Journal of Theoretical and Applied Information Technology
31st March 2017. Vol.95. No 6
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A HYBRID MITIGATION TECHNIQUE FOR CACHE ASSISTED SIDE CHANNEL ATTACK IN CLOUD COMPUTING

[1]ADI MAHESWARA REDDY G, [2]K VENKATA RAO, [3]JVR MURTHY

[1]Research Scholar, Department of CSE, JNTUK, Kakinada, AP, INDIA

[2]Department of CSE, Vignan Institute of Information Technology Visakhapatnam, AP, INDIA

[3]Department of CSE, JNTUK College of Engineering, Kakinada, AP, INDIA

Email Id: [1]adi.maheswar.2010@gmail.com

ABSTRACT

Cloud computing has attained extensive approval for administrations in addition to individuals by presentation computation, storage and software aided amenities. Since, the Cloud services become further prevalent, approaches in the current past has revealed liabilities exclusive to such systems. Different from mainstream computing, the framework assisting Cloud permits communally suspecting individuals to instantaneously access an essential cache consequently encouraging a threat of data leakage through virtual machines through side channels. Therefore, numerous mitigation techniques has been developed by scholars for various side channel attacks. In this paper, author chiefly focuses on cache aided side channel attacks and a novel hybrid mitigation approach is suggested that amalgamated both the sequential mitigation approach and parallel mitigation approach. This technique is mainly employed when the cache is accessed through single core and through multiple core. The experimental results for the proposed approach is carried out using SPEC 2006 specifications and has been shown that the efficiency and usefulness of the cache memory is more in the proposed approach compared to the existing sequential and parallel schemes.

Keywords: *Cloud Computing, Cache Aided Side Channel Attack, Virtual Machines, Cache Wait, Cache Partitioning*

1. INTRODUCTION

Cloud computing is a novel framework variation with service prototypes in distributed computing and has attained higher focus in the investigation societies. It is a prototype for permitting appropriate, on-demand network access to a communal device of configurable computing assets like networks, servers, stowage, applications and other amenities) that could be quickly provisioned and freed with minimum organization effort or service providing interaction. Global network, scalability, minimization in cost and elasticity are certain characteristics that made Cloud Computing the subsequent generation construction of IT enterprise. Cloud computing provides the benefits of scalability, agility, reliability, security, performance, and maintenance to enterprises and has emerged to become a reality.

Infrastructure-as-a-service (IaaS) is a cloud computing paradigm where cloud provider leases out computing resources to cloud customers/tenants in terms of virtual machines (VM). Multiple VMs may implement on the similar physical machine and share entire essential hardware resources. Recent investigation has revealed that this sharing might permit allow a malicious VM to mine complex data from neighbor VMs, a solemn security concern for cloud adoption [1, 2, 3, 4, 5]. Several aspects are pierced out to be a hindrance to the adoption of cloud computing like working on complex information external to the enterprise, shared information, in effectiveness of encryption, amongst others.

Side-Channel is a mode of circumventing virtual machine for obtaining information from the physical execution instead of brute force or hypothetical weaknesses in the approach. Instances comprises of validating if two virtual machines are neighbor [3], and further hazardously, abstraction of encrypting private keys from innocent hosts [5]. Identical investigation for side- channels obtains added attacks that could be employed to the cloud [6, 7, 8]. Whenever concentration was initially obtained to side-channels [9] there were numerous suggestions for how to alleviate their possibilities [6], [10].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Side-channels are employed to drip data through virtual obstacles in traditional systems. Cache aided side-channels are employed to disrupt the enhanced encryption frameworks and data encryption standard (DES) prototypes. For a cache aided side-channel attack to be probable, the invader and the victim need to share certain levels of cache. By means of current hardware, it is conceivable in two dissimilar means: One is for the two procedures to attain consecutive access to the cache, and another one is for them to attain simultaneous access. Sequential access is attained through procedure context switch on the similar processor core, whereas simultaneous access necessitates a cache to be shared amongst numerous cores and is a constraint of the hardware.

