International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-11, September 2019

An Automated VM Security Framework for Live Migration

A. Jyothi, B.Indira

Abstract: ***With the unbounded growth in the infrastructures for application hosting, demand from the consumers of the applications and the trade-off between the application availability with cost for application hosting is pushing the application providers towards cloud.The support from the cloud computing towards the application development are the dynamic load balancing, saving cost for energy and in premises hardware dependency. The dynamic load balancing is made possible by the concept of migrations of virtual machines or VMs. The migration includes identification of high loads on specific hosts, identification of possible virtual machines to be migrated and possible target hosts, where the VMs can be migrated. The challenge of migrating the virtual machines from the source to the destination physical hosts is during the migration process the virtual machines are exposed to the network and the other users available or have access to the same communication channels. Also, the virtual machines data to be made live in the target physical system, popularly called the VM images, are treated as regular files before those are live. Hence in the migration cycle, starting from the transfer of the virtual machine image and till the transferred virtual machine is live, there is a gap of security and which needs to be filled. The virtual machine images often contain the application, data generated by the application and data to be consumed by the application. Regardless to mention these three components are critical and must be prevented from the unauthorised access. Hence, a number of research attempts have proposed various schemes to secure the VM images by employing various encryption mechanisms. There methods are criticised for consuming high amount of computing capabilities for encrypting – decrypting VM images and resulting into violation of service level agreements by making the application not available for higher time.Thus, this work proposes a novel method for encrypting a higher volume VM image in less time by deploying a progressive and adaptive encryption method. The work also establishes the thought of the improvement by testing the algorithm in the light of SLA violation reduction compared with existing methods.***

Keywords: Adaptive Encryption, Channel Security, VM File progressive Encryption, VM File Security, VM File Segmentation

I. INTRODUCTION

The use of cloud computing intrigued the reduction in cost and increase in efficiency of application hosting industry.The basic of these advancements in the field of infrastructure management is the virtualization. Virtualization allows multiple operating systems to run in parallel on a single physical system, thus enables all the virtual machines to work under isolation. The work by Barham et al. [1] elaborates on the techniques and advantages of virtualization. With the visible advantage of higher utilization of the physical resources available, virtualization gains a high popularity among the owners and managers of data centres for hosting applications. Also, the consumers and the application owners are also inclined towards this benefit. The statistics provided by the Clark et al. [2] is a significant evidence of this fact. The capabilities of virtual machines are not limited in segregating the physical resources from one physical system, rather using virtualization, a number of physical resources can be collated together to serve a higher demand. This significantly reduces the cost of maintaining individual servers separately for single application. The reduction of cost can also be visualized in terms of physical space required for hardware peripherals, power consumptions, cooling and as mentioned by various researchers, in terms of man power as well. The work by Padala et al. [3] and the work by Murugesan et al. [4] particularizes on the strategic benefits from virtualization.

The advantage of bringing cloud computing into the mainstream of application hosting is the availability of load balancing and migration of the virtual machines from one physical host to another with higher or lower capabilities. The load balancing is made possible by deploying the concept of VM migration. The migration in practical can be static or live. The static migration is the mechanism where the actual virtual machine is shut down, then migrated to another physical host and then start again to host the application or services. During a static migration, the service is expected to be interrupted. In the other hand, during the live migration, the initial image of the virtual machine is still running during the migration process. Once the migration is completed and the next virtual machine is up and running, then the first image of the virtual machines can be shut down. This method can compromise on the cost of power consumption as both the virtual machines on both of the physical servers will be running during the complete migration process. Regardless to mention that the virtual machine images or the virtual machine data will be transferred from the source physical system to the destination physical system over the network, thus can be visible to all the users having access to the same communication channel. This problem has to be addressed in order to make cloud computing a better and safer place.

The work of Djenna et al. [5] elaborates on this issue and attracts the attention of the researchers. In further, the notable survey by Ristenpart et al. [6] and W. Fan et al. [7] points to the fact that during migration, the virtual machine data is migrated as plain text and can easily be sniffed.

Thus, this work aims to solve this issue and in parallel address the other relevant issues to this fact as:

- During migration of the virtual machine data or the VM images, the encryption is a must to incorporate.

Manuscript published on 30 September 2019.
*****Correspondence Author(s)
A.Jyothi, Asst.Prof. In Anurag Group of Institutions, Hyderabad
Dr. Baddam Indira, Assistant Professor in Chaitanya Bharathi Institute of Technology, Hyderabad,India.

