IUG Journal of Natural and Engineering Studies
Peer-reviewed Journal of Islamic University-Gaza

ISSN 2409-4587

IUGNES
Vol 24, No 1, 2016, pp 14-20

Secure Live Virtual Machine Migration by Proposed Security Center

Hatem M. Hamad[1], Alaaeddin B. AlQazzaz[1,*]

[1]Department of Computer Engineering, Faculty of Engineering, Islamic University of Gaza, Gaza Strip, Palestine

Received on (26-05-2015) Accepted on (28-08-2015)

Abstract

In this paper, we propose a Security Center (SC) to provide a secure environment for Live Virtual Machine (VM) migration which is defined as the movement of a virtual machine from one physical host to another. In the cloud systems, the migration has appeared based on the need of transferring VMs among resources. At most, researchers have focused on the performance of migration process; whereas the security aspects in migration have not been fully explored. So, we show how our proposed mechanism analyzes and fulfills the major security requirements for secure live VM migration in cloud environments to become protected against different types of passive and active attacks.

Keywords Virtual Machine, Security Center, Hypervisor, Security, Encryption.

الترحيل الآمن لجهاز افتراضي من خلال مركز أمان

ملخص

في هذا البحث، تم اقتراح إضافة مركز أمان يهدف إلى توفير بيئة آمنة ومناسبة لترحيل الأجهزة الافتراضية من جهاز خادم لآخر. حيث أنه وفي أنظمة السحابة الإلكترونية تظهر الحاجة لعملية الترحيل في ظروف عدة مثل توزيع الأحمال وأعمال الصيانة والأعطال المفاجئة. كما أن معظم الباحثين تركزت أبحاثهم على قياس مستوى السرعة وتحسن الأداء لعملية الترحيل دون الاهتمام بمستوى الأمان خلال هذه العملية، مما دفعنا لعمل الدراسة المقترحة والموضح فيها مركز أمان يحقق متطلبات البيئة الآمنة لعملية الترحيل في بيئة السحابة الإلكترونية لتكون مقاومة لمعظم طرق القرصنة المختلفة.

كلمات مفتاحية: جهاز افتراضي، مركز أمان، الحاضن للأجهزة الافتراضية، أمن المعلومات، التشفير.

* Corresponding author e-mail address: aqazzaz@hotmail.com

1. Introduction:

Virtualization technology was introduced in the late of 1960s by IBM[1]. Virtualization is defined as the abstraction of hardware resources to facilitate the sharing of resources well. So, virtualization helps enterprises reduce investments and operational cost. The term of virtualization means the ability to run entire virtual machine components, including the Guest Operating System (VMs), on another operating system called Host Operating System. The Hypervisor is the layer of software that emulates the hardware interface seen by the VM. The hypervisor completely controls system resources.

Virtual Migration (VM) migration is defined as the movement of virtual machine from one physical host to another. In the cloud environment, the VM live migration is introduced to obtain multiple benefits which mainly include high availability, hardware maintenance, fault takeover and workload balancing.

The unsecured live VM migration may open up the security risks and exposure for not only the migrated VM but also for other guest OSes running on that physical server. So, There is an intensive need for researching on security issues of the live migration process in the cloud.

In this paper, we highlight a number of proposed mechanisms for secure live migration and discuss the advantages and disadvantages of each one. Then, we show how our proposed mechanisms satisfies all security requirements and treats most attacks on live VM process; however, our suggested solution with Security Center (SC) guarantees the secure environment for live VM migration from one Hypervisor to another; where each hypervisor runs a local Security Process (SP) to communicate with the SC.

2. Live VM Migration Security Evaluation:

2.1 Analysis of Live VM Migration Security Attacks:

The live VM migration process is prone to active and passive attacks. Attacks on the live VM migration process are categorized into control plane, data plane and migration module classes [1].

2.1.1 Control Plane:

Hypervisor operations such as initiation and management of live VM migration should be authenticated and resistant against tampering. Furthermore, protection against spoofing and replays attack should be provided. The various vulnerabilities and threats on the control plane are identified as following:

- Incoming Migration Control:
  By initiating unauthorized incoming migrations, an attacker may cause guest VMs to be live migrated to the attacker's machine and gain full control over guest VMs.
- Outgoing Migration Control:
  Similarly, by initiating outgoing migrations, an attacker may migrate a large number of guest VMs to a legitimate victim Hypervisor, overloading it and causing disruptions or denial of service.
- False Resource Advertising:
  In an environment where live migrations are automatically initiated to distribute load across a number of servers, an attacker may be able to advertise falsely about available resources via the control plane. The attacker may be able to influence the control plane to migrate a VM to a compromised Hypervisor.

