A Secure Architecture for Inter-cloud Virtual Machine Migration

Tayyaba Zeb[1](✉), Abdul Ghafoor[1], Awais Shibli[1],
and Muhammad Yousaf[2]

[1] School of Electrical Engineering and Computer Science,
National University of Sciences and Technology, Islamabad, Pakistan
{11msccstzeb,abdul.ghafoor,awais.shibli}@seecs.edu.pk

[2] Riphah Institute of Systems Engineering,
Riphah International University, Islamabad, Pakistan
myousaf@ieee.org

Abstract. Virtual machine migration is an important tool that can be used in cloud computing environment for load balancing, disaster recovery, server consolidation, hardware maintenance, etc. Currently a few techniques have been proposed to secure the virtual machine migration process. However, these techniques have number of limitations e.g. lack of standard access control, mutual authentication, confidentiality, non-repudiation and integrity of VM data. Some of the techniques provide security services such as mutual authentication using TPM (Trusted Platform Module), however, not all the hardware platforms yet possess the TPM capability. This limits the deployment of such solutions in legacy systems. The architecture, presented in this paper, attempts to overcome these limitations with existing hardware support. In particular, we designed a secure and efficient protocol that migrates virtual machine from source cloud domain to destination cloud domain by considering fundamental security services such as confidentiality, integrity, standard access control and non-repudiation.

当前的方案的缺陷：
1.缺少标准访问控制、成熟的认证、隐私性、不可否认性、VM数据的完整性。
2.需要TPM的支持

Keywords: Authentication · Authorization · Cloud computing · Confidentiality · ECDH · Integrity · SHA-256 · Virtual machine migration

1 Introduction

Virtual machine (VM) migration is an administrative tool supported by many virtualization software or Virtual Machine Monitors (VMMs). For example XEN [1], VMware [2], KVM [3], Hyper-V etc. provide flexible migration and management of VMs. In distributed computing environment such as cloud computing, VM migration allows transfer of complete operating system that runs inside a VM along with applications running on it, from one physical location to other. The service of VM migration aids in load balancing, elastic scaling, fault tolerance, disaster recovery and easier hardware maintenance [4–6]. VM migration can be of two types i.e. Offline or Cold VM migration 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

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 24–35, 2015.
DOI: 10.1007/978-3-319-23829-6_2

running on being migrated VM might face varying downtime during final synchronization. In offline migration, VM is paused or stopped at source, then sent over the network and resumed at destination. Migration of VMs is a useful tool in data centers and cloud environments in which a virtual machine is migrated from one storage location to another for the sake of load balancing or in a scenario where a hardware failure is imminent.

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 [7]. 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 whereas lack of proper access control may result in unauthorized VM migrations causing release of sensitive data to adversary. Also, large number of unsolicited migration requests may cause DoS or clogging attack [8, 9]. As these security issues have not yet been dealt properly therefore, there is a need to design some comprehensive security solution for the VM migration process. In this regard, we propose a protocol for secure virtual machine migration among clouds that preserves confidentiality, authenticity and integrity of virtual machine before, during and after transit; both on source and destination platform.

The proposed approach provides the authenticated and authorized migration of virtual machine from source cloud domain to destination cloud domain. In source domain, system administrator is first authenticated and authorized to initiate the VM migration process. The designed approach provides the access control for initiating and responding to the VM migration process thus preventing unauthorized VM migration. The migration request is evaluated against the policy rules that are set using XACML 3.0 (eXtensible Access Control Markup Language) [10]. Source and destination cloud mutually authenticate each other and validate migration request. This helps 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 performed 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-256 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.

Rest of the paper is organized as follows: Sect. 2 covers related work and limitations of existing techniques. Section 3 discusses proposed architecture and protocol description. Section 4 presents the discussion on performance modeling in terms of delay of the proposed scheme and in the end Sect. 5 concludes the paper.

2 Related Work

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 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). 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, however, no complete architecture is presented that comprehensively addresses these issues.