Pertaining to above given situations, the need of consecutive access and simultaneous access at a single will not be achieved from the existing parallel and sequential cache aided side channel approaches. Thus, there is need of a hybrid approach that could access both sequentially and in parallel situations where the single cache can work with single core or multiple core that leads to the sequential and parallel access of data. There are some of the existing mitigations techniques that are employed separately for the sequential side channel attack and parallel side channel attack. Therefore, this paper endeavored to introduce a novel hybrid approach that functions efficiently with the sequential and parallel cache access. In addition, the proposed hybrid mitigation technique also minimized the drawbacks of cache-wait approach and cache partitioning approach and maximized the benefits of both the approaches.

1.1 Organization of the paper

A brief introduction to the cloud computing environment and side channel attacks along with the motivation for the proposed approach is given in this section. In section 2, numerous existing approaches for side channel attacks and its mitigations techniques are briefly discussed. The proposed hybrid mitigation technique for cache aided side channel attacks in section 3. The experimental outcomes and its brief investigation for the suggested approach is presented in section 4 followed by conclusions and references given in section 5 and section 6.

2. LITERATURE SURVEY

Earlier research have indicated that cache aided side-channels could be exploited in the Cloud to gather data. The consideration presented side channels in non-Cloud atmosphere from 70s [9], there are numerous outcomes to the issues suggested [6, 10]. The initial standard procedure of a cache aided side-channel attack exploited in the cloud was given by Ristenpart et al. [3] who revealed the usage of side-channels to validate virtual machine co-residence in Amazon's EC2 and is also a sequential type of side channel attck. As the fragment of this approach, they exploited the usage of priory recognized cache aided side-channel approaches that are denoted as the Prime+Probe approach, in a cloud atmosphere. The need of this experimentation, employing the PTP approach, was to consider if the cache aided side channel might be developed amongst two visitor domains in the cloud. The channel was constructed such a way that the initial VM (denoted as probing instance) could obtain a message that the subsequent VM (the target instance) encrypts in its usage of the cache.

Kim et al. [11] have built an outcome for cache aided side-channels in Cloud systems. In its solution, they averted cache aided side-channels through obtaining for every VM exclusive access to a segmented portion of the cache they demand a stealth page. So as to have software applications admittance, these concealed sheets and its outcomes necessitates the individual to make client-side amendments to the software being implemented in the guest VM. Hund et al. [12] used a cache aided timing side channel attack to de-randomize kernel space ASLR so as to accurately perform code-reuse attacks in the kernel address domain. Since we build our system on techniques verified to be an efficient against code-reuse attacks, our dynamic control-flow diversity with NOP insertion is a perfect fit to defend in depth against this attack. To specifically target cache-based attacks, Page [10] proposed partitioning the cache into disjoint configurable sets so that a sensitive program cannot share cache resources with an attacker. However this would require a radical change to current cache designs.

Bernstein [13] suggested the addition of a new CPU instruction to load an entire table into L1 cache and perform a lookup. This approach provides consistent cache access behavior regardless of input, and as such would eliminate cache side channels through table lookups. Wang and Lee [14] also proposed two new hardware cache designs to mitigate cache side channels: PLcache and RPcache. PLcache has the new capability of locking a sensitive cache partition into

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

cache, while RPcache randomizes the mapping from memory locations to cache sets. While these techniques are powerful mitigations against cache side-channel attacks, they all require additional hardware features which major processor vendors are unlikely to implement.

[15] shows how such LLC side channel attacks can be defeated using a performance optimization feature recently introduced in commodity processors. Since most cloud servers use Intel processors, it illustration how the Intel Cache Allocation Technology (CAT) could be employed to deliver a system-level defense mechanism to defend from side channel attacks on the shared LLC. CAT is a way based hardware cache based segmenting approach for imposing quality-of-service pertaining to LLC occupancy. Nevertheless, it cannot be directly used to overthrow cache side channel attacks due to the very limited number of partitions it provides. We present CATalyst, a pseudo-locking mechanism which uses CAT to partition the LLC into a hybrid hardware software managed cache. [16] provides a defense capable of preventing cache-based side-channels in the Cloud while not interfering with the Cloud model and the system against canonical attacks is validated while demonstrating that they can be prevented without client-side or hardware modifications.