2.1.2 Data Plane:

Live VM Migration occurs in this plane, memory contents such as kernel states and application data transfer from one physical server to another. Attacker can use ARP spoofing or DNS poisoning techniques to launch Man in the Middle (MITM) attack on insecure communication channel. This introduces active and passive attacks during the migration process. Therefore, secure and protected channel must be used to minimize snooping and tampering attempts on migration data. The various vulnerabilities and threats on data plane are identified as following:

- Passive Snooping:
  Passive attacks against the data plane may result a leakage of sensitive information. By monitoring the migration transit path and associated network stream, an attacker can extract information from the memory of the migrating VM such as passwords, keys, application data, and other protected resources.
- Active Manipulation:
  One of the most severe attacks, an inline attacker may manipulate the memory of a VM as it is migrated across the network such as a Man-in-the-Middle attack may result in a complete and covert compromise of the guest OS.

2.1.3 Migration Module:

VM Migration functionality of hypervisor is implemented by software component which is known as the migration module. Vulnerabilities in migration module may allow attacker to compromise the hypervisor and any guest OSes as well. The Hypervisor component that implements live migration functionality must also be resilient to attacks. As the migration module provides a network service over which a VM is transferred, common software vulnerabilities such as stack, heap, and integer overflows can be exploited by a remote attacker to subvert

the hypervisor. Given that VM migration may not commonly be viewed as a publicly exposed service, the code of the migration module may not be scrutinized as thoroughly as other codes. If an attacker is able to compromise a hypervisor through its migration module, the integrity of any guest VMs running within the hypervisor, and any VMs that are migrated to that hypervisor in the future, may also become compromised.

2.2 Security Requirements for VM Migration:

We have identified the following security requirements for the live VM migration process [2].

1. Integrity Verification of Platform:
   The destination platform cryptographically identifies itself to source for trust establishment.
2. Authentication:
   Attacker can launch MITM attack using techniques such as route hijacking or ARP poisoning in the migration process. In order to avoid MITM attacks on live VM migration, source and destination platforms must mutually authenticate each other.
3. Authorization (Access control policies):
   Appropriate access control policies must be provided to secure the live VM migration process. An unauthorized user/role may launch VM initiate, migration operation. Unauthorized activities can be prevented by using access control list (ACL's).
4. Confidentiality and Integrity of VM during migration:
   An encrypted channel must be established so that an attacker cannot get any information from VM contents and modification of contents can be properly detected. This will help to avoid active attacks such as memory manipulation on live migration and passive attacks such as leakage of sensitive information.
5. Replay Resistance:
   Attacker can capture traffic and replay it later to get authenticated in VM migration process. Therefore, live VM migration process should be replay resistant. Nonce's can be used to prevent replay attack in migration.
6. Source Non-Repudiation:
   Source host cannot deny from VM migration operation. This feature can be achieved by using Public key certificate.

3. Related Work:

One approach for secure live VM migration against attacks discussed is to assign a small group of VMs or even a single VM to its own host-based Virtual LAN (VLAN) [3]. VLAN is basically a segmentation and isolation tool. The VLAN isolates migration traffic from other network traffic and defines a secure transmission channel for migration. A major drawback of VLAN-based security approach is the growth in complexity and administrative costs as the VM population grows. The complexity lies in setting up and maintaining VLANs for each VM, synchronizing VLANs, configuring the virtual and physical switches, troubleshooting and fix configuration errors, manage the growth and complexity of ACLs as number of VMs increased, ensure compatibility between physical network and virtual network security policies. VLAN-based security approach does not support for any of security requirement.

Network Security Engine-Hypervisor (NSE-H) approach [4] is based on hypervisors included with network security engines to eradicate intrusions occurring in virtual network. So, protecting virtual machine (VM) residing in virtual network. NSE includes firewall, intrusion detection systems and intrusion prevention system to provide security to virtualized environment. They include intelligent packet processing capability built in them. The NSE firewall work in state full way. They maintain security context for each packet and make decisions based on security context and packet content. but the downtime is increased according to two factors. The first is that the SC iterative copy processing. The second is that SC migration process competes with VM migration process for computing resources, and this slows down VM migration process. This approach does not support for any of security requirement.

