avoid failure of VM start up at the destination platform. IPsec (Internet Protocol Security) and its protocols provide encryption, authentication and authorization while keeping the transmitted data integrity intact. Therefore, in this paper, implementing IPsec on the transmission channel for a secured transmission is proposed.

It is assumed that IPsec implementation greatly reduces the performance of the live migration such as an increased amount of packets transmitted and migration time along with a high CPU usage. Thus, several measures are proposed to increase the migration performance and are discussed in chapter VI. Besides that, the performance of a live migration performed under an IPsec implemented transmission channel is evaluated and discussed in chapter VII. Evaluation is performed by comparing the migration time, packets transmission and CPU usage between a live migration performed without IPsec implemented transmission channel and live migration performed with IPsec implemented transmission channel.

2. Related Work

2.1. *A Survey on Techniques of Secure Live Migration of Virtual Machine*

Reference [1] investigates attacks on a live migration and presents a survey in techniques of securing a live migration. It discusses the basic characteristics of a secure live migration and types of attacks or risks that a live migration of virtual machines is exposed to. It also discusses the currently proposed approaches in securing migrations along with its compatibility and implementation issues

when applying it with live migrations. In conclusion, there is still no integrated suitable approach that satisfies the needs of a secured live migration of virtual machines and that a framework that addresses vital security aspects is to be developed as future work.

2.2. *Downtime Analysis of Virtual Machine Live Migration*

Reference [2] investigates the downtime analysis of virtual machines during a live migration. They represented a result of an experimental study that analyzed downtime and average migration time based on a variety of factors that can have a significant impact on service availability. This research acts as a reference for this paper on how a live migration is affected by various factors and the resulting downtime and migration time in a detailed manner. They also divided the experiment into two categories regarding hypervisors as the migration platform and investigated how it affects the migration performance. The results of this research are used to investigate the applicability of VM live migration in the context of proactive fault management. Besides that, proposed ways of improving live migration performance is also taken into consideration in performing a secured live migration with minimal overhead costs and migration time.

3. Requirements and Considerations

3.1. *Requirements*

As mentioned above, performing a live migration under a secured transmission between two physical servers is essential. To preserve confidentiality and privacy of transmitted data, a secured live migration needs to satisfy the following characteristics.

1) Trusted Source and Destination Platforms: During a live migration, the source and destination platforms must be trusted and authenticated. The host machine operator must be able to identify and validate the destination host machine so that the migrated virtual machine arrives to a designated platform that is validated and identified. Therefore, an authentication system that authorizes both the source and destination platform is essential in performing a secure live migration.

2) Authenticated and Authorized Management Capabilities: This characteristic involves access control policies where an inappropriate one would allow an unauthorized user to initiate, migrate and terminate virtual machines freely without requiring authorization needs. Besides that, access control policies also decides and control access towards the hypervisor in controlling shared resources and operating the VMs. Therefore, steps need to be taken to allow only the authorized and authenticated user to authenticate, execute and control the migration.

3) Confidential and Unmodified Migration Data: As with all transmission channel in the Internet that are required to be safe and secure, the same thing applies with performing a live migration. In most cases, the migration protocol does not include the modules to encrypt and secure the transmitted migration data as it travels over the network. Performing a live migration under an insecure and unprotected transmission channel between the two physical servers over the network with migration data sent as clear text is dangerous. It opens the window for an attacker to sniff on the transmission channel by techniques such as ARP or DHCP poisoning [3], and follow the data stream to view and edit the migration data easily. Therefore, security measures to encrypt and secure the migration data need to be taken in consideration.

4) Detection of suspicious activities mechanism: Performing a live migration under an insecure and unprotected transmission channel results in it being susceptible to attacks. Prevention of these suspicious activities is also a measure that could be taken in order to secure and protect a transmission of a live migration of VMs.