- During the process of encryption and the decryption of the virtual machine data in the source and destination end can make the migration process more time complex. Thus, reducing the time for applying security measures on large files like VM images must be reduced.
- Also, the cost for live migration cannot be ignored and a firm strategy has to be made in terms of reducing this cost without causing high SLA violation.

This work proposes to solve the above-mentioned issues and the rest of the work is furnished as, in the Section – I, the parallel research outcomes are measured, the virtual machine formats are important to realize in order to apply better security measures and discussed in Section – IV, the VM security challenges associated with the specific VM formats are elaborated in the Section – V, the proposed framework and the algorithm is demonstrated in the Section – V,the obtained results are discussed in Section – VII, the comparative analysis in order to establish the fact of improvements is enlisted in Section – VIII and the final conclusion is presented in Section – IX

II. OUTCOMES FROM THE PARALLEL RESEARCH

The live migration is a well-accepted strategy for majority of the virtualization software and tools. Nevertheless, the migration process exposes the virtual machine data for all the software variation from Xen to KVM to VMotion and others. The notable work by Oberheide J. et al. [8] demonstrated the security risks during migration for three layers of any network of data centre protocols. The risks get higher and higher from control layer to data layer to migration module layer. This vulnerability is criticized for all major virtual machine management software stacks. Yamunadevi L. et al. [9] tried simulating the attacks on VM data on KVM software stack and demonstrated the simplicity of the attacks programs needed to take control of the data. Also, W. Q. Huang et al. [10] proposed the similar simple attack models on XEN and VMWare and demonstrate the parallel results.

Many parallel research attempts have also demonstrated that the common communication channel can be highly vulnerable. The work by Dawoud W.et al. [11] is a significant proof that the attackers can get easy access to sensitive information of the users. M. R. Anala et al. [12] had stepped up the experiment and demonstrated that the virtual machine memory can be modified to gain the permanent access to the virtual machine after migration [Fig – 1].

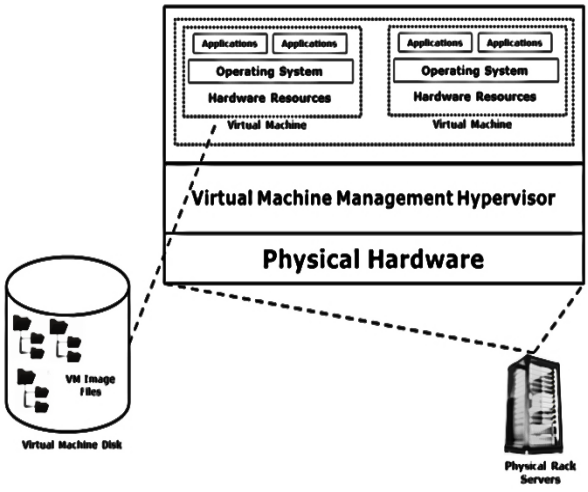


Fig. 1 Hardware Components for Virtualization

Henceforth, with the understanding of virtual machine migration problems and simulations proposed by various parallel research attempts, this work makes an attempt to understand the virtual machine data formats to gain higher understandings and possibilities of securing the VM data in the next section.

III. VM FILE FORMATS

Understanding of the virtual machine data formats is essential for making the right decision and design for the algorithms of encryption and decryption. Thus, in this section of the work, the vm data formats are analysed.

The virtual machine data files are similar to the hard drive formats and the partitioning of the VM files are also similar to the hard drive partitions. The popularity of the VM data files are increasing due to the benefits as easy to migrate, backup mechanisms, snapshot managements for quick deployments and application of software patches or stacks [Fig – 2].

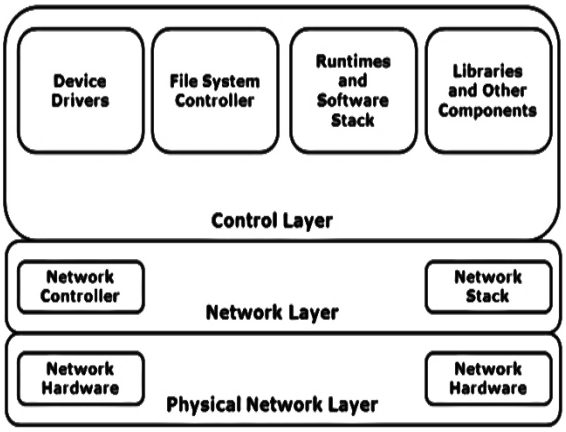


Fig. 2 Generic Architecture for VM Data Formats

A. Fixed Length Files

The fixed length files for the virtual machines are the files where the complete file allocated in continuous blocks of the memory. The allocation of the file is done during the virtual machine initialization for the first time. Hence regardless to the utilization, the size of the file is unchanged.