Role based migration approach [2,5] is based on use of Intel vPro and TPM hardware for protection of migration process. It consists of Attestation Service, Seal Storage, Policy Service, Migration Service and Secure Hypervisor Components. These features help the scheme to secure Virtual Machine during migrations between open platforms. Secure migration includes three key steps, which are building trustworthy container for virtual machine, securing VM Migration, and securing hypervisor. The drawback of role-based secure migration; it cannot be integrated with current deployed infrastructure because changes are required at software and hardware levels.

Virtual TPM -vTPM based migration protocol [2,6] is the integration of Trusted Computing technologies into virtualized computing environments enables the hardware-based protection of private information and detection of malicious software. Their use in virtual platforms, however, requires appreciate virtualization of their main component, the Trust Platform Module (TPM) by means of virtual TPMs (vTPM). The challenge here is that the use of TPM virtualization should not impede classical platform processes such as VM migration. In fact, there is typically a single TPM module per hardware platform. Therefore, its

functionality has to be efficiently shared by the virtual machines (VM) running on the same hardware. This is typically achieved by virtual TPMs (VTPMs) that mimic the interface and functionality of the hardware TPM. One important challenge is to realize vTPM that comply with TPM specifications while not impeding platform processes such as VM migration. The main drawbacks are: it is not support live migration and the keys of vTPM are stored outside the TMP, therefore prone to leakage and unauthorized modification. The vTPM state is also migrated, so it is an overhead and increases the downtime and total migration time and performance decreasing. This approach does not support authorization security requirement.

VM Migration using SSH tunnel between proxies is the approach [7] proposed by consisting of inter cloud proxies, secure channel between proxies, migration with non-shared storage and virtual network migration components. Inter cloud proxies used to restrict access to those hosts which are used in inter cloud VM mobility. The proxy server at the source and destination clouds communicates with each other and hides the details of source and destination Cloud hosts. SSH tunnel is established between proxies for secure VM migration, VM states and memory is transferred during the migration process. The main drawback of this approach is not support authorization; Furthermore it requires port forwarding on firewalls.

RSA with SSL [8] based Secure VM migration process is consists of three steps. First, load calculation on physical host then RSA with SSL protocol is used for authentication and encryption mechanism as well as for protection and privacy of memory contents. Finally, Pre-copy or Post-copy migration techniques used for live migration between source and destination. The drawbacks of this approach: RSA based authentication required public keys of all hypervisors for authentication in migration process. So, the management of Public keys difficult. This approach does not comply with authorization and integrity verification of platform. This approach increases the migration time and degrades the performance. Trusted Token (TT) based migration approach [9] consists of set policy, implement migration policy and audit migration components. User's policy contains the acceptable Trust Assurance Level (TAL) value of the target cloud platform for VM migration. TT is a trust credential which contains TAL value issued by Platform Trust Assurance Authority (PTTA) based on hardware and software components of platform. VM migration occurs if TAL value in TT of destination platform is acceptable against the TAL value of user migration policy.

X.805 security standard investigates attacks on live virtual machine migration [10]. The analysis highlights the main source of threats and suggests approaches to tackle them.

X.805 standard defines three security layers (applications, services and infrastructure), three security planes (end user, control and management) which are identified based on the activities performed over the network, and also eight security dimensions to address general system vulnerabilities (Access Control, Authentication, Non- Reputation, Data Confidentiality, Communication Security, Data Integrity, Availability, and Privacy).

4. Proposed Mechanism:

We suggest a secure architecture for live Virtual Machine (VM) migration through a proposed Security Center (SC) infrastructure which will be added to the Virtual Machine (VM) environment. The proposed mechanism should satisfy all security requirements for live VM migration and should be resistant to different types of attacks described in *Live VM Migration Security Evaluation Section* . The Architecture of Secure Live Migration shown in figure 1 where the SC consists of different modules: Access Control List (ACL), Update Center, Certification Authority (CA) and Auditing Database (ADB). The ACL is responsible for allowing/denying operation for each VM migration request and is responsible for preventing the VM migration flooding. Update Center is responsible for updating all hypervisors software in the environments, keeping them up-to-date and installing all available security patches by both periodically and manually. CA is responsible for storing all hypervisor certificates which are used in authentication, confidentiality and is responsible for renewing the certificate when it is expired. Finally, All the events should be recorded into ADB for historical and reporting purposes. Public keys, symmetric key, nonce, sequence number and hashing are techniques that will be used to ensure the integrity of the platform and to ensure that the source and destination are authenticated and certified the confidentiality (the details explained later).

