I.J. Wireless and Microwave Technologies, **2018, 1, 50-59**
Published Online January 2018 in MECS(http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2018.01.06

Available online at ***http://www.mecs-press.net/ijwmt***

Previously-Selected-Server-First based Scalable VM Placement Algorithm for Mitigating Side Channel Attacks in Cloud Computing

Adi Maheswara Reddy G[a], K Venkata Rao[b], JVR Murthy[c]

[a]*Research Scholar, Department of CSE, JNTUK, Kakinada, AP, INDIA*
[b]*Department of CSE, Vignan Institute of Information Technology Visakhapatnam, AP, INDIA*
[c]*Department of CSE, JNTUK College of Engineering, Kakinada, AP, INDIA*

Received: 08 September 2017; Accepted: 11 October 2017; Published: 08 January 2018

Abstract

Pertaining to the rapid usage of cloud computing, cloud based approaches are growing as an fascinating domain for numerous malignant tasks. Security is one of the vital issues faced by the cloud computing environment while sharing resources over the internet. Consumers are facing distinct security hazards while using cloud computing platform. Previous works mainly attempted to mitigate the side channels attacks by altering the infrastructure and the internal procedures of the cloud stack. However, the deployments of these alterations are not so easy and could not resist the attacks. In this paper, the authors attempted to solve the issues by enhancing the VM Placement policies in such a way that, it is complex for the invaders to collocate their object. A secure Dynamic VM placement approach is presented for the VM allocations into different servers in the cloud. The performance comparison of the suggested methodology is shows that the proposed approach has better efficiency evaluations such as hit rate, loss rate and resource loss when compared to other V M placement policies.

Index Terms: Cloud Security, Co-resistance Attacks, VM placement policy, PSSF, Greedy Algorithm, VM migration

1. Introduction

There is an expanding complication and rapid enlargement in the network appliance amenities in the current era. Thus, it is essential to incorporate and consolidate the IT infrastructure for flexible centralized control and

* Corresponding author. Tel:
E-mail address:

administration such that the complete ownership price could be diminished consistently. Under this situation, the cloud computing conception is introduced [16].Cloud computing is a developing scientific archetype that contributes to a flexible, expandable and good infrastructure and amenities for organizations. Cloud information are accumulated and accessed in an isolated server along with the facilities given by cloud service holders. Certain instances of cloud service holders are Amazon, Google's Application, IBM, etc., that has enlarged as a standard computing and storage service prototype for individual, business, and government projects apart from autonomous computing [14, 15], grid computing [13] and service computing [12].

Cloud computing security defines both physical and logical security problems over entire diverse service prototypes of software, platform and infrastructure. While sharing the resources over the internet, it also means customers are disclosed to added liability obtained through other inhabitants with whom the resources are being shared. Such sharing enables malicious tenants to have chance to attack another inhabitants on the similar physical node. Side-channel attack is absolutely this type of approach that the invaders beneath the cloud atmosphere that could attain the accessible patterns of CPU, memory and network of other inhabitants on the identical hardware by means of side-channel attack, such that they could achieve the private, financial and economical secrets of other inhabitants.

Thus numerous techniques are priory introduced to defend against the side channels attacks in Cloud Computing. Amongst which the VM placement algorithm or VM allocation policy is the utmost important and straight forward control which is employed to effect the possibility of co-location. Thus, directed to construct a secure protocol that could considerably maximize the complexity for invaders to obtain co-residence and mitigate side channel attacks. Accordingly, in this paper, a new dynamic VM Placement algorithm is suggested that attempts to the address the issue of scalability in already existing approaches.

The paper suggested the dynamic approach to allocate the VM in the server domains such that the influence of co-resistance amongst the machines are minimized and hence mitigates the side channel attacks by means previously-selected-server-first policy (PSSF) and baseline greedy algorithm. The issue of scalability in the greedy approach is addressed with the PSSF policy. Here, instead of considering the n-ways swaps of the VM of every user, the proposed approach minimized this search domain by selecting only the essential moves that resists the co-resistance attacks using PSSF policy. Thereby, a dynamic VM placement algorithm is proposed to mitigate the side channel attacks.