B. Dynamic Length Files

The dynamic files are also allocated during the configuration of the virtual machine. Nevertheless, the size of the file is not static, rather dynamic. Hence, the allocation blocks are not continuous and can be segmented. The new block on the memory is created based on the usage of the virtual machine and grows or sinks based on the data generated by the application.

The issue with Dynamic Length Image File is if the little sum changes should be connected to the parallel imitated frameworks, at that point the entire file should be supplanted.

C. Dependent Length Files

Because of the issues of delta change administration in past file designs, the Dependent VM file deals with the entire change administration with a property called Undo Change.

This property interfaces the altered file with the past record as youngster record just with the change data.

Henceforth this VM file arrange is significantly acknowledged because of its transportability for developments over numerous host servers.

D. Linked Length Files

The Linked VM data designs are significantly like the other file groups, with the exception of the property of availability between the physical hard drive and the intelligent hard drives.

This component empowers the designers and analysts to nearly examinations and consolidates the server-based record frameworks into the application and builds the efficiencies for a few instances of use improvement like copied stockpiling or physically perusing the system parameters through the applications.

IV. VM SECURITY CHALLENGES

The security challenges are not limited to the migration phase of virtual machine life cycle alone. Rather, in every phase of the virtual machine management. In this section of the work, the types of security threads and possible solutions are discussed.

A. Access Control

Firstly, the inappropriate access controls to any virtual machine is the data centre can potentially allow any unauthorized user to have complete control over the virtual machine and allow those users to create, migrate or terminate the virtual machine available. The work of M R. Anala et al. [13] have demonstrated the devastating effects on the data centre due to unavailability of proper access controls.

B. Authentication to the Virtual Machine

Secondly, during the live migration of the virtual machines over the network demands a high mutual trust between the physical source, network controllers and the destination physical system. Any violation of the trust factors can lead to the exposure of the data in the network. Once again, the work by M R. Anala et al. [13] have significantly shown the losses caused by this kind of situations.

C. Non – Monitoring

Thirdly, due to non – monitoring of the virtual machine life cycles can lead to the permanent damage to the virtual machines available in the data centre. Also, it is regardless to mention that the monitoring of the virtual machines must be secured in order to prevent the unauthorized access by the auditors. D. Perez-Botero et al. [14] have demonstrated the possibilities of loss of data in the virtual machines due to unidentified access by the auditors

D. Confidentiality of Data

Fourthly, as repeatedly discussed in this work, during the migration process of the virtual machines, the data can be highly insecure without providing any significantly efficient encryptions. The virtual machines during migration can be available as clear text to the network users and can easily be tampered. In the work by S. Biedermann et al. [15] have listed the possible damages in terms of memory access for the virtual machines during migration.

E. Secure Communication

Fifth, elaborating the fourth point, the attackers in the network can completely gain access to the communication channel and tamper all data transmissions. These kinds of attacks are called channel poisoning. The attackers often use the ARP or DHCP or DNS protocols to deploy these attacks. The work by J. Oberheide et al. [16] listed out the potential indemnities to the data centre due to insecure communication channels.

F. Integrity of Data

Further extending the previous points, the data can be permanently altered during the migration process and this can affect the data to be consumed or produced by the application running on the virtual machines. Habitually, it has been observed that the data generated by the affected virtual machine applications causes serious damages to the outcomes of the applications.The recommendations from the Trusted Computing Group [17] shows the vulnerabilities, which are to be avoided.

G. Availability of Virtual Machine

It is also being observed that the unauthorized access to the data centre virtual machine stacks can initiate high volume of live migrations to the targeted destination systems. This may cause high and unmanaged loads to the destination systems making the application late responder to the requests and resulting into unavailable applications or virtual machines. The work of A. Back et al. [18] is a clear indication of the increase of network traffic resulting into network congestions.

V. PROPOSED FRAMEWORK

Upon understanding the threats and limitations in the parallel research outcomes, in this section of the work, the proposed framework to protect the VM data during migration is formulated.

The novelty of this framework is to reduce the size of the virtual machine by applying progressive segmentation on large VM files and then applying the security policies. This reduces the time complexity for each encryption and decryption methods applied. The phases of the framework are elaborated here:

Firstly, the smart segmentation of the VM data files are carried out. This phase ensures the reduction of time and considers the dependency of the VM data in order to avoid the time losses for starting up the VM in the destination physical server.