3. PROPOSED HYBRID MITIGATION TECHNIQUE FOR CACHE BASED SIDE CHANNEL ATTCAK

A novel hybrid approach for mitigation of cache aided side channel attack is suggested in this paper. Usually, every approach that is employed for defending the cache based side channel attack has its own benefits and drawbacks. In this paper, the author attempted to hybridize both cache flushing-waiting approach suggested for sequential based side channel preventive measure and cache partitioning approach suggested for parallel based side channel preventive measure. Generally, the caching flushing-waiting approach has the drawback of minimized usefulness of cache and augmented price. At the same time, the cache based partitioning approach had drawback of reduced efficiency in the cache's usage. Since, the usage of cache in VM is too small, it could negatively impact the performance. So as to overcome, the above given issues and minimize the disadvantages of both the techniques, the hybrid mitigation technique is introduced. The proposed hybrid approach is employed on the existing Prime-Trigger-Probing (PTP) techniques as to efficiently make the usage of cache. The complete cache memory is partitioned into two partitions for the usage of different virtual machines where one part is functioned using the cache-wait function approach and the other part is functioned using the cache partitioning approach. A brief explanation for these techniques are given in below section. The block diagram from the proposed hybrid mitigation technique is given in Fig 1.

3.1 Sequential Cache Wait Approach

In case of the sequential approach, the cache wait function is employed. This methodology comprises of two novel methods within the virtual machines: One is a server that runs the procedure to make the cache execution to a wait stage (cache-wait), and another one is a tainting approach in the scheduler for determining when waiting the cache execution is essential. Cache-Wait function waits the cache execution for the specific time and operates only in one condition. When the context switch happens and the current domain requires data, both from the cache memory (cache hit) and the main memory (cache miss). And if the total time required for accessing the main memory is higher than the cache memory then Cache-Wait function initiates the operation as shown in Fig 1. That is, when context switch happens and the current VM performs cache hit and cache miss. If the time taken for the cache miss is greater than the cache hit then Cache-Wait function operates. The function is such that Cache-Wait will hold the cache execution process for the specific time that is determined from the difference in the accessing time required for fetching data from the main memory and the cache memory. Nothing, but the difference in accessing time required amongst cache miss and cache hit. The methodology for this tainting approach is given in Algorithm 1.

$$
\begin{aligned}
&Function\ contextSwitch(VM1, VM2)\\
&\quad \{\ \#\ From\ VM1\ to\ VM2\\
&\qquad if\ fetch_{main_{time}}\\
&\qquad > fetch_{cache_{time}}\\
&\qquad cache\_wait();\\
&\qquad return;\\
&\}\ EndFunction
\end{aligned}
$$

Algorithm 1: Pseudo-code for Tainting Approach in Cache – Wait Function

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2 Parallel Cache Partitioning Approach

In case of parallel approach, the cache partitioning technique using cache coloring function is employed. In this technique, the shared cache memory is partitioned into a numerous smaller segments, known as pieces or segments, and allowed every VM access to a subsection of these segments. In this manner, every cache line pertains to merely single partition. Thus, if two VMs are specified dissimilar segments of the cache that are impotent to eject, or else touch, each other's cache lines. The influence of this segment is performed on the existing PTP Approaches. There are two cache segments obligated on cache data. One with the red dotted line specified as Partition 1 that is employed for VM1 nothing but the Probing instance and the other with the green dotted line specified as Partition 2 that is employed for VM2 that is the Target instance as given in the block diagram of the proposed approach in Fig 1.

When VM1 attempts to prime the cache, it merely primes cache lines in its division, i.e., the initial three lines in the second partition, and has left another four partitions vacant. Whenever the triggering instance that is VM2 is employed formerly it efforts to change another cache line. It could merely perform in its individual division and consequently terminate ejecting no VM1's data. Whenever probing instance again reads from the cache lines which is available, it could consider no change whenever it left, and consequently no communication was capable to happen.

The effective partitioning in the Cache Partitioning approach is performed using the cache coloring approach. Cache coloring is an approach through which memory pages are plotted to cache lines through clustering or partitioning known as colors. The mapping of memory addresses to cache positions, denoted as cache lines are executed in the hardware. In spite of its initial usage like an optimization agent, the specified mapping of memory addresses to cache lines could be exploited for other issues. Thus, the cache line or cache partition memory address plots to could be specified through considering at its least significant bits. Every partition is considered with dissimilar colors having numerous bit value assigned. These bits are referred as the page color, since the pages that share the bits in mutual would be capable to compete for the similar cache lines. In this way, the hypervisor could allot pages to guest VMs depending on color, constraining a VM's cache access to subset which follows the bit combination. Therefore, this kind of cache coloring is employed to impose as to restrict every VM to a distinctive segments, or clusters of segments, of the shared cache.