3.2. *Usable Implementations for Securing a Transmission Channel*

1) Application Layer Security Implementation: In order to make the migration safe and secure, one way is to apply security measures over the transmission where the migration is executed. One of the popular measures taken in managing the security of a message transmission and operates at the application layer is the Secure Shell or SSH (Secure Socket Shell). SSH is easy to implement but theoretically requires much overhead and is not

meant to handle a connection with a lot of bandwidth requirements like migration of VMs.

2) Transport Layer Security Implementation: At the transport layer, SSL (Secure Socket Layer) is the popular security measure. However, the issue with SSL is that not all setups have implemented both server and client authentication which is an aspect that should be taken in consideration where it is necessary to authenticate both the source and the target destination hosts in a trusted connection where migration is performed. Moreover, there is the need to replace the transport so implementing SSL as a security measure for processes like live migration would require a more sophisticated modification of the SSL configurations.

3) Network Layer Security Implementation: Another measure of security is the IPsec that provides security directly on the IP network layer, securing everything that is put on top of the IP network layer. IPsec with its protocols provide authentication and authorization with AH protocol while keeping the data integrity intact. Meanwhile, the ESP protocol provides encryption for transmitted data. Because IPsec operates at the network layer and is transparent to applications, it has essentially no impact on the higher network layers. However, implementation of IPsec as the security measure for a secured transmission channel results in a great level of overhead due to a great amount of CPU processing of servers during encryption and processing of transmitted packets. Moreover, the resulting high granularity in the user authorization and authentication processes causes detailed controls to be difficult to carry out. During a normal data transfer of a single file, the transfer of data occurs at a constant rate. Meanwhile, during a data transfer of a live migration, different types of data such as memory, disk state and CPU state are transferred. Hence, it is possible that the rate of transfer varies for each different type of data transferred during the one session of live migration. Therefore, when IPsec is implemented, the overhead is assumed to be much bigger during a live migration compared to during a single file transfer because there might be further processings for the different types of data transferred. In this research, the secureness of implementing IPsec during a live migration of virtual machines between servers is observed and also the performance evaluation of the feasibility and compatibility along with tradeoffs on deploying IPsec as a security measure for a secure live migration is conducted.

4. System Configuration

This section explains about the system configuration and implemented IPsec mode and protocols used for the performance evaluation experiment. The main features of the system configuration and IPsec implementation are network settings, IPsec mode and protocol and system specifications and configuration as explained below.

4.1. *Network Settings*

For performance evaluation, migration traffic is isolated by placing the source host machine, destination host machine and the guest machines under a private VLAN (Virtual Lan Network). The network configuration settings is as shown in figure 1.

4.2. *IPsec Mode and Protocol*

The IPsec implementation used in this experiment is provided by the ipsec-tools [5] package which is a utility for IPsec implementation in Linux servers. The mode selected for implementation is the transport mode for end-to-end communication because not that many servers are included and transport mode is suitable for small scale networks, opposite to the tunnel mode. As for protocol, both AH and ESP protocol are combined and applied to provide authentication by the AH protocol and encryption by the ESP protocol.

4.2. *System Specifications and Configuration*

The system configuration and placement in the network is shown in figure 1. As shown in figure 1, ipsec-tools is implemented on both source and destination platforms. This is to provide a secured transmission channel between both the host and target host machine. The specifications of all the servers and guest machine are as shown in table I, II and III.

5. Secureness of Implementing IPsec on the Transmission Channel

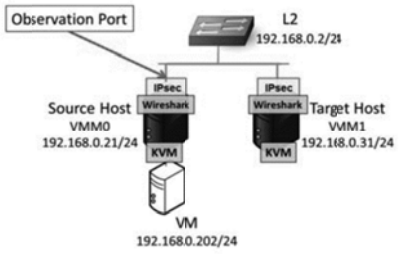

Fig. 1. Figure of System and Network Configuration

TABLE I. SOURCE HOST MACHINE SPECIFCATION

Category	Specification
Model	HP ProLiant ML110 G7
OS	Ubuntu 13.10 Server (3.11.0-15-generic_x86_64)
CPU	Intel(R) Celeron(R) CPU G530 @ 2.40 GHz
RAM	2 GB
HDD	250 GB
IPsec	IPsec-tools version 0.8.0