Timothy et al. discussed how active connections of applications can be seamlessly redirected while migrating a virtual machine from an enterprise to a cloud over the WAN [11]. The CloudNet platform developed by authors uses VPLS (Virtual Private LAN Services) that bridges the VLANs at the cloud and the enterprise thus enabling open network connections to be seamlessly redirected to the VM's new location. The optimization technique and algorithm helped to reduce the bandwidth issue and 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. Authors used layer 2 VPN's for protecting transmission channel in order to provide the confidentiality service. Analysis of the processes that allow live migration of VMs over long-haul networks is presented in [6]. The paper explains how VMs can be migrated across geographical distances transparently to applications. Optimization techniques through data de-duplication for a group of migrating VMs is presented in [12].

Security issues in VM migration are being studied in recent years. A few protocols are proposed for secure migration of VMs. Attacks on data and control plane of migrating VM are categorized and implemented in [4]. Authors demonstrated that integrity of data can easily be harmed during migration. However, they did not provide solution for it which drew our major inspiration to devise a secure protocol for VM migration. Security issues regarding the protected processes running inside a VM are discussed in [5]. 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.

An approach that checks for software updates and scans virtual machines for known security vulnerabilities is presented in [13]. Similarly advanced cloud protection system provided by [14] is integrated into virtualization software (virtual machine monitor) to monitor the integrity of guest VMs. It provides integrity of VMs and cloud's critical infrastructure. However both of above mentioned approaches do not help in secure migration. The process of live migration of virtual machine using KVM (Kernel based Virtual Machine) was carried out in [15]. The authors state that KVM and Xen expose entire machine state i.e. operating system kernel and applications during the process of migration however, they do not provide solution for it.

Two major security issues of VM migration i.e. platform authenticity and confidentiality of VM data during transit are discussed in [16]. For platform authenticity, authors proposed a Platform Trust Assurance Authority (PTAA) which assigns trust levels to platforms based on their configurations. 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, Trust Assurance Level (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 in [17]. 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-migratable keys. In both of the above mentioned approaches, the protocols work only if the infrastructure has TPM support, thus introducing the hardware dependency. Moreover, these approaches also lack standard access control for the process of migration.

Most of the existing solutions for VM migration are either TPM based and fail to work with legacy hardware, or they cater VM migration security issues individually. The 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 access control, mutual authentication and data integrity. For example, lack of access control may cause unauthorized VM migration resulting in VM to be migrated to a platform under the control of an attacker, even if VM was encrypted during transmission [18]. The focus of proposed solution is to address the limitations of existing techniques and devise a comprehensive protocol for securely migrating the virtual machine in an authenticated and authorized process. Moreover, the approach presented in this paper does not introduce hardware dependency and works with legacy hardware support.

After a deliberate review of literature, following security requirements are considered while designing our proposed solution:

- Standard Access Control for VM migration process
- Mutual Authentication of source and destination domain
- Confidentiality of VM data in transit
- Integrity of VM data in transit
- Non-Repudiation of migration process

The approach presented in this paper attempts to cover all the above mentioned security issues as a single comprehensive solution.

3 Proposed Inter-Cloud VM Migration Architecture

As shown in Fig. 1, in the proposed architecture, the process of inter-cloud virtual machine migration consists of following steps:

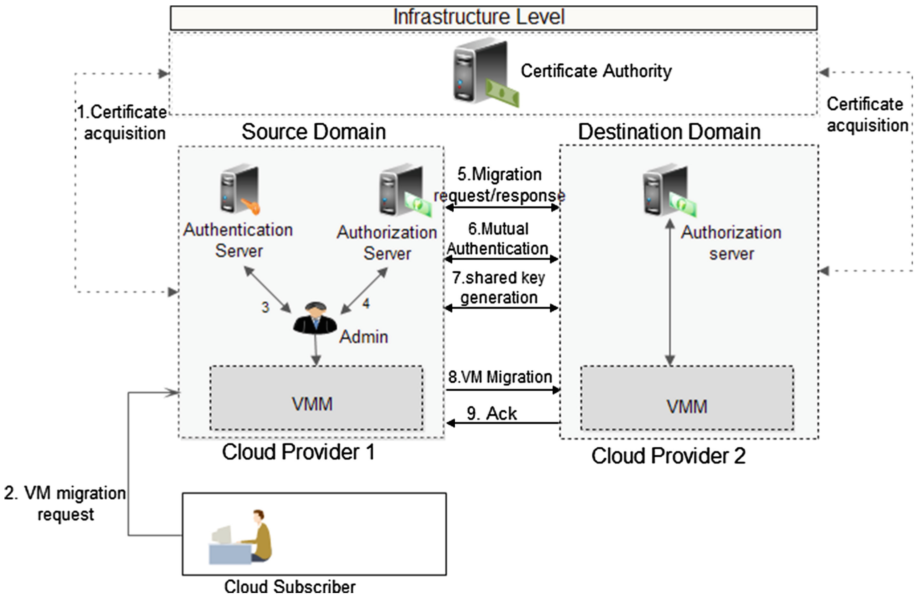

Fig. 1. Proposed architecture for secure migration of virtual machine

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.