The proposed algorithm is divided into two parts; firstly; the registration part. The second part is the secure live (VM) migration process from Hypervisor (Hyp1) to Hypervisor (Hyp2) through SC.

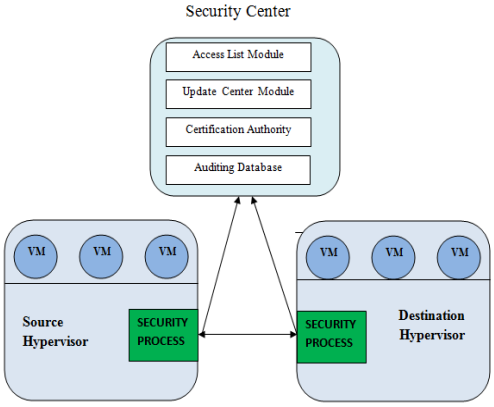

Figure 1 *Architecture of Secure Live Migration*

The secure live (VM) migration process passes into three phases as following:

1. Hyp1 communicate with SC;
2. SC communicate with Hyp2;
3. Hyp1 communicate with Hyp2;

In the registration process part, before the hypervisor installation finished on a physical machine, the hypervisor should be registered in the SC to complete the installation. To do it; the hypervisor should be sent to the SC a ***register_request*** concatenates with ***sequence_number*** (to protect the order) and ***time_stamp_random_nounce*** (to protect the reply attack and used as a hash key). The hashing techniques used in all messages between the hypervisors and SC to guarantee the message integrity. The SC receives the ***register_request*** and checks the integrity through hashing. The SC responds with SC's certificate; the hypervisor stores the SC certificate and extracts the SC public key. Now, the hypervisor is responsible for creating a local Security Process (SP) which has a specific SP ID. The local SP is responsible for providing a secure manner to exchange all messages with the SC and migration traffic with other SPs on other hypervisors. After the SP creation, the SP collects the hypervisor information (HI) for example, IP address, MAC Address, SP ID and hypervisor version. SP uses this information to generate a hypervisor certificate and store the private key locally. Only SP has a permission to read the private key. The SP creates a new message containing ***its certificate encrypted*** by SC public key (only SC can decrypt it) concatenated with ***HI which is signed by the hypervisor private key and encrypted the signed message by SC public key*** (only SC can decrypt it which certifies the confidentiality and the sign process certifies the hypervisor authentication which ensures the message is coming only from this hypervisor because the hypervisor is the only which has its private key) .Also concatenated with ***sequence_number*** and ***time_stamp_random_nounce***. Now; SP is ready to send this message to the SC.

The SC receives a message, extracts the hypervisor certificate and public key and creates a new record for the hypervisor into CA module including (Hyp ID, local SP ID, digital certificate, certificate status, certificate age, IP address, MAC address and Hypervisor software version). The SC sends ***OK_registered*** message back to the hypervisor to confirm the completed registration process. The ***OK_registered*** was encrypted using the hypervisor public key.

When launching a new VM on a hypervisor, the administrator should create a proper record into the ACL table into SC to control the VM Migration operations. As shown in table 1, the example of SC ACL table, the destination hypervisor is the result of the other function like CalculateLoad() which returns a trust (registered in SC) hypervisor with lowest load.

Table 1 *Example of SC ACL*

Hyp_ ID	VM_ ID	Allowed_Users	Destination	Age (sec)
1	1	Admin	CalculateLoad()	0
2	1	Aqazzaz; Admin	CalculateLoad()	0
Otherwise deny				

In the ACL table, the Age field is required to prevent migration flooding of specific VM more than once in specific interval. So, the migration requests for specific VM should not occur more than once in specific interval. This interval needs other experiments to determine it. When generating a new ***migration request***, the Age value is set. So, any new migration request for the same VM will not be allowed during this interval.

In secure live VM migration process part, the first step is starting when the source hypervisor communicates with SC through the hypervisor local SP. Hypervisor activates the local SP. The SP sends ***migration_request*** including (SP ID, VM ID, user) ***which is signed by the hypervisor private key and the signed message is encrypted by the SC public key***. SP concatenates the requested message with ***SP ID, sequence_number*** and ***time_stamp_random_nounce***. The SP sends this message to the SC.