TABLE II. TARGET HOST MACHINE SPECIFICATION

Category	Specification
Model	HP ProLiant ML110 G7
OS	Ubuntu 13.10 Server (3.11.0-15-generic_x86_64)
CPU	Intel(R) Celeron(R) CPU G530 @ 2.40 GHz
RAM	2 GB
HDD	250 GB
IPsec	IPsec-tools version 0.8.0

TABLE III. VIRTUAL MACHINE SPECIFICATION

Category	Specification
OS	Ubuntu 13.10 Server (3.11.0-12-generic_x86_64)
CPU	QEMU Virtual CPU version 1.5.0 @ 2.40 GHz
RAM	512 MB
HDD	12 GB

This chapter explains about the analysis of the secureness of implementing IPsec on the transmission channel where live migration is performed towards the confidentiality and integrity of migration data.

In this section, migration data is captured using the network analyzer tool Wireshark [4] at the network interfaces used during the live migration as shown in figure 1 and the condition of the migration data before implementing IPsec is captured and analyzed. By following the stream of data sent, the raw data proved to be readable which is insecure. After implementing IPsec, the captured packet showed ESP as the header and the raw data could not be followed and unreadable which is encrypted and secure. This shows that the migration data is encrypted and processed by IPsec and that the packet protocol number is modified. This result also shows that the data is signed by AH and its integrity is kept intact. This proves that the migration data is secure while integrity and confidentiality is preserved.

6. Improving the Migration Performance

While security is achieved by using IPsec to secure the transmission channel for live migration to be performed, performance evaluation experiment is conducted to evaluate how IPsec implementation affects the migration performance. To improve the performance of a live migration executed under an IPsec implemented transmission channel, proposed ways are performing the live migration under a higher value of MTU and MSS. The proposed ways above are set as the variable in the performance evaluation experiment to investigate their relationships with the migration performance. The VM disk size, VM memory size, VM operating system is set as constant.

6.1. *Maximum Transmission Unit*

Here, the relationship between the value of MTU and the migration performance is observed. The variable MTU is increased accordingly from a lower MTU towards higher ones.

6.2. *Maximum Segment Size*

Here, how the value of MSS affects the migration performance is evaluated. For this experiment, MTU is configured and set to the highest value from previous experiment which is 8000 because it resulted in the highest performance from the previous experiment. Thus, MTU is set as constant at 8000 while MSS value is increased from the default value of 1460 to the highest possible value which is 7960.

7. Performance Evaluation

Migration is executed in two scenarios for evaluation purpose. The first scenario is where the VM is migrated from the source host (VMM0) to the destination host (VMM1) and the second scenario is where the migrated VM is migrated back to the original source host. This is to simulate the situation in real scenarios where a VM is temporarily moved to another host for maintenance on the original host and then the same VM is then migrated back to the

original host to continue it's service. Here, when migrating the same VM back to VMM0 from VMM1, any difference in the migration performance is observed between the two scenarios.

Evaluation of performance is done for both scenarios. Performance evaluation is divided into two categories which are live migration with and without IPsec implementation. For each of the category, the migration time, packet transmission and the amount of CPU usage of a migration performance are measured and how their relationship is with the changing variables of value of MTU and value of MSS is observed. In one of the experiments, a ping test and iperf test is conducted to evaluate the overhead due to IPsec implementation and the result showed that IPsec implementation resulted in a decreased network throughput by ten times and a decreased bandwidth by six times. Meanwhile, the ping test showed that the latency also doubled by implementating IPsec on the transmission channel.

7.1. *Migration Time*

In this category, migration time is defined as the time taken for one complete TCP conversation between the two hosts during the migration. It is measured using the network analyzer tool Wireshark where packets are captured from the network interface used for the migration and is observed. Migration time is measured and compared between the live migration executed with IPsec implementation against the live migration migration executed without IPsec implementation.