Algorithm - I: VM Data Smart Segmentation (VDSS)

```
Step - 1. Identify the VM data ready for migration
Step - 2. For each VM data files
        a. Identify the server transmission ratio
        b. Separate the VM data into equal number of blocks
           based on transmission ratio
        c. For each block [VM(i)]
                i. If VM(i) not dependent VM(i+1)
                        1. Then identify VM(i) as
                           independent block
                ii. Else
                        1. Merge VM(i) and VM(i+1)
                        2. Consider merged VM as
                           independent block
                iii. Send for encryption
Step - 3. Finalize the data blocks
```

Secondly, after the segmentation of the VM data blocks, the generation of the keys plays a major important role. This work applies a progressive and adaptive key generation mechanism to encrypt the VM data blocks.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Algorithm - II: Progressive and Adaptive Key Generation (PAKG)
Step - 1. Populate random 32-bit initial key Sets Step - 2. For each key in the collection (n) a. Generate parent key as n(i) * length of n(i) * total number of keys * probability of selecting n(i) from n b. Use each patent key to encrypt each block of VM data Step - 3. Repeat the Step - 2 for generating verification code until n = 1 a. Denote as master key Step - 4. Finalize the keys

Further, the major component of this framework is the encryption of the data blocks.

Algorithm - III: VM Data Encryption (VMDE)
Step - 1. Accept the independent VM blocks Step - 2. For each block a. Generate the final block as Header and Data block b. Header = master key + Dependency Map c. Data Block = Independent VM block d. Apply RSA algorithm to encrypt the Header Block e. Apply parent keys to encrypt the Data Block Step - 3. Send for transmission

Once the VM data files are encrypted, the transmission of the data blocks can commence

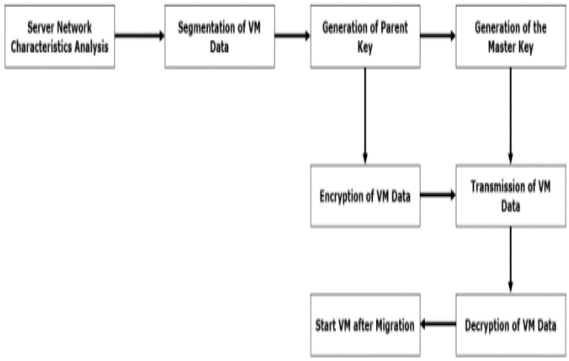
Algorithm - IV: VM Block Transmission Protocol (VBTP)
Step - 1. Transmit the Random number generation function signature Step - 2. Transmit the Secure RSA keys Step - 3. Transmit the VM Blocks

After the transmission is over, it is the responsibility of the physical host or the recipient of the VM blocks to decrypt the received file and perform de-segmentation and finally start-up the VM.

Algorithm - V: VM Data Decryption (VMDD)
Step - 1. Accept VM blocks Step - 2. For each VM blocks a. Decrypt the Header Block b. Extract the dependency map and master key c. Generate the parent keys using Random number generation function d. If master key not equal Generated master key i. Request for further transmission e. Else i. Proceed for decryption f. Apply each sequence of parent keys on VM blocks g. Apply dependency map to collate VM Blocks h. Deploy VM blocks Step - 3. Start the VM

Hence, the VM data is not transmitted securely over the network without increasing significant amount of time and making any data vulnerable to the attackers.

The complete framework is also visualized graphically here [Fig – 3].



Henceforth, in the next section of the work, the obtained results from each phase of the framework are analysed and discussed

II. RESULTS AND DISCUSSION

In this section of the work, the results obtained from each phase of the framework are furnished. The results are highly satisfactory and improved compared to the existing methodologies, which is again elaborated in the next section

A.Experimental Setup

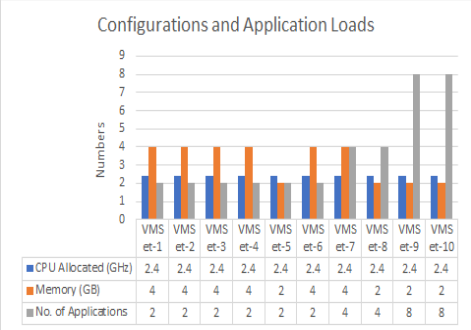
During the experimental phase of this work, a total of 10 virtual machines are configured to test the proposed framework. Firstly, the experimental setup is analysed [Table – 1].