1.1 Organization of the paper

An introduction to the cloud computing security and issues such as side channel attacks and its related techniques along with the motivation for the proposed approach is given in this section. Previous literature works on the mitigation techniques of side channel attacks and enhanced VM placement policies are briefly discussed in section 2. A detailed explanation of the proposed PSSF based Scalable VM Placement Algorithm is given in section 3. The experimental results for the proposed approach is briefly given in section 4 followed by conclusions and references given in section 5 and section 6 respectively.

2. Related Works

A lot of recent work has proposed isolation techniques to reduce unpredictability by eliminating resource interference. Below we discuss related work with respect to cloud vulnerabilities, such as VM placement detection and side-channel attacks.

In [1], the usage of virtualization to distinguish an evaluation from malevolent customers those co-locate with its expanding ubiquitous. Demonstrating the side-channel attacks with loyalty is adequate to exhilarate a cryptographic key from a victim VM which could be mounted. It includes preventing the target VM with adequate frequency to facilitate fine-grained observing of its I-cache movement, filtering out enormous sources of noise in the I-cache rising from influence of hardware and software and core immigration that renders

numerous invaders interpretations inappropriate to the job of mining victim secrets.

A virtual cloud resource allotted model VCRAMU (Utility-based Virtual Cloud Resource Allocation Model) is suggested in [3]. In this approach, the issue of allocation of virtual cloud resources is preoccupied as a utility enlargement issue considering the trade-offs amongst the efficacy of the data centre and the efficiency of the applications into consideration, and exploiting the efficacy on the basis of meet user's presentation. An indigenous decision procedure and a universal decision procedure are likewise deliberated to resolve the issue. Additionally, this prototype could obtain an advanced service of the data centre when matched with other prototypes. Nevertheless, whenever the dimension of the cloud computing atmosphere becomes higher and higher, there would be certain issues with the prototype and approaches like performance holdups.

In [4], employing side channel attack, it could be very flexible to obtain the confidential data from a machine as it is appreciable notion to provide security in contrast to side channel attack in cloud computing employing the amalgamation of simulated firewall appliance and arbitrary encryption decryption since it accomplishes security against both front end and back end of cloud computing structure and likewise provide RAS (Reliability, Availability, and Security). It executes virtual firewall in cloud server thus whenever enemies recognize targeted VM in cloud infrastructure and formerly place an instantiate VM to targeted VM, simulated firewall avert this assignment phase inside channel attack due to the execution of virtual firewall in cloud server. Applies arbitrarily encryption decryption using concept of confusion and diffusion.

Cloud multi-tenancy has motivated a line of work on locating a target VM in a public cloud. Ristenpart et al. [5] showed that the IP machine naming conventions of cloud providers allowed adversarial users to narrow down where a victim VM resided in a large-scale cluster. Xu et al. [10] and Herzberg et al. [8] extended this study, resulting, in part, in cloud providers changing their naming conventions, reducing the effectiveness of network topology-based co-residency attacks. Following this evolution Varadarajan et al. [9] evaluated the susceptibility of three cloud providers to VM placement attacks, and showed that techniques like virtual private clouds (VPC) render some of them ineffective.

Xu et al. [7] studied the extent of co-residency threats in EC2 and the efficiency of their detection using network route traces. Bates et al. [11] proposed a system where adversarial VMs introduce traffic congestion in host NICs, which is then detected by remote clients. Similarly, Zhang et al. [6] designed HomeAlone, a system that detects VM placement by issuing side-channels in the L2 cache during periods of low traffic. Finally, Han et al. [2] proposed VM placement strategies that defend against placement attacks, although they are not specifically geared towards public clouds. With Bolt, we show that leveraging simple data mining techniques on the pressure applications introduce in shared resources increases the accuracy of VM co-residency detection significantly. Bolt does not rely on knowing the cloud's network topology or host IPs, making it resilient against recent techniques, such as VPCs.