4. EXPERIMENTAL RESULTS AND ITS ANALYSIS

The performance of proposed hybrid mitigation approach is measured through the execution to evaluate the effectiveness and realism of the hybrid mitigation technique. The approach implemented on an HP Z400 work place having a 2.67 GHz 4 core Intel Xeon W3520 CPU with 16 GB of DDR3 RAM. The cores were executing at 2.8 GHz. Every CPU core has a 32 kB 8-way L1 D cache, a 32 kB 4-way L1 I-cache and a 256 kB 8-way L2 cache. Additionally, the four cores share an 8 MB 16-way L3 cache. The machine implemented a 64-bit version of Windows Server 2008 R2 HPC Edition. 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 has 2 GB of RAM. This was the suggested minimal quantity of memory for SPEC 2006 CPU benchmark.

The Experimental Results for the proposed approach is carried out using five different virtual machines against average number of functions for the existing and proposed approaches. The proposed hybrid mitigation technique is compared with the existing sequential side channel attack approaches such as cache flush technique and Cache wait Technique. Fig 2 represents the comparison of different approaches against the sequential scheme [3] having average number of functions employed in the approach with respect to the number of virtual machines. Fig 3 represents the comparison of parallel existing approach [16] that is cache partitioning approach and proposed hybrid mitigation approach with average number of functions with respect to the number of virtual machines.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

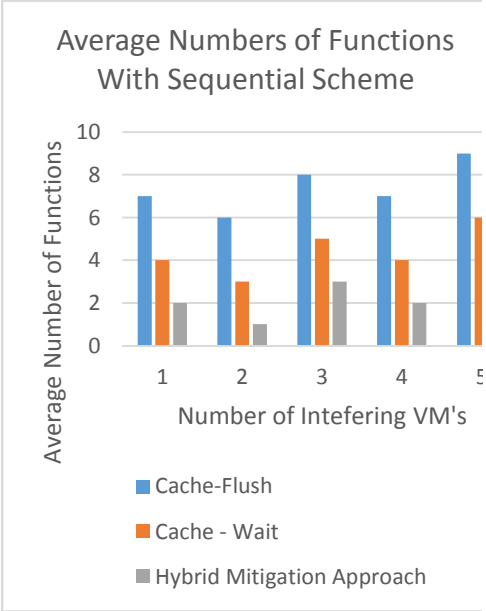

Fig 2: Average Numbers of Functions with Sequential Scheme

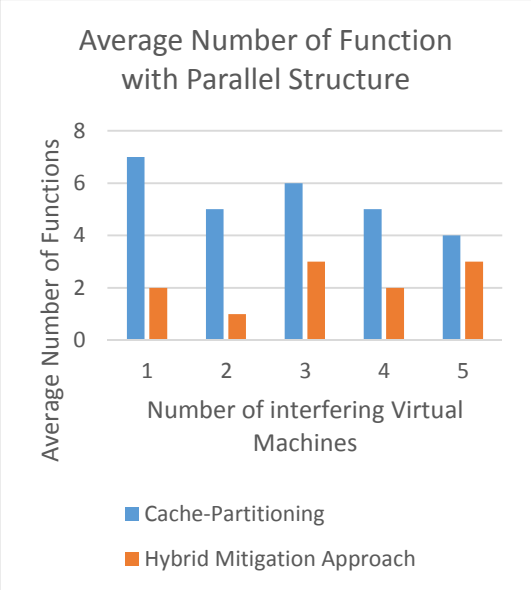

Fig 3: Average Number of Functions with Parallel Structure

4.1 Metric for Side Channel leakage

1. Side-Channel Vulnerability Factor (SVF): SVF is a metric and method for evaluating a side channel's leakiness. It depends on the observation that there are two appropriate parts of data in a side channel attack: the data that an attacker is attempting to attain (delicate information), and the knowledge an attacker 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 difficult it is for the attacker to figure out useful information from the noise.