Retrieval Number: K19200981119/19©BEIESP
DOI: 10.35940/ijitee.K1920.0981119
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

VM Name	Operating System	CPU Allocated (GHz)	Memory (GB)	No. of Applications	VM File Size (MB)	Type
VMSet-1	Windows 2000 Server	2.4	4	2	20	Linked
VMSet-2	CentOS	2.4	4	2	11	Fixed
VMSet-3	osX	2.4	4	2	21	Linked
VMSet-4	Ubuntu	2.4	4	2	24	Fixed
VMSet-5	RedHat	2.4	2	2	12	Linked
VMSet-6	Windows XP	2.4	4	2	12	Fixed
VMSet-7	Ubuntu	2.4	4	4	12	Linked
VMSet-8	CentOS	2.4	2	4	27	Fixed
VMSet-9	Ubuntu	2.4	2	8	29	Linked
VMSet-10	CentOS	2.4	2	8	20	Fixed

The configurations and the applications loads are visualized graphically here [Fig – 4].



Further the VM Sizes are analysed here [Fig – 5].

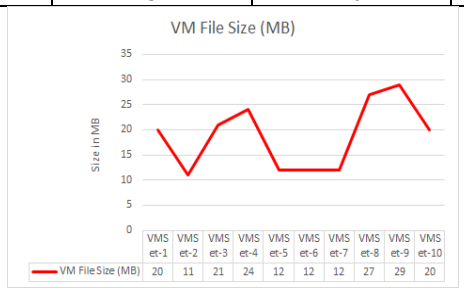


Fig. 3 VM Data File Sizes

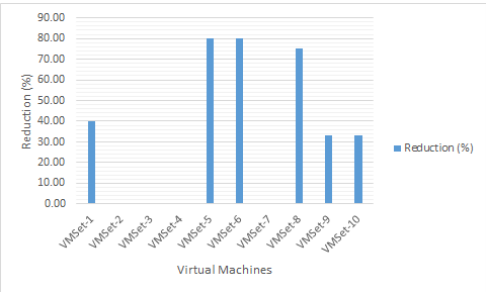
B. VM Data Segmentation Phase

Secondly, the VM data files are segmented in order to reduce the amount of time required for encryption and decryption [Table – 2].

TABLE I: SEGMENTATION REPORT

VM Name	Operating System	CPU Allocated (GHz)	Memory (GB)	No. of Applications	VM File Size (MB)
VMSet-1	5	2	3	2	40.00
VMSet-2	2	0	2	0	0.00
VMSet-3	1	0	1	0	0.00
VMSet-4	1	0	1	0	0.00
VMSet-5	5	4	1	4	80.00
VMSet-6	5	4	1	4	80.00
VMSet-7	1	0	1	0	0.00
VMSet-8	4	3	1	3	75.00
VMSet-9	3	1	2	1	33.33
VMSet-10	3	1	2	1	33.33

The size reduction for the tested virtual machine files are visualized graphically here [Fig – 6].



C.VM Data Segmentation Time

Thirdly, the VM data files segmentation times are analysed [Table – 3].

TABLE II: SEGMENTATION TIME

VM Name	Segmentation Time (msec) (A)
VMSet-1	10
VMSet-2	30
VMSet-3	24
VMSet-4	18
VMSet-5	39
VMSet-6	18
VMSet-7	42
VMSet-8	15
VMSet-9	14
VMSet-10	18

The results are visualized graphically here [Fig – 7].

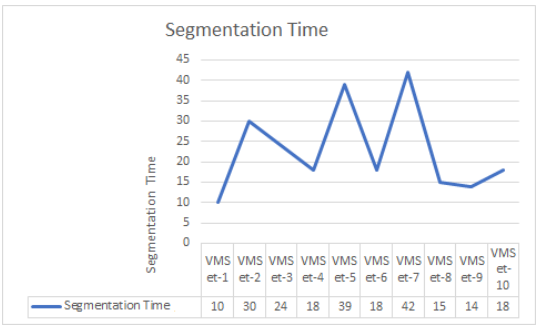


Fig. 4 VM Data File Segmentation Time

D.VM Data Encryption Time

Fourthly, the VM data files encryption times are analysed [Table – 4].

TABLE III: SEGMENTATION TIME

VM Name	Encryption Time (msec) (B)
VMSet-1	1074
VMSet-2	144
VMSet-3	73
VMSet-4	349
VMSet-5	290
VMSet-6	997
VMSet-7	632
VMSet-8	144
VMSet-9	73
VMSet-10	997

The results are visualized graphically here [Fig – 8].