The SC receives the message and checks ACL. If it is OK, then SC sets the Age value and continues with lookup into CA table to match the received SP ID with SP ID record which is previously stored; then determines the specific certificate and extracts the source hypervisor public key. SC decrypts the message with the SC private key and unsign it with the source hypervisor public key. Then it checks the updated center module; the destination hypervisor software version should be equal or higher than

the source version. If not, update the destination hypervisor version through destination local SP. After that, the SC generates symmetric Session Key (s_key) to encrypt all traffics between source and destination. We proposed symmetric key encryption because it is more efficient and faster than public/private keys.

The SC creates ACK message including s_key ***which is signed by the SC private key and encrypted the signed message by source hypervisor public key***, concatenated this message with**,** ***sequence_number*** and ***time_stamp_random_nounce***. The SC sends the ACK message back to source hypervisor. Now the source SP has an s_key and changes its mode into ***wait_mode*** for waiting ***ready_message*** from the SC.

The second step, the SC communicates with the destination hypervisor through the destination hypervisor SP. The SC creates a ***ready_migrate_request*** including (source SP ID, VM ID, user, source IP, s_key) ***which is signed by the SC private key and the signed message is encrypted by the destination hypervisor public key*** which is got from CA table on SC. This requested message is concatenated with ***sequence_number*** and ***time_stamp_random_nounce***. The SC sends it to the destination SP on the destination hypervisor. Now, the source and destination SP have the s_key to encrypt all migration traffic with pre-copy migration method.

The destination SP creates ***ready_message which is signed by the destination hypervisor private key and the signed message is encrypted with the SC public key***. This message is concatenated with ***sequence_number*** and ***time_stamp_random_nounce***. This message is sent to the SC.

The SC receives the ***ready_message and is signed by the SC private key and the signed message is encrypted with source hypervisor public key***. This message is concatenated with ***sequence_number*** and ***time_stamp_random_nounce***. This message is sent to the source SP. The SC starts monitoring the migration process and begins counting down the Age value in ACL table.

The source SP receives ***ready_message*** and creates ***hello_message*** which is encrypted with s_key. This message is concatenated with ***sequence_number*** and ***time_stamp_random_nounce***. This message is sent to the destination SP. The destination SP receives ***hello_message*** and responds with a new ***hello_message*** which is encrypted with s_key. This message is concatenated with ***sequence_number*** and ***time_stamp_random_nounce***. This message is sent to the source SP.

The source SP receives the ***hello_message*** and validates it. Now, the source SP starts the secure live VM migration using pre-copy. All traffics of the VM migration transmission are encrypted with s_key symmetric encryption algorithm to exchange the migration traffic between hypervisors.

5. Security Analysis and Discussion:

The proposed algorithm discusses different types of security issues related to live VM migration process.

A. **Incoming Migration Control**: means initiating unauthorized incoming migrations. The proposed algorithm solves it by proposing access control list module (ACL) to determine the authorized users who have permission to initiate the migration.

B. **Outcoming migration control**: means migrating a large number of guest VMs to a legitimate victim Hypervisor, overloading it and causing disruptions or a denial of service. The proposed solution is to create the Age value into the ACL table to prevent the flooding of VM migration requests. Also the destination hypervisor is determined by the CalculateLoad() function which returns a trust hypervisor with lowest load.

C. **False Resource Advertisement**: means the attacker may be able to falsely advertise available resources. The proposed algorithm
solves it by the registration stage, which guarantees all trust and healthy hypervisors had been registered into SC and all migration processes will occur only through SC.

D. **Passive Attack:** causes leakage of sensitive information by monitoring the migration transit path and associated network stream. The proposed algorithm is resistant to the passive attack by providing a secure manner by an asymmetric encryption algorithm to exchange all messages between hypervisors and the SC. As well as using

E. **Active Attack (Man-in-the-Middle):** means the attackers have the ability to intercept messages between the source and the destination. The proposed algorithm resists the active attack by using hashing techniques to detect any manipulating into memory of a VM as it is migrated across the network. In addition, all messages are signed by the sender's private key and then encrypted using the receiver's public key to certify the concept of mutual authentication.

F. **Migration Module Bugs:** means the migration module has common software vulnerabilities such as stack, heap, and integer overflows which can be exploited by a remote attacker to subvert the hypervisor. The proposed algorithm solves it by Update Center Module into SC which is responsible for keeping all registered

hypervisors up-to-date and installing all available security patches on the hypervisors when updated.