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) [19]. 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 [20] and then this encrypted data is sent to the destination cloud domain. The integrity of VM data during transit is ensured using SHA-256 hash algorithm [21]. The reason for using SHA-256 is its recommendation by the standard for the message size up to $(2)^{64}$ bits. As migratable VM data is far less than this size therefore SHA-256 is sufficient for this purpose.

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.

Figure 2 shows the message exchange between different components of source and destination cloud domain for secure VM migration process.

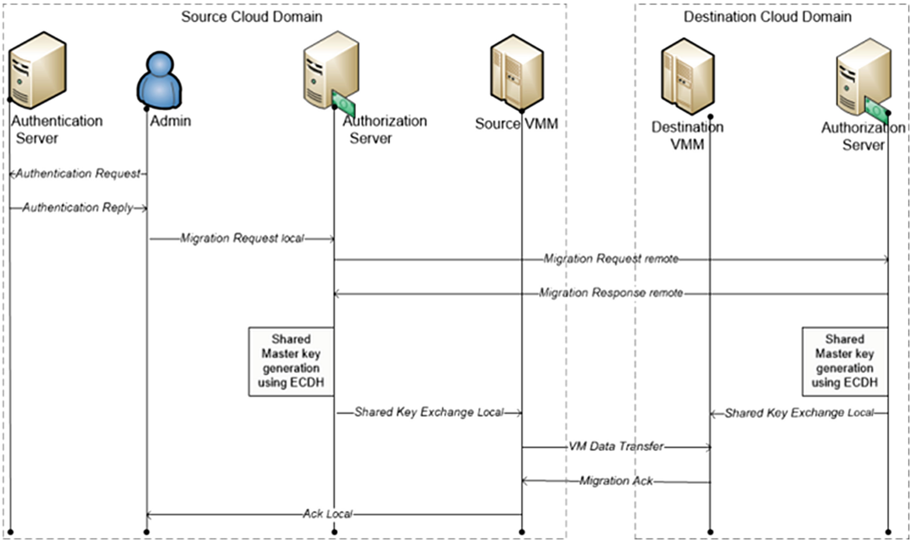

Fig. 2. Message exchange for secure migration of virtual machine

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 to the Authorization Server. 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; 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 migration response message to source domain.

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

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

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

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

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 authorization servers at both ends exchange the session key with Virtual Machine Monitor (VMM) at the respective ends. VMM of source domain encrypts the VM states using this session key (SK_S) and a SHA-256 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.

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

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

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.

4 Performance Modeling

As delay involved in migrating the virtual machine across the wide area network 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
D_{PL} = 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
D_{PW} = Propagation Delay in WAN

Figure 3 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 4 shows the effect of propagation delay between two datacenter locations over the migration delay. The graph shows that the propagation delay has linear affect over the migration delay i.e. with the increased the propagation delay the delay involved in migrating the virtual machine from one datacenter location to another datacenter location over the WAN will linearly increase. The factors that may affect the

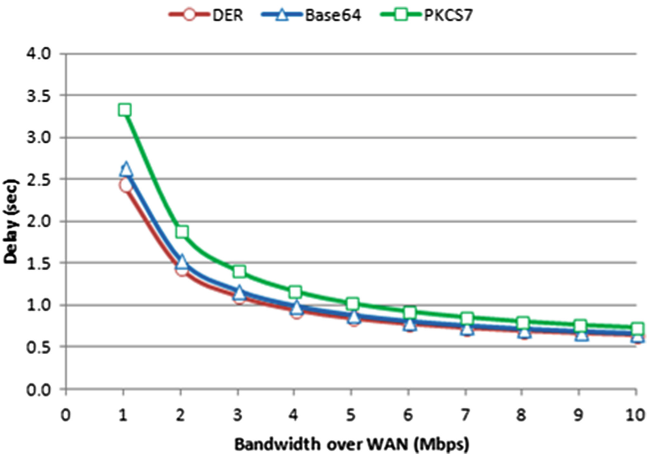

Fig. 3. Delay for migrating virtual machine with increasing bandwidth over WAN link

