Cache Based Side Channel Attack: A Survey

Conference Paper · October 2018

DOI: 10.1109/ICACCCN.2018.8748811

3 authors, including:

Sandeep Saxena
IMS UNISON UNIVERSITY DEHRADUN
21 PUBLICATIONS **92** CITATIONS
SEE PROFILE

Prof(Dr.) Goutam Sanyal
National Institute of Technology, Durgapur
209 PUBLICATIONS **1,356** CITATIONS
SEE PROFILE

Cache Based Side Channel Attack: A Survey

Sandeep Saxena, Goutam Sanyal
[1]*Asst. Prof, Sharda University, Greater Noida*
[1,2]*National Institute of Technology, Durgapur, INDIA*
saxena.s.in@ieee.org,nitgsanyal@gmail.com

Manu
[1]*Sharda University, Greater Noida, UP*
manu@sharda.ac.in

Abstract

The Cloud Computing (CC) is famous due to shared resources technology. Cloud computing share resources among distrusting customers and provide on demand, cost effective, elasticity services. Due to rapid growth of cloud computing environment, vulnerabilities and their preventions methods are potential increase. We had seen that conventional prevention methods for Side Channel (SC) attack are not suitable for avoidance of cross-VM cashed based SC attacks.

In 2016, shared technology issues is a one of top threat consider by cloud security alliance (CSA), which has been published in February 2016 in *The Treacherous 12* [1]. This is a under top threat by CSA from last 5 year. In this paper we will discuss multiple method for performing side channel attack and prevention methods. We also discuss the strength of prevention method as well as drawbacks of that method. So that this paper will generate more research scope and new effective idea for prevention of side channel attack, this paper is provide support and background knowledge for new researchers in area of side channel attack in different environments.

Keywords: side channel attack, cross-VM, prime-probe method, cross-VM cached based attack, Virtual Machine (VM).

I. INTRODUCTION

Cloud computing is used to resolve the sharing of numerous, large scale, computing assets in to a single abstract entity that is generally known as cloud. This will allow multiple users to access all these resources concurrently, which similar to workstation model o f past decade [5]. Cloud computing is used some complex software technologies are need to combine all such individual machine into a distinct dynamic manageable resource.

Cloud computing allow users to outsource their required hardware, platform and software. Cloud computing provide the high computing resources on rent and that resource are very costly. Users are unable to purchase all that resources. Cloud computing basically used Virtualization technology for providing the On-demand service.

In present technology, side channel attack is one of top threat in cloud computing technology that is one of shared technology issues listed in *The Treacherous 12* [1]. Generally cloud service provider (CSP) promoted and delivers their services as scalable/elasticity by shared hardware/infrastructures, PaaS & SaaS, that is not secure for side channel attack because given architecture is not having strong isolation property. So given architecture leads to shared technology vulnerabilities. We need to secure application and user security, computer, storage and must be monitor our user as it is recommended by defense in depth strategy. By all these issues in present architecture leads to vulnerability to private key breaches/leaks that will compromise user system in entirely in cloud environment.

Presently most famous side channel attack is cache based attack in which attacks perform on cache memory and attacker extract the private key of victim.

Generally as shown in figure 1, processor has three types of cache memory, which names are L1, L2 and L3. L1 and L2 cache are same size and there are not shared among the multi core o f processor, which means each core has L1 and L2 cache of their own. On ly the L3 cache shared among the different core of processor and that's why side channel attack is possible in L3 cache. When the data is required by processor (for the register), then first required from L1 cache. L1 cache is divided in to separately into data and instruction cache. It will access most often and it is virtually indexed. That means mapping of cache location is determined by its virtual address (view by procedure) as different to its physical address (view by OS). That means cache is high speed but mapping is not sealed across contexts (different process will have different mapping) that why there is prospect of information leakage crossways L1 cache.