3. Dynamic PSSF Based Scalable VM Placement Algorithm

In this section, a novel dynamic VM placements Algorithm is introduced that addressed one of the issues in NOMAD Virtual Machine placement Algorithm [17]. This Algorithm primarily addresses three main challenges such as to obtain an efficient algorithm, large search space due to more numbers of moves amongst the machines and Deployment of NOMAD system into cloud environment. The baseline greedy algorithm is employed as the VM placement policy in NOMAD [17] where the author mainly focuses on the scalability issues of large search space that is generated due to large number of moves in the baseline greedy algorithm i.e. numerous servers with numerous kinds of moves such as freely inserting a VM into the empty slots, pair wise swapping between VM, n-ways swapping and so on.

The primary goal of the proposed dynamic VM placement algorithm is to mitigate the co-resistant attacks in clouds environment and minimize the information leakage amongst the shared virtual machines of different users in the same servers. This issue is addressed by introducing the dynamic methodology for the VM Placement policy. The proposed methodology is implemented in two phases. In first phase the set of moves for

the VM of different users in different servers are obtained and in the subsequent phase these set of moves are employed for the VM baseline greedy placement algorithm.

3.1 Generation of set of moves using Previously-Selected-Server-First policy (PSSF)

From the existing VM placement policy, it is inferred that if the number of servers to which every individuals VMs that is allotted is restricted, formerly the victim VMs are lesser exposed to the invader, which limits the influence of co-resident attacks. PSSF policy also works on this idea where the highest priority is given to servers that previously host or once hosted VMs from the same user, whenever a novel VM appeal is being processed. The PSSF policy determines the set of moves that need to be initiated initially if any new VM of any user comes in. This policy mainly optimizes the search spaces by reducing the average number of individuals per server. In order to minimize the number of individuals per server, whenever an individual generates new VMs, they would initially be allocated to those servers that are previously host or once hosted VMs began by the same user. The Algorithm for PSSF policy is given as:

1. Initialize the list of PSSF and NPSSF as $PSSList = \{\}\ and\ NPSSList = \{\}$
2. For every server s_i in S
    If (s_i has adequate residual resources)
        If (s_i previously hosts or hosted u's VMs)
            If (s_i hosts fewer compared to N* of u's VMs)
                $PPSList.add(s_i)$
            Else
                $NPPSList.add(s_i)$
3. $if\,(!PSSList.isEmpty())$
    $return\ PSSList.get(random(PSSList.size()))$
   Else
    $sort(NPPList, groupIndex, ResourceLeft)$
4. I = the number of servers with the similar group index and residual resources as the initial server in $NPSSList(NPSSList.get(0))$
5. Mark $NPSSList.get(random(I))$ as earlier designated for individual and return it.

The list of term present in PSS List and NPSS List represents the set of moves or types of moves that is required to initiate the baseline Dynamic placement Algorithm. Along with the generations of different types of moves, this approach also resists the co-resistance attacks to certain extent. For instance: the victim individual initiates ten VMs, and they are equally allotted to two servers, s_1 and s_2. As a consequence, these two servers host higher VMs compared to other servers currently, and it is improbable for additional VM appeals to be allotted to them till entire other servers likewise host the similar amount of VMs. Nevertheless, it is complex for invaders to obtain co-residence, as the victim VMs are assigned in combined and as outcomes are lesser exposed.

3.2 Baseline Dynamic VM placements Algorithm

The Baseline Dynamic VM Placement Approach prerequisites to evaluate the VM placements for each epoch with the aim of reducing the data leakage amongst random pairs of cloud clients, whereas guaranteeing that the global price of performing so (i.e., amount of relocations) is low. Specifically, need to reduce the complete data leakage function tend to certain budget on the immigration overhead measured in terms of total amount of immigrations. The issue victim dimensions such higher public cloud deployment with tens of thousands of servers with coarsely 5-6 VM slots per server. In the baseline dynamic approach, the numbers of

moves are generated from the PSSF policy involving VM's. Instead of considering n-way swaps between the pair of VM, the obtained set of moves determines the size of the search space.