7.2. *Packet Transmission*

In this category, network traffic is observed by measuring the transmitted packets in the migration transmission between the two servers. Like migration time, the amount of transmitted packet is measured by capturing the packets of the transmission using Wireshark.

7.3. *CPU Usage*

In this category, how much CPU overhead occurs during an IPsec implemented live migration is observed by measuring the peak CPU usage (%) during the live migration for each category. In this experiment, the peak CPU usage is measured using the top command because it provides an ongoing look at the processors' activities in real time which is suitable as to measure and calculate the peak CPU usage during the migration. Here, the top command is executed for every second from the start of migration execution and terminated when the migration ends. The peak CPU usage of the source host machine is calculated during the migration process.

7.4. *MTU Effects on Performance Evaluation Results*

1) Migration Time: How migration time changes with MTU values is shown in figure 2. The results of migration time for executing a direct live migration without IPsec Implementation is as shown in figure 2. This figure shows that migration time decreases as the value of configured MTU increases. Meanwhile, figure 3 shows the differences of migration time after implementing IPsec on the transmission channel. This figure shows that IPsec implementation affects the migration time by increasing it greatly.

2) Packet Transmission: How packet transmission changes with MTU values is shown in figure 4. The results of the amount of packets transmitted for executing a direct live migration with and without IPsec Implementation is shown in figure 4. This figure shows that amount of packets transmitted decreases as the value of configured MTU increases. It also shows that IPsec implementation affects the amount of packets transmitted by increasing it greatly.

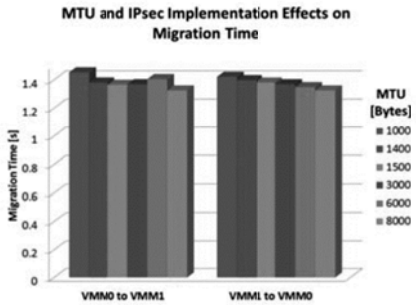

Fig. 2. Migration Time for Live Migration Without IPsec Implementation

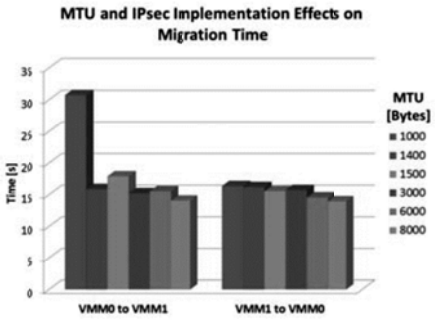

Fig. 3. Migration Time for Live Migration With IPsec Implementation

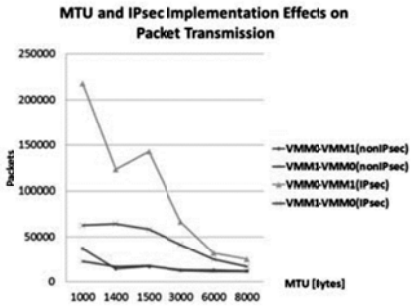

Fig. 4. MTU and IPsec Implementation Effects on Packet Transmission

1) Migration Time: How migration time changes with MSS value is shown in the figure 5. Here, results are presented according to the two scenarios