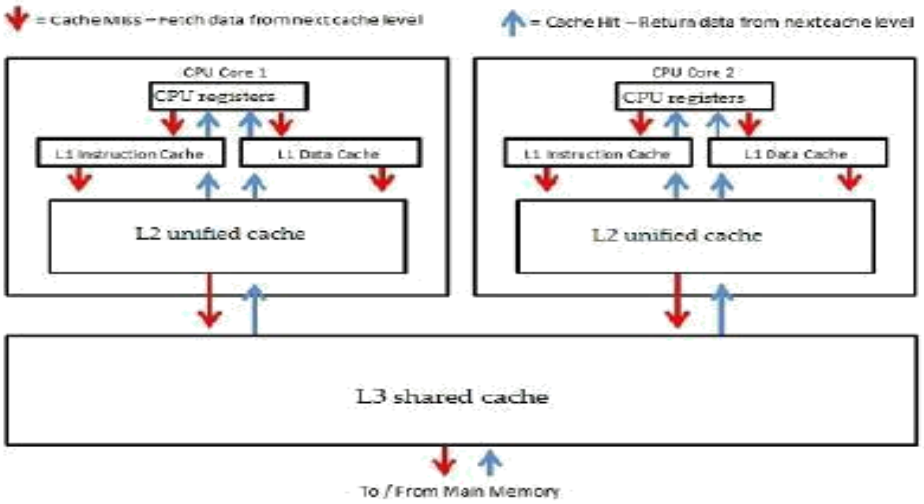

Figure 1: Cache memory hierarchy [12]

The L2 cache is normally much big than L1 cache and contains both instruction and data. In this mapping of memo ry address to cache locations are done via physical memo ry and are sealed across contexts. Processes which have pooled access to data may find those data located in cache exclusive of causing a cache miss. As this will leads to information leakage between processes. As above figure 1 mentioned that L3 cache store data from multiple core simultaneously. This scheme is distinctive that allow multiple CPU cores access to shared resources, and data leakage can happens. This feature is called multi-tenancy or co-residency in Cloud Computing.

Cloud computing one more attribute is called virtualization; it involves the concept of physical machine to operating system in VMs on equivalent physical machine. In virtualization, the hypervisor is responsible for the communication between VMs. Hypervisor uses sandboxing technique for provide logical segregation between VMs, but this is not only sufficient prevention against SC attacks. SC attacks are performing to acquire details about the access pattern of memory by using access time difference of another process like cryptographic algorithms that perform the encryption. Such types of SC attacks monitor the cache states and analyze the effect on the encryptions completing time during the execution. SC attacks bypass the isolation provided by sandboxing mechanism.

Cryptographic devices are just like black boxes, in which attacker only giving the inputs and corresponding outputs after calculation, what is going inside the device is entirely hidden from attacker. SC attacks try to gain more information concerning the data used in the device. SC attacks use physical process to extract confidential information of the victim VM fo r example private encryption key.SC attack is physical attack so it uses all kind of physical quality of computation like time consumes to run a program, power consumption during a process execution, electromagnetic radiation and temperature to extract the secret information.

Some of the general Classes of SC Attacks are:

- Cache based attack
- Timing attack
- Power monitoring attack
- Electromagnetic attacks
- Data remanence
- Differential fault analysis

These attacks are comes under the hardware-based side channel attack as well as power analysis, bus probing and a software based side channel attacks includes timing attack, cache attack and memory attack.

RSA cryptographic algorithm is widely used with web browser in SSL connection and data transfer over internet. Therefore cached based SC attack easily harms the web browser.

In *cache based side channel attack* , attacker and victim resides on same hardware and attacker is able identified cache memory spaces allocated for victim to store decryption functions like , square (S), Reduce (R) and Multiply (M). As we know cache memory size is very small in comparison with main memory and as per Cache memory mapping, some particular space data of main memory is stored in fixed location of cache memory so attacker use this vulnerability and store

Some of his memory data in same location where victim S, M and R function are stored and called repeatedly. He will access his data from cache with periodically by any process with very short interval and he will store access time. If total access time is greater than cache access time its means data is not available in cache and first fetch from main memory then provide access for particular process so its means cache miss is performed otherwise cache hit performed. With the help of cache hits and miss he is able to find out which functions is called by victim system and extract decryption key. Once attacker get victim private key, victim system is compromised by attacker.

a) Business impact

A compromise of private key of victim system will increase the information disclosure and no information is confidential on victim computer. It will increase the elevation of privilege because private key is compromise and attacker may take more access rights on victim computer or their own system [1].

II. Literature survey on Side channel

Side channel attack is existed from the past [2], in cloud computing novel co-residency features which leads particularly in this context. As all knows that cloud computing is most popular and useful technology [3] because all new technology will develop according to cloud architecture but due to co-residency, a shared technology issues will raise increasingly. In present year many papers has been published on cloud computing vulnerabilities, specific on shared technology issues or data privacy issues [4].