Each movement comprises of price acquired as number of immigrations essential to implement the change and the profit it earns in terms of the minimization in data leakage. Formerly, in every repetition of the approach, the finest movements are selected in between the migration budget that provides the extreme profit in terms of minimization in data leakage. Every movement theoretically fluctuations the position of the system and consequently the benefit of upcoming movements might minimize or maximize pertaining on the movements that are priori made such as moving VM might understand that whole priori deliberated group of movements comprising this instance might no longer offer any kind value. Therefore, the group of permitted movements are unambiguously re-estimated and the profit that is yield using PSSF policy is employed for next phase. The Algorithm for Baseline Dynamic Placement algorithm is given a

$function\ DYNAMIC_{ALG}(Cur_{place}, Budget, PSSF(Types\_of\_Moves))$
1. $initiate\ NumMig\ to\ 0\ and\ Chosen_{Move} = \{\}$
2. $Move_{Set} = Initialize\_Moves(\ PSSF(Types\_of\_Moves), Cur_{place})$
   $\rightarrow Returns\ a\ group\ of\ (Movement, Price)$
3. $While\ (NumMig < Budget)$
   do
   $Movement_{Profit} = InfoLeakRed(Movement_{Set}, Cur_{place}, Client_{Constraint})$
   $\rightarrow Returns\ a\ group\ of\ (Movement, Price, Profit)$
$Movement = PickBest_{move}(Movement_{Profit})$
$Chosen_{Move}.insert(Movement)$
$NumMig += move.price()$
$Cur_{place} = Update\_Placement(Cur_{place}, movement)$
$Movemrnt_{group} = Update\_Moves(Movement_{group}, movement)$
$end\ While$
4. $Return\ Chosen_{Movement}$
$end\ function$

4. Experimental Results and Its Analysis

The Experimental Results for the proposed dynamic PSSF based VM placement algorithm is given in this section. The performance of this approach is carried out using 20 different Virtual Machines in 4 different servers. The Experimental setup for the proposed approach is implemented using a local Open Stack Icehouse deployment for the test bed armed with 2.50 GHz 64-bit Intel Xeon CPU L5420 processor having 8-cores, 16 GB RAM, 500 to 1000GB disks, and two network interfaces with 100Mbps and 1Gbps speed. Every System executes on Ubuntu 14.04. For this experimentation, numbers of users are similar to the number of servers present and the initial structure comprises of 2 VMs per user. For each iteration, 15% of new VMs will attain and 15% of prevailing VMs will leave, generating continuous churn for each iteration. The migration budget was set to 15% for validating the experimentation and an ILP solution. The power sites are aligned to execute the CPU continuously at complete speed so as to minimize the measurement noise. The virtual machines employed in the executed the 64-bit version of Windows 7 Enterprise Edition and have 2 GB of RAM. This is suggested minimal quantity of memory for SPEC 2006 CPU standard.

4.1 Metric for Side Channel leakage

1. Side-Channel Vulnerability Factor (SVF): SVF is a parameter and method for evaluating a side channel's leakiness. It depends on the surveillance that there are two appropriate parts of data in a side channel attack: the data that an invaders is attempting to attain (delicate information), and information that an invaders could truly acquire. To estimate leakiness, it merely ought to calculate the correlation amongst these two parts of datasets.
2. Signal-to-noise ratio (SNR): the SNR of the signal x as is given as $\frac{x(k)-\bar{x}}{\sqrt{Var(x)}}$, where $x(k)$ is the value of the signal taken at the key, and $\bar{x}$ and $Var(x)$ are the mean and variance of x respectively. SNR is used to compare the level of the desired signal to the level of the background noise. It measures how complex it is for the invaders to obtain beneficial knowledge from the noise.

Table 1. Comparison of different Mitigation Technique for SVF and SNR

Mitigation Techniques	SVF	SNR
PSSF VM Placement Policy	0.23	0.358
Hybrid Mitigation	0.15	0.268
Proposed Dynamic VM Placement Policy	0.135	0.215

Table 1 refers to different mitigation approaches for cache based side channel attack. The comparison SVF and SNR values are given in this table. From the table we can infer that the SVF and SNR values are less in the proposed Dynamic VM Placement Policy compared to the hybrid and PSSF VM Placement policy which in turn indicates that, the leakage of information in this model is less compared to other approaches.