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

Mitigation Techniques	SVF	SNR
Cache Flush	0.874	1.177
Cache Wait	0.434	0.934
Cache Partition	0.395	0.89
Hybrid Mitigation	0.15	0.268

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 hybrid approach which in turn indicates that, the leakage of information in this model is less compared to other approaches.

5. CONCLUSIONS

Cloud computing assures enhanced effectiveness, nevertheless opens up novel threats pertaining to the sharing of hardware through mutually distrustful individuals. Cloud's architecture is particularly vulnerable to cache driven side channel attack. In this paper, a novel hybrid mitigation approach is suggested where the sequential approach and the parallel approach is combined. The cache wait approach of the sequential scheme and cache partitioned approach of the parallel scheme are both developed in a single cache memory with different virtual machines where a single cache memory has the access to single core systems and multiple core systems. This proposed approach prevent cache

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

aided side-channels and respect the association amongst the cloud provider and the customer. The experimental results of the proposed approach is carried out using the SPEC 2006 specification with different virtual machines and results showed that proposed approach has higher efficiency compared to the existing approaches. However, the computational complexity is expensive when compared to the other existing approaches and need more resources for computation.

REFERENCES

[1] G. Irazoqui, T. Eisenbarth, and B. Sunar, "S$A: A shared cache attack that works across cores and defies VM sandboxing – and its application to AES," *in IEEE Symposium on Security and Privacy (S&P),* May 2015.

[2] F. Liu, Y. Yarom, Q. Ge, G. Heiser, and R. B. Lee, "Last-level cache side-channel attacks are practical," *in IEEE Symposium on Security and Privacy (S&P),* May 2015, pp. 605–622.

[3] 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 ACM conference on Computer and communications security (CCS*), Nov. 2009, pp. 199–212.

[4] Y. Zhang, A. Juels, A. Oprea, and M. K. Reiter, "Homealone: Co-residency detection in the cloud via side-channel analysis," *in IEEE Symposium on Security and Privacy (S&P),* May 2011, pp. 313–328.

[5] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-VM side channels and their use to extract private keys," *in ACM Conference on Computer and Communications Security (CCS*), Oct. 2012, pp. 305–316.

[6] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: The case of AES," in Proc. Cryptographers' Track RSA Conf. Topics Cryptol., 2006, pp. 1–20.

[7] D. X. Song, D. Wagner, and X. Tian, "Timing analysis of keystrokes and timing attacks on SSH," in Proc. 10th Conf. USENIX Security Symp., 2001, vol. 10, p. 25.

[8] Y. Tsunoo, T. Saito, T. Suzaki, and M. Shigeri, "Cryptanalysis of DES implemented on computers with cache," in Proc. 5th Int. Workshop Cryptograph. Hardware Embedded Syst., 2003, pp. 62–76.

[9] B. W. Lampson. (1973, Oct.). A note on the confinement problem. Commun. ACM [Online]. 16(10), pp. 613–615. Available: http:// doi.acm.org/10.1145/362375.362389.

[10] D. Page. (2003). Defending against cache-based side-channel attacks. Inf. Security Tech. Rep. [Online]. 8(1), pp. 30–44. Available: http://www.sciencedirect.com/science/article/pii/S1363412703001043

[11] T. Kim, M. Peinado, G. Mainar-Ruiz, STEAL THMEM: System-level protection against cache-based side channel attacks in the cloud, in Proc. 21st USENIX Conf. Security Symp., Berkeley, CA, USA, 2012, p. 11.

[12] R. Hund, C. Willems, and T. Holz, "Practical timing side channel attacks against kernel space ASLR," *in Proceedings of the 34th IEEE Symposium on Security and Privacy (S&P '13*), 2013.

[13] D. J. Bernstein, "Cache-timing attacks on AES," Preprint, 2005

[14] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache based side channel attacks," *in Proceedings of the 34th International Symposium on Computer Architecture (ISCA '07*). ACM, 2007, pp. 494–505.

[15] Fangfei Liu, Qian Ge, Yuval Yarom, "CATalyst: Defeating last-level cache side channel attacks in cloud computing", *International Symposium on High Performance Computer Architecture (HPCA),* IEEE, 2016.