propagation delay include the available bandwidth, geographical distance between two datacenter locations, congestion over the WAN path, etc. Depending upon these mentioned parameters, propagation delay over the Internet usually varies between 100 ms to 350 ms and overall migration delay that is affected from this propagation delay varies only from 1 s to 2 s.

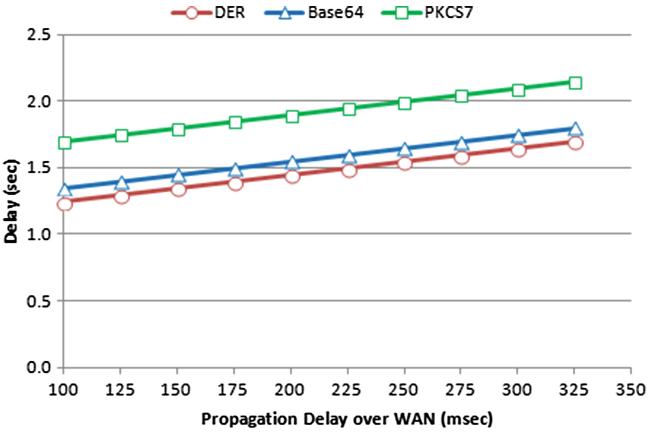

Fig. 4. Delay for migrating virtual machine with increasing propagation delay over WAN link

Figure 5 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.

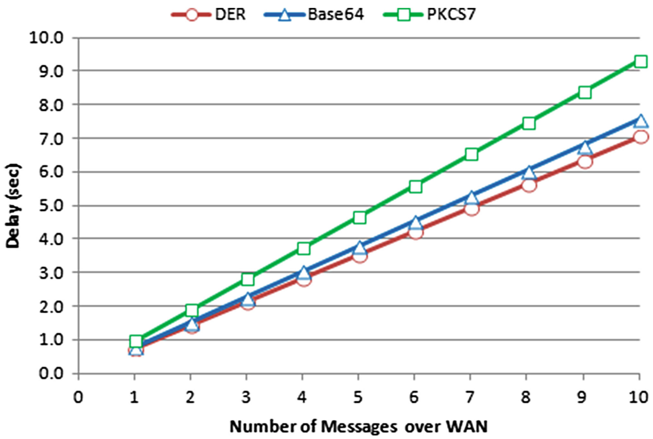

Fig. 5. Delay for migrating virtual machine with increasing number of control messages over WAN link

Figure 6 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 [22]. Similarly Transport Layer Security Protocol (TLSv1.2) takes at least nine messages for this purpose including the Ack messages [23]. 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.

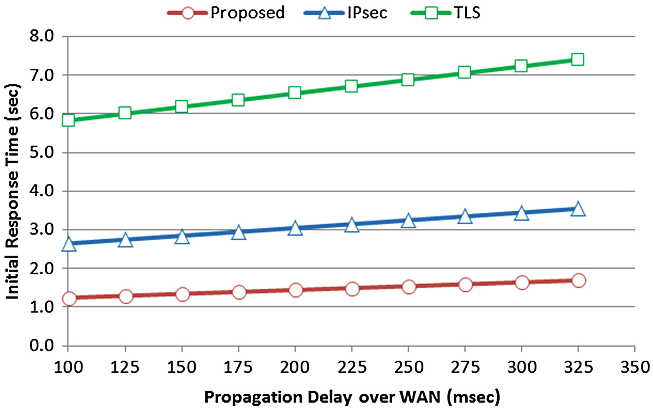

Fig. 6. Comparison of initial response time of the proposed architecture with IPsec and TLS

Result of Figs. 3, 4, 5, and 6 shows that out of number of factors e.g. available bandwidth, distance between two datacenter 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.