4.2 Algorithms Efficiency Evaluation

In this paper, hit rate, loss rate and Resource loss are considered to be measures for Algorithm efficiency evaluation. The mischievous consumers would be capable to achieve greater rate of hit through merely applying for huge VMs once. The mischievous residents will be capable to cause a higher loss rate of victim

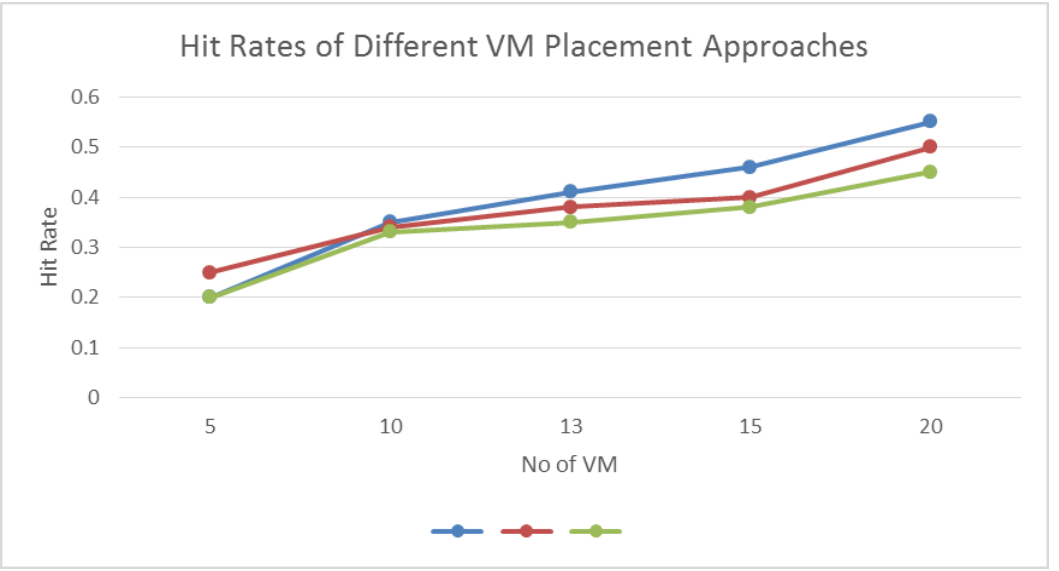

Fig.1. Hit rates of Different VM Placement Approaches

residents through applying for VMs in batches. Fig 1 represents the Hit rates of different VM placement Approaches. From this figure it is clearly shown that the hit rate of the proposed dynamic approach is very less compared to the other approaches.

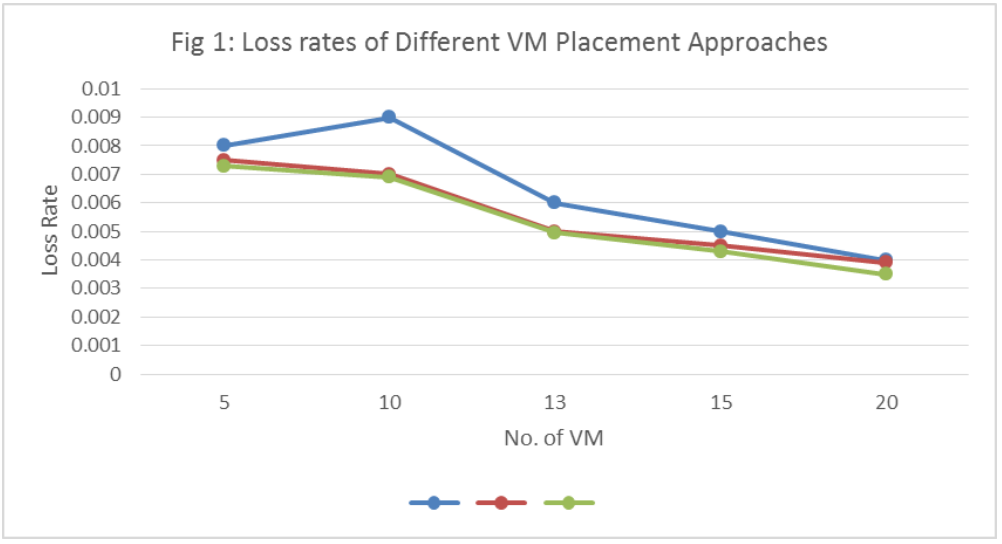

Fig.2. Loss rates of Different VM Placement Approaches

Fig 2 represents the loss rates of different VM placement Approaches. From this figure it is clearly shown that the loss rate of the proposed dynamic approach is very less compared to the other approaches. Fig 3 represents the Resource Loss of different VM placement Approaches. From this figure it is clearly shown that the Resource loss of the proposed dynamic approach is very less compared to the other approaches.