[16] M. Godfrey and M. Zulkernine, "A server-side solution to cache based side-channels in the cloud," *in Proc. IEEE 6th Int. Conf. Cloud Comput*., 2013, pp. 163–170.

Journal of Theoretical and Applied Information Technology
31st March 2017. Vol.95. No 6
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

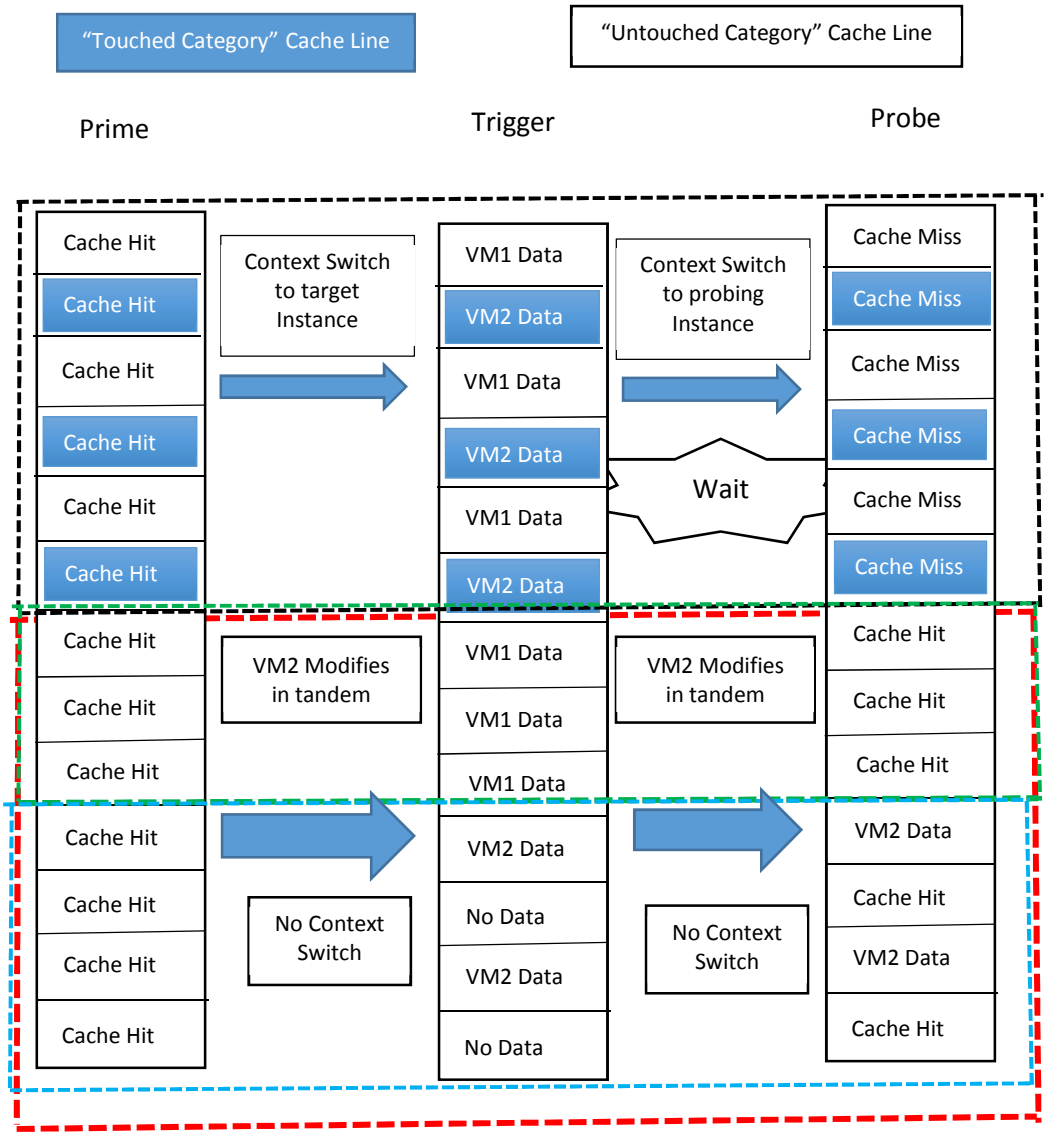

Fig 1: Block Diagram of the Proposed Hybrid Mitigation Approach