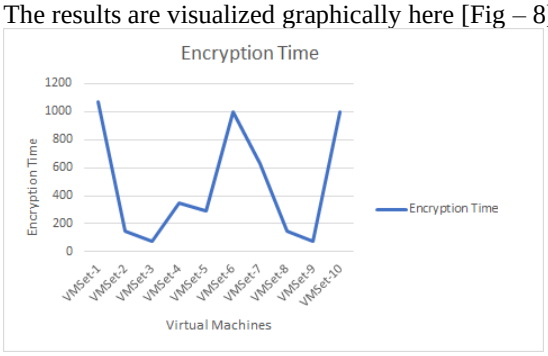


Fig. 5 VM Data File Encryption Time

E.VM Data Transmission Time

Further, the VM data files transmission times are analysed [Table – 5].

TABLE IV: Transmission Time

VM Name	Transmission Time (msec) (C)
VMSet-1	1074
VMSet-2	144
VMSet-3	73
VMSet-4	349
VMSet-5	290
VMSet-6	997
VMSet-7	632
VMSet-8	144
VMSet-9	73
VMSet-10	997

The results are visualized graphically here [Fig – 9].

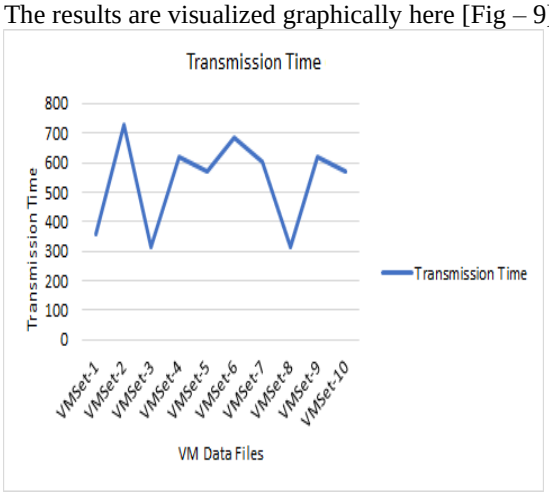


Fig. 6 VM Data File Transmission Time

F.VM Data Decryption Time

Further, the VM data files Decryption times are analysed [Table – 6].

TABLE V: DECRYPTION TIME

VM Name	Decryption Time (msec) (D)
VMSet-1	114
VMSet-2	157
VMSet-3	265
VMSet-4	194
VMSet-5	395
VMSet-6	596
VMSet-7	666
VMSet-8	265
VMSet-9	194
VMSet-10	596

The results are visualized graphically here [Fig – 10].

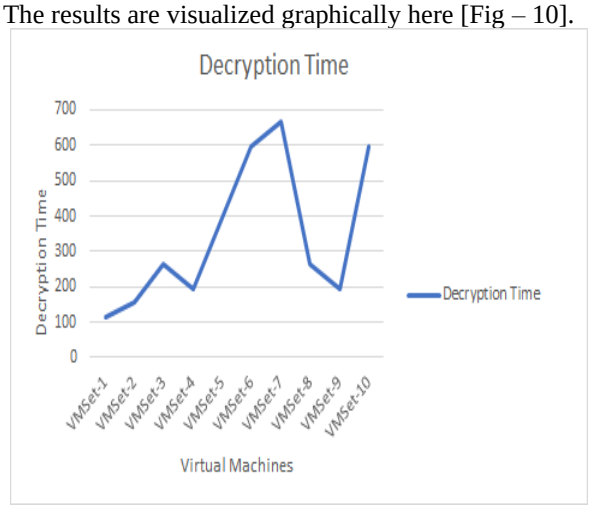


Fig. 7 VM Data File Decryption Time

G.VM Data De-Segmentation Time

Furthermore, the VM data files de-segmentation times are analysed [Table – 7].

Retrieval Number: K19200981119/19©BEIESP
DOI: 10.35940/ijitee.K1920.0981119
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

TABLE VI: DE-SEGMENTATION TIME

VM Name	De-Segmentation Time (msec) (E)
VMSet-1	10
VMSet-2	30
VMSet-3	24
VMSet-4	18
VMSet-5	39
VMSet-6	18
VMSet-7	42
VMSet-8	15
VMSet-9	14
VMSet-10	18

The results are visualized graphically here [Fig – 11].