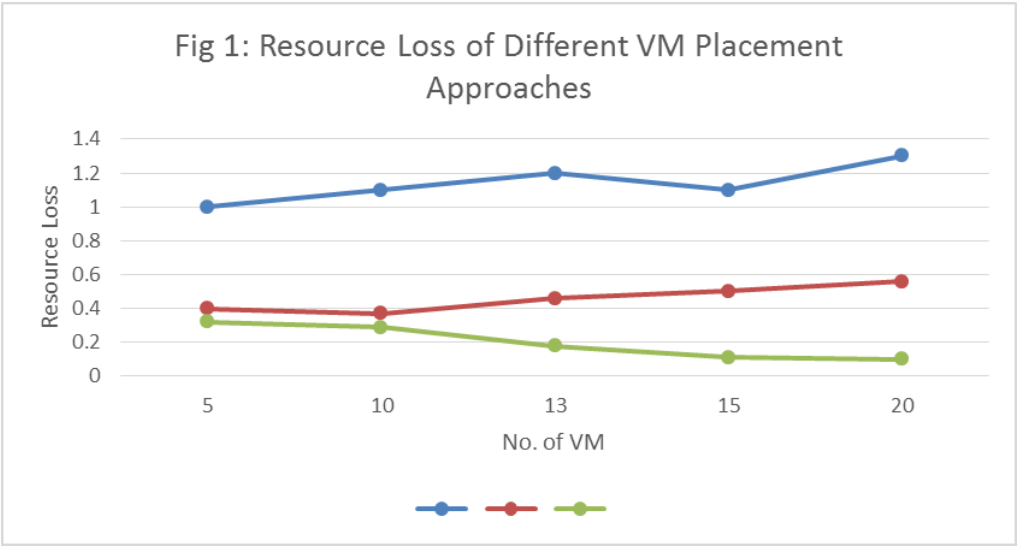

Fig.3. Resource Loss of Different VM Placement Approaches

5. Conclusions

Security is the foremost issues faced by the cloud computing environment while sharing resources over the internet. Users could face novel security risks whenever they employ cloud computing environment. The primary goal of the proposed dynamic VM placement algorithm is to mitigate the co-resistant attacks in clouds environment and minimize the information leakage amongst the shared virtual machines of different users in the same servers. This issue is addressed by introducing the dynamic methodology for the VM Placement policy. The proposed methodology is implemented in two phases. In first phase the set of moves for the VM of different users in different servers are obtained and in the subsequent phase these set of moves are employed for the VM baseline greedy placement algorithm.

References

[1] Y. Zhang, A. Juels, M. Reiter, and T. Ristenpart, "Cross-VM Side Channels and Their Use to Extract Private Keys," Proc. 19th ACM Conference on Computer and Communications Security (CCS 2012), pp. 305-316, 2012.

[2] Y. Han, T. Alpcan, J. Chan, and C. Leckie, “Security games for virtual machine allocation in cloud computing,” in 4th International Conference on Decision and Game Theory for Security, Fort Worth, TX, 2013.

[3] Zhu Jianrong, Li Jing and Zhuang Yi “Utility-based Virtual Cloud Resource Allocation Model and Algorithm in Cloud Computing” International Journal of Grid Distribution Computing Vol.8, No.2 (2015), pp.177-190

[4] Bhrugu Sevak “Security against Side Channel Attack in Cloud Computing” International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-2, December 2012.

[5] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds,” in Proc. of the ACM Conference on Computer and Communications Security (CCS), Chicago, IL, 2009.

[6] Y. Zhang, A. Juels, A. Oprea, and M. K. Reiter, “Homealone: Co-residency detection in the cloud via side-channel analysis,” in Proc. of the IEEE Symposium on Security and Privacy, Oakland, CA, 2011.

[7] Z. Xu, H. Wang, and Z. Wu, “A measurement study on coresidence threat inside the cloud,” in Proc. of the 24th USENIX Security Symposium (USENIX Security), Washington, DC, 2015.