5 Conclusion

In this paper, the security requirements for secure migration of virtual machine, are analyzed and it is identified that lack of single security feature may arise many other vulnerabilities in the process of VM migration. The approach presented in this paper provides various security services as a single comprehensive solution for secure VM migration to an authenticated and authorized environment. The proposed protocol initially performs the local authentication and authorization of migration client. The authorization servers on both of the source and destination domains mutually authenticate the domains (using FIPS-196) through exchange of digitally signed tickets. A symmetric session key is generated on both ends using ECDH and VM data is encrypted during transmission using AES. For data integrity SHA-256 is used. Moreover, least possible inter domain message exchange for mutual authentication of domains make the protocol not only secure but efficient as well.

References

1. The Xen Project. www.xenproject.org. Accessed 11 December 2013
2. VMware Virtualization for Desktop & Server, Application, Public & Hybrid Clouds. www.vmware.com. Accessed 11 December 2013
3. Kernel based Virtual Machine. www.linux-kvm.org. Accessed 11 December 2013
4. Oberheide, J., Cooke, E., Jahanian, F.: Empirical exploitation of live virtual machine migration. In: Proceedings of BlackHat DC Convention (2008)
5. Zhang, F., Huang, Y., Wang, H.: PALM: security preserving VM live migration for systems with VMM-enforced protection. In: The 3rd Asia-Pacific Trusted Infrastructure Technologies Conference, pp. 9–18 (2008)
6. Travostino, F., et al.: Seamless live migration of virtual machines over the MAN/WAN. Future Gener. Comput. Syst. **22**(8), 901–907 (2006)
7. Devi, Y., Aruna, P., Sudha, D.: Security in virtual machine live migration for KVM. In: International Conference on Process Automation, Control and Computing (PACC), pp. 1–6. IEEE (2011)
8. Wang, W., Zhang, Y., Lin, B., Wu, X., Miao, K.: Secured and reliable VM migration in personal cloud. In: The 2nd International Conference on Computer Engineering and Technology (ICCET), vol. 1, pp. 705–709. IEEE (2010)
9. NIST Guide to Security for full Virtualization, Special Publication 800–125 (2011)
10. eXtensible Access Control Markup Language (XACML) Version 3.0, Candidate OASIS Standard 01 (2012). http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cos01-en.html

11. Wood, T., Ramakrishnan, K.K., Shenoy, P., Merwe, J.V.: CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines. In: Proceedings of the 7th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE-11), NY, USA, pp. 121–132 (2011)
12. Price, M.: The paradox of security in virtual environments. IEEE Comput. **41**(11), 22–28 (2008). IEEE
13. Schwarzkopf, R., Schmidt, M., Strack, C., Martin, S., Freisleben, B.: Increasing virtual machine security in cloud environments. J. Cloud Comput.: Adv. Syst. Appl. vol. 1. Springer (2012)
14. Lombardi, F., DiPietro, R.: Secure virtualization for cloud computing. J. Network Comput. Appl. **34**(4), 1113–1122 (2010). Elsevier
15. Al-Kiswany, S., Subhraveti, D., Sarkar, P., Ripeanu, M.: VMFlock: virtual machine co-migration for the cloud. In: Proceedings of the 20th International Symposium on High Performance Distributed Computing, pp. 159–170. ACM (2011)
16. Aslam, M., Gehrmann, C., Bjorkman, M.: Security and trust preserving VM migrations in public clouds. In: IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, (TrustCom), pp. 869–876 (2012)
17. Danev, B., et al.: Enabling secure VM-vTPM migration in private clouds. In: Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC), pp. 187–196. ACM (2011)
18. Xianqin, C., et al.: Seamless virtual machine live migration on network security enhanced hypervisor. In: IEEE 2nd International Conference on Broadband Network & Multimedia Technology, (IC-BNMT), pp. 847–853. IEEE (2009)
19. Recommendation for Pair Wise Key Establishment Schemes using Discrete Logarithm Cryptography (Revised), NIST Special Publication 800–56A (2007)
20. Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197 (2001)
21. Secure Hash Standard (SHS), Federal Information Processing Standards Publication 180-4 (2012)
22. Kaufman, C., Hoffman, P., Nir, Y., Eronen, P.: Internet Key Exchange Protocol Version 2 (IKEv2), IETF RFC-5996 (2010)
23. Dierks, T., Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2, IETF RFC-5246 (2008)