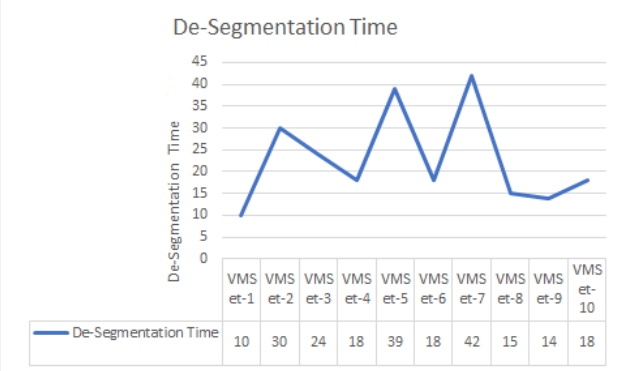


Fig. 8 VM Data File De-Segmentation Time

H. Total VM Migration Time

Finally, regardless to mention and natural to understand that the segmentation, encryption, transmission and decryption times will be added to the VM migration time and the total time for the framework to complete must be analysed [Table – 8].

III. COMPARATIVE ANALYSIS

Any additional phase in the migration process will result into additional time for the VM migration process. The additional time will certainly reduce the SLA. Thus, it is important to realize the amount of SLA violations due to this added security protocols [Table – 9].

TABLE VII: TOTAL VM MIGRATION TIME

VM Name	Segmentation Time (msec) (A)	Encryption Time (msec) (B)	Transmission Time (msec) (C)	Decryption Time (msec) (D)	De-Segmentation Time (msec) (E)	Total Migration Time (msec) (A+B+C+D+E)
VMSet-1	10	1074	357	114	10	1565
VMSet-2	30	144	729	157	30	1090
VMSet-3	24	73	313	265	24	699
VMSet-4	18	349	618	194	18	1197
VMSet-5	39	290	572	395	39	1335
VMSet-6	18	997	682	596	18	2311
VMSet-7	42	632	604	666	42	1986
VMSet-8	15	144	313	265	15	752
VMSet-9	14	73	618	194	14	913
VMSet-10	18	997	572	596	18	2201

The average time during testing load for the framework is 1404.9 milli sec or 0.02 mins.
Thus, it is natural to understand that these additional times for the framework must be added to the live load balancing times in order to realize total SLA violation times
Henceforth, in the next section of work, this framework is combined with the popular VM load balancing mechanisms

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

TABLE VIII: INCREASE IN TIME DUE TO SECURITY PROTOCOLS

Load Balancing Policies	VM Selection Time (Min)	Proposed VM Security Framework Time (Mins)	Total Time (Mins)	Difference (%)
VM Selection=IQR, VM Migration=MC	48.24	0.02	48.26	0.05
VM Selection=IQR, VM Migration=MMT	7.92	0.02	7.94	0.30
VM Selection=LR, VM Migration=MC	46.80	0.02	46.82	0.05
VM Selection=LR, VM Migration=MMT	15.84	0.02	15.86	0.15
VM Selection=LR, VM Migration=MU	6.12	0.02	6.14	0.38
VM Selection=LR, VM Migration=RS	37.44	0.02	37.46	0.06
VM Selection=LRR, VM Migration=MC	7.92	0.02	7.94	0.30
VM Selection=LRR, VM Migration=MMT	19.44	0.02	19.46	0.12
VM Selection=LRR, VM Migration=MU	3.96	0.02	3.98	0.59
VM Selection=LRR, VM Migration=RS	5.76	0.02	5.78	0.41
VM Selection=MAD, VM Migration=MC	79.20	0.02	79.22	0.03
VM Selection=MAD, VM Migration=MMT	7.92	0.02	7.94	0.30
VM Selection=MAD, VM Migration=MU	9.72	0.02	9.74	0.24
VM Selection=MAD, VM Migration=RS	25.56	0.02	25.58	0.09
VM Selection=THR, VM Migration=MC	80.28	0.02	80.30	0.03
VM Selection=THR, VM Migration=MMT	6.12	0.02	6.14	0.38
VM Selection=THR, VM Migration=MU	1.80	0.02	1.82	1.30
VM Selection=THR, VM Migration=RS	3.96	0.02	3.98	0.59

Thus, it is natural to understand that the average violation of the SLA can be 0.30% for any load balancing policies.

Henceforth, with the complete understanding of the improvements over existing VM migration policies in terms of security, in the next section, this work presents the conclusion.