[8] A. Herzberg, H. Shulman, J. Ullrich, and E. Weippl, “Cloudoscopy: Services discovery and topology mapping,” in Proceedings of the ACM Workshop on Cloud Computing Security Workshop (CCSW), Berlin, Germany, 2013.

[9] V. Varadarajan, Y. Zhang, T. Ristenpart, and M. Swift, “A placement vulnerability study in multi-tenant public clouds,” in Proc. of the 24th USENIX Security Symposium (USENIX Security), Washington, DC, 2015.

[10] Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting, “An exploration of l2 cache covert channels in virtualized environments,” in Proc. of the 3rd ACM Workshop on Cloud Computing Security Workshop (CCSW), Chicago, IL, 2011.

[11] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, “Detecting co-residency with active traffic analysis techniques,” in Proc. of the ACM Workshop on Cloud Computing Security Workshop (CCSW), Raleigh, NC,2012.

[12] Bhattacharya, J., Vashistha, S.: Utility computing-based framework for e-governance. In: Proceedings of the 2nd International Conference on Theory and Practice of Electronic Governance, ICEGOV’08, Cairo, pp. 303–309. ACM, New York (2008). doi:10.1145/ 1509096.1509160.

[13] Caracas, A., Altmann, J.: A pricing information service for grid computing. In: Proceedings of the 8th ACM/IFIP/USENIX International Middleware Conference: 5th International Workshop on Middleware for Grid Computing,MGC'07, Newport Beach, pp. 4:1–4:6. ACM, New York (2007). doi:10.1145/1376849.1376853.

[14] Kephart, J.O.: Autonomic computing: the first decade. In: Proceedings of the 8th ACM International Conference on Autonomic Computing, ICAC'11, Huddersfield, pp. 1–2. ACM, New York (2011). doi:10.1145/1998582.1998584.

[15] Maggio, M., Hoffmann, H., Santambrogio, M.D., Agarwal, A., Leva, A.: Decision making in autonomic computing systems: comparison of approaches and techniques. In: Proceedings of the 8th ACM International Conference on Autonomic Computing, ICAC'11, Karlsruhe, pp. 201–204. ACM, New York (2011). doi:10.1145/1998582.1998629.

[16] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the clouds: A berkeley view of cloud computing. Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley (Feb 2009).

[17] Soo-Jin Moon, Vyas Sekar, Michael K. Reiter, "Nomad: Mitigating Arbitrary Cloud Side Channels via Provider-Assisted Migration", Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1595-1606, ACM, 2015.

Authors' Profiles

Adi Maheswara Reddy G: Research Scholar, Department of CSE, JNTUK, Kakinada, AP, INDIA. Completed M.Tech in Computer Science & Engineering from JNTU and currently working as Manager, Software Engineer in PAREXEL INTERNATIONAL, having overall 11 years of experience in field of Information Technology. Major research interests include Parallel Programming, Distributed Systems, Network Security and Cloud Computing.

Dr. Koduganti Venkata Rao received the Ph.D degree from Andhra University in Computer Science and Systems Engineering in 2008. He is currently Professor in the department of Computer Science and Engineering and Dean IQAC at Vignan's Institute of Information Technology, Visakhapatnam. His major research interests include key management, authentication protocols and light weight protocol analysis and design in networks.

Dr.J.V.R.Murthy working as Professor in the Department of Computer Science & Engineering, University College of Engineering, Kakinada, and Director Incubation Center JNTUK. Held the positions Director Industry Institute Interaction Placement & Training, JNTUK, Director CoERD and Chairmen Board Of Studies CSE & IT JNTUK. A.P., INDIA. Over 24 years of Teaching, Research and Industrial experience in the field of Computer Science with specialization in Data warehousing and Mining. Started the career as computer programmer and occupied various positions such as lecturer, Assistant professor and professor.

How to cite this paper: Adi Maheswara Reddy G, K Venkata Rao, JVR Murthy," Previously-Selected-Server-First based Scalable VM Placement Algorithm for Mitigating Side Channel Attacks in Cloud Computing", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.8, No.1, pp. 50-59, 2018.DOI: 10.5815/ijwmt.2018.01.06