G. **Integrity Verification of Platform**: means the platform cryptographically identifies itself to other for trust establishment. The proposed algorithm solves it by the registration stage where each hypervisor identifies itself with its certificate to the SC and creates a specific record for each hypervisor into CA.

H. **Authentication**: The proposed algorithm designs the authentication through an encryption by private keys. When the message is signed by the sender's private key ,it guarantees the sender because he, and only he, is the owner of the private key.

I. **Confidentiality:** means a privacy; the proposed algorithm solves it through encryption by the public key. When the message is encrypted by the receiver's public key, that guarantees only the receiver has the ability to decrypt the message because he, and only he, is the owner of private key. Another technique used is a symmetric encryption key to encrypt all migration traffics between the source and destination hypervisors.

J. **Replay Attack Resistant**: means attacker can capture traffic and replay it later to get authenticated in VM migration process. The nonce's are used to prevent replay attacks in migration, which is combined with a time stamp in our proposed algorithm.

K. **Auditing:** The proposed algorithm provides an auditing database for recording all events for historical and reporting purposes. This database is hosted into SC.

Conclusion and Future Work:

In this paper, we highlight a number of proposed mechanisms for secure live migration and discuss the advantages and disadvantages of each one . We show that our proposed mechanisms which satisfy all security requirements and treat most attacks on live VM process ; however, we have complemented the existing work which use Security Center (SC) to guarantee the secure environment for live migration from one Hypervisor to another; where each hypervisor runs a local Security Process (SP) to communicate with the SC. The SC is responsible for all security requirements and has CA module to manage the Hypervisor's certificates. For authentication; the messages are signed with the sender's private key. For messages integrity; the messages are hashed (digitized). For authorization, the SC has ACL module. For confidentiality, the messages are encrypted by symmetric key encryption where it is more efficient than Asymmetric key encryption. For reply attack resistance, we used nonce's and time stamp. For auditing, all events will be recorded into ADB. However, the implementation and actual performance evaluation of the proposed algorithm are left for future work.

References:

[1] Oberheide, J., Cooke, E., and Jahanian. F. Empirical Exploitation of Live Migration of Virtual Machines. *Proceedings of Black Hat DC convention*, March 24 (2008).

[2] Shetty, J., Anala, M. R, Shobha, G. A Survey on Techniques of Secure Live Migration of Virtual Machine. *International Journal of Computer Applications*, ***39(12)***, (2012) 0975 – 8887

[3] Open Stack Security Guide, (2013). http://docs.openstack.org/security-guide/security-guide.pdf

[4] Xianqin, C., Han, W., Sumei, W., Xiang, L. Seamless Virtual Machine Live Migration On Network Security Enhanced Hypervisor. *International Conference on Broadband Network and Multimedia Technology*, Beijing, IEEE, 18-20 Oct (2009).

[5] Wang, W., Zhang, Y., Lin, B., Wu, X., and Miao, K. Secured and Reliable VM Migration in Personal Cloud. *2nd International Conference on Computer Engineering and Technology*, Chedgdu, IEEE, 16-18 April (2010).

[6] Danev, B., Masti, R. J., Karame, G. O., and Capkun, S. Enabling Secure VM-vTPM Migration in Private Clouds. *Proceedings of 27th Annual Computer Security Applications Conference*, Orlando, Florida 05-09 Nov (2011)

[7] Nagin, K., Hadas, D., Dubitzky, Z., Glikson, A., Loy, I., Rochwerger, B., and Schour, L. Inter-Cloud Mobility of Virtual Machines. *International Conference on Systems and Storage*, Haifa, Israel, May 30-June 01 (2011).

[8] Patil, V. P., and Patil, G. A. Migrating Process and Virtual Machine in the Cloud: Load Balancing and Security Perspectives. *International Journal of Advanced Computer Science and Information Technology*, ***1(1)***, (2012) 11-19.

[9] Aslam, M., Gehrmann, C., and Bjorkman, M. Security and Trust Preserving VM Migrations in Public Clouds. *International Conference on Trust, Security and Privacy in Computing and Communications*, Liverpool, IEEE, 25-27 June (2012).

[10] Aiash, M., Mapp, G., and Gemikonakli, O. Secure Live Virtual Machines Migration: Issues and Solutions. *28th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, UK, IEEE, 13-16 May (2014).