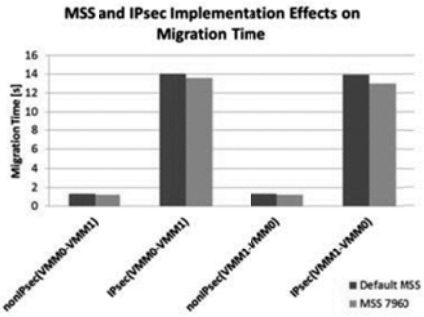

Fig. 5. MSS and IPsec Implementation Effects on Migration Time

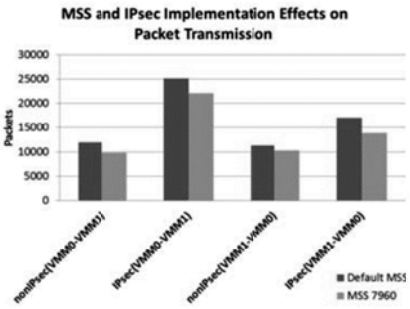

Fig. 6. MSS and IPsec Implementation Effects on Packet Transmission

7.5. *MSS Effects on Performance Evaluation Results*

In this section, the effects of MSS towards migration time and packets transmission is observed. This experiment is to observe the effects of MSS change on the migration performance. For this experiment, the results from previous section is used where the MTU with the least migration time which is MTU with the value of 8000, is selected and set as constant while MSS is changed from the default value of 1460 to the highest possible value. Here, the value of MSS must not exceed the value of MTU as there is the TCP and IP header which acquires for 40 bytes. Thus, the maximum value of 7960 is selected and compared with the default value Fig. 5. MSS and IPsec Implementation Effects on Migration Time Fig. 6. MSS and IPsec Implementation Effects on Packet Transmission of MSS to observe the relationship between the value of MSS and migration performance.

results are presented according to the two scenarios of migration from VMM0 to VMM1 and from VMM1 back to VMM0. The migration time results for migration from VMM0 to VMM1 and vice versa under changing MSS is shown in figure 5. From figure 5, it is understood IPsec implementation greatly increases the migration time and that a higher value of MSS results in a lower migration time.

2) Packet Transmission: How the amount of packets transmitted changes with MSS value is shown in the figure 6. Here also, results are presented according to the two scenarios of migration from VMM0 to VMM1 and from VMM1 back to VMM0. The results for amount of packets transmitted for migration from VMM0 to VMM1 and vice versa under changing MSS is shown in figure 6. It is understood that IPsec implementation greatly increases the number of packets transmitted and that

a higher value of MSS results in a lower amount of packets transmitted.

7.6. *Peak CPU Usage*

Based on the results of the previous sections, the optimum MTU and MSS value is selected and used to perform live migration of the least migration time

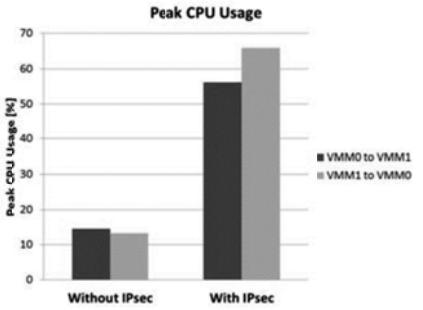

Fig. 7. Peak CPU Usage

and transmitted packets. Therefore, while MTU value is set to 8000 and MSS value is set to 7960, the peak CPU usage for both migration from VMM0 to VMM1 and VMM1 to VMM0 is measured and shown in figure 7. From figure 7, it is understood that the highest peak CPU usage result is when direct live migration is performed under IPsec implementation from VMM1 to VMM0.

8. Discussion

8.1 *IPsec Implementation Effects on Migration Performance*

Based on the results of the performance evaluation, performing live migration under an IPsec implemented transmission channel indeed reduces the performance of the migration as migration time and CPU usage is greatly increased. This could cause lower efficiencies of running services of both the virtual machine and the host machine. Therefore, in this paper, ways to increase the performance of a live migration performed under the IPsec implemented transmission channel are proposed which are increasing the MTU of each and every server's interfaces' MTU and MSS. Based on the results on chapter VII, the lowered performance of live migration under the IPsec implementation can indeed be increased by setting up a higher MTU and MSS values for the interfaces.