IV. CONCLUSION

The growth in cloud computing motivated the application development and hosting practitioners to use virtualization and virtual machine migration for load balancing. The migration of the virtual machine is a crucial task as the application and data will be migrated over the network and the traditional methods recommend that the VM data to be migrated as plain text. Thus, making the data vulnerable to tamper. The effects of tampering the VM data is elaborated in this work and in the works by other research attempts. The primary challenge of securing the VM data is to increase the load balancing time, resulting into deviation in Service Level Agreements or SLA. Thus, this work proposes an adaptive segmentation and encryption method to secure the VM data with least time and least violation of the SLA. The framework results into a remarkable average of 0.30% increase in the time, which is nearly negligible in any SLA. The contribution from the work will significantly improve the criticism of cloud computing load balancing by providing a higher security and making the cloud computing domain a protected practice.

REFERENCES

1. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (SOSP19), pp. 164–177. ACM Press (2003)
2. Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: Proceedings of NSDI, pp. 273–286. USENIX Association, Berkely (2005)
3. 3 Padala, P., Zhu, X., Wang, Z., et al.: Performance evaluation of virtualization technologies for server consolidation. Virtualiz. VMware ESX Serv. 9, 161–196 (2007)
4. Murugesan, S.: Harnessing green IT: principles and practices. In: Proceeding of IT Professional, vol. 10, pp. 24–33. IEEE Computer Society (2008)
5. Djenna, A., Batouche, M.: Security problems in cloud infrastructure. In: The 2014 International Symposium on Networks, Computers and Communications, pp. 1–6. IEEE (2014)
6. Ristenpart, T., Tromer, E., Shacham, H., et al.: Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In: CCS Conference, pp. 199–212 (2009)
7. Fan, W., Kong, B., Zhang, Z.J., Wang, T.T., Zhang, J., Huang, W.Q.: Security protection model on live migration for KVM virtualization. J. Softw. 27(6), 1402–1416 (2016). (in Chinese)
8. Oberheide, J., Cooke, E., Jahanian, F.: Empirical exploitation of live migration of virtual machines. In: Black Hat DC Briefings, Westin Washington DC City Center (2008)
9. Yamunadevi, L., Aruna, P., Sudha, D.D., et al.: Security in virtual machine live migration for KVM. In: 2011 International Conference on Process Automation, Control and Computing (PACC), pp. 1–6. IEEE (2011)
10. 10 Fan, W., Huang, W.Q., Jiang, F., Liu, C., Lv, B., Wang, R.R.: Research on security of memory leakage in live migration based virtualization. In: Twenty-Fourth National Conference on Information Security (IS 2014), vol. 09, pp. 12–17 (2014)
11. 11 Dawoud, W., Takouna, I., Meinel, C.: Infrastructure as a service security: challenges and solutions. In: The 7th International Conference on Informatics and Systems (INFOS), pp. 1–8 (2010)
12. 12 Anala, M.R., Shetty, J., Shobha, G.: A framework for secure live migration of virtual machines. In: 2013 International Conference on IEEE Advances in Computing, Communications and Informatics (ICACCI), pp. 243–248 (2013)
13. 13 M R. Anala, J. Shetty, G. Shobha. A Framework for Secure Live Migration of Virtual Machines. International Conference on Advances in Computing, Communications and Informatics. 2013.

14. 14 D. Perez-Botero. A Brief Tutorial on Live Virtual Machine Migration From a Security Perspective.
15. 15 S. Biedermann, M. Zittel and S. Katzenbeisser. Improving Security of Virtual Machines during Live Migrations. Eleventh Annual Conference on Privacy, Security and Trust (PST). 2013.
16. 16 J. Oberheide, E. Cooke, F. Jahanian.Empirical Exploitation of live migration of virtual machines. Proc of Black Hat DC, March 24, 2008.
17. 17 TrustedComputingGroup. http://www.trustedcomputinggroup.org/resources/tpmmainspecification
18. 18 A. Back, U. Mller, and A. Stiglic. Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems. Information Hiding, volume 2137 of Lecture Notes in Computer Science, page 245-257. Springer, 2001.

AUTHORS PROFILE

A.Jyothi received her MTech from JNTUH, Hyderabad, Telangana, India in 2012 and pursuing research in Osmania University. She is working as Asst.Prof. In Anurag Group of Institutions, Hyderabad .She has 12 years of teaching experience .Her research interest includes security in cloud computing and IOT

Dr. Baddam Indira received her MCA from Kakatiya University, Telangana,India in 1996 and a Ph.D. from Sri. Padmavati Mahila Visvavidyalayam, Tirupati, India in 2008. She is currently working as an Assistant Professor in Chaitanya Bharathi Institute of Technology, Hyderabad,India. She also has 23 years of teaching experience. Her research interests include Digital Image
Processing, Neural Networks, Data Structures and Cloud Computing.