8.2 *MTU and MSS Effects on Migration Performance*

Based on the results in chapter VII, it is understood that a higher value of MTU increases the performance of a live migration executed under an IPsec secured implemented transmission channel by decreasing the time it takes for a complete migration of a VM. Like MTU, evaluation results also shows that a higher MTU with a larger MSS also increases the performance of the live migration by further decreasing the migration time. This is because for a transmission between hosts, if the datagram size is higher than the allowed MTU configured at the network interfaces, fragmentation occurs. In this case, if a receiving host receives a fragmented packet, it has to reassemble the datagram and pass it to the higher layer. This results in a longer migration time and thus decreases the migration performance. Besides a decreased migration performance, fragmentation also uses CPU procesing power which results in an excessive processing power used by the CPU, resulting in a high CPU usage. While increasing MTU to avoid fragmentation of IP datagrams increases the migration performance, a larger MTU of an ethernet frame does not mean that the data to be transmitted fits the configured MTU size perfectly but there is redundancy. This is because the transmitted data size passed along from the higher TCP/IP layer is limited by the MSS of the TCP segments. Thus, an increased size of MSS along with MTU results in a higher performance of live migration performed under an IPsec implemented secured transmission channel.

9. Future Work

9.1. *Enhancing the Performance of an IPsec Implemented Live Migration*

This experiment proved that by implementing IPsec, requirements 1) to 3) mentioned in chapter III can be fulfilled. However, as mentioned in chapter III, detection of suspicious mechanisms or activities during a live migration of VMs is also imperative but is yet to be implemented in this experiment. Besides that, as mentioned above, performing a live migration under an IPsec implemented transmission channel indeed results in a great amount of CPU overhead and affects the migration time greatly which reduces the migration performance and service efficiency. Thus, other possible means of improving the performance of a live migration performed under an IPsec implemented transmission channel needs to be developed and researched further in order to further increase the performance of the migration until it is practical enough to be applied at commercial data centers or cloud services.

9.2. *Application and Further Evaluation of Implemented Method of Live Migration*

Based on the results, IPsec implementation indeed meets the requirement for a secure live migration of virtual machines but with penalty due to the great amount of CPU overhead and longer migration time. There is also need to study the stages of the data transfer during a live migration to further understand the mechanism which could be applied to lower the overhead due to IPsec implementation to further increase the migration performance. With a secured transmission achieved by implementing IPsec, practically applying this method of live migration at data centers and evaluation of its performance need to be carried out to evaluate further its feasibility and compatibility in real case scenarios.

10. Conclusion

As for conclusion, performing a live migration under an IPsec implemented transmission channel is indeed secure because the confidentiality and integrity of the migration data remains intact but as penalty, a great amount of CPU overhead is produced due to encryption and packets processing and an increased migration time which decreases the migration performance. Proposed ways to reduce the penalty such as increasing the interface MTU and segment MSS does indeed reduce the penalty by decreasing the long migration time. Therefore, it can be concluded that in order to perform a secure live migration under an IPsec implemented transmission channel, the ethernet frame needs to be able to send a high size of payload to reduce the migration time and also the MSS needs to be set to the maximum value for optimum transmission of packets which could also reduce the migration time and increases the migration performance. Thus, the method of performing live migration under an IPsec implemented transmission channel is only practical when used with network infrastructures of high specification that can support faster and a more efficient transmission of data.

References

[1] J. Shetty, M. R. Anala and G. Shobha, 'A Survey on Techniques of Secure Live Migration of Virtual Machine', *International Journal of Computer Applications*, Vol.39, pp.35 - 39, February 2012.

[2] F. Salfner, P. Troger and A. Polze, 'Downtime Analysis of Virtual Machine Live Migration', *The Fourth International Conference on Dependability*, 2011.

[3] J. King and K. Lauerman, 'ARP Poisoning and Mitigation Techniques', A CSSTG SE Residency Program White Paper, http://www.cisco.com/c/en/us/products/collateral/switches/catalyst- 6500-series-switches/white paper c11 603839.html

[4] U. Lamping, 'Wireshark Developer's Guide For Wireshark 1.11', Wireshark,

http://www.wireshark.org/download/docs/developer-guide-a4.pdf.
[5] The KAME project, D. Atkins, M. Ludvig, E. Dreyfus, Y. Vanhullebus, M. Grooms, T. Teras, F. Senault and A. Kasparas, 'IPsec-Tools', *ipsectools*, http://ipsec-tools.sourceforge.net/