Using exploitation of this vulnerability many attacker are extract private key of victim systems and able to see all private and confidential data of the victim computer. When two machines are co-resident [7], it is more dangerous that any attacker can extract ion cryptographic private key from unwary host [8]. Many papers are demonstrate the strictness of side channel attack in cloud environments and the possible for side channel attack when migrated to a cloud environment [9,10].

In [12] had developed novel cache design for protecting from side channel attack. They proposed novel security-aware cache designs, used the Partition-Locked cache (PLcache) and the Random Permutation cache (RPcache), analyze and establish their security, and estimate their performance.

Jingfei Kong et al., [13], study and analyses these novel cache designs and identify significant vulnerabilities and weaknesses of those new cache designs. They also proposed probable solutions and improvements over the unique new cache designs to beat the identified shortcomings.

Carles Hernandez et al., [4], proposed Random Modulo (RM), novel cache designs that provide the probabilistic behavior mandatory by Measurement-Based Probabilistic Timing Analysis (MBPTA) and with the subsequent advantages over existing MBPTA - acquiescent cache designs:

(i) An outstanding fall in worst-case execution time (WCET),
(ii) Lower latency and area operating cost, and
(iii) Economical average performance w.r.t usual caches.

Boris Kopf et al. [15], proposed a new way for automatically derive upper bounds on the quantity of information regarding the input that an opponent can extract from a program by analyzing the CPU's cache behavior. In their approach is a novel method for efficient including of concretizations of conceptual cache states that enables us to attach up to date techniques for stationary cache analysis and quantitative information. They implement their counting procedure on top of the Timing surveyor, one of the most superior engines for static cache analysis. They use their tool to perform a case study where they derived upper bounds on the cache leak of a 128-bit AES executable on an ARM processor with a practical cache configuration. They also analyze this functioning with a commonly suggested (but until now heuristic) countermeasure applied, obtain a prescribed account of the resultant increase in security.

a) *Motivation and Contribution*

This is a survey paper on side channel attack, as when someone starts work on side channel attack or shared technology issues [1], and then need one of paper on previous work and need to know the scope and importance of this area. So as per need of researchers, I did start writing this paper. This paper includes previous techniques for side channel attack as well as comparison and generates scope for new researchers for their future research.

III. Analysis of previous researches

Various researchers work on this area and proposed various method and techniques has been proposed as we did discuss in our previous section but still this problem is in existence, as we know this vulnerability added recently in CSA documents in shared technology issues [1].

a) Time Driven cache based SCA

Paper	Crypto System	Algorithm used	Severity
In [16]	Symmetric key	AES	HIGH
In [17]	Symmetric key	AES	HIGH(used Bonneu's attack process to test the weakness of AES)
In [18]	Asymmetric key	RSA	HIGH(use FLUSH+RELOAD Based Attack)
In [19]	Asymmetric key	ElGamal	HIGH(Using libgcrypt cryptographic library)
In [20]	Asymmetric key	DSA	HIGH(using lattice methods)
In [22]	Asymmetric key	DSA	MEDIUM (Using Spy Program)

b) Trace Driven Cashed based SCA

Paper Title	Crypto System	Algorithm used	Severity
In [21]	Asymmetric	AES	HIGH (Use two metrics: "expected number o f traces" and "average number of operations")
In [23]	Asymmetric	RSA	HIGH
In [24]	Asymmetric	AES	HIGH (proposed the numerous deductions -based algebraic side-channel attack to cope with the error in leakage capacity and to explo it new leakage Models)

IV. Comparison and Conclusion

Paper Title	**Method used**	**Implementation**	**Degradation in system performance**	**Improvements to prevent from SCA**	**Drawbacks**
In [12]	PLcache and RPcache	Implementation on M-Sim v2.0	**RPcache** :0.3% **PLcache** : 12% and 14% with floating point benchmarks and integer benchmarks	**RPcache**: 0.07%	**1. PLcache:** unnecessary locking could cause wrongness problems [2] ..... **2. RPcache** : vulnerable to crash based time driven attacks [2] ..... 3. St ill susceptible to software cache-based side channel attacks.[2]
In [2]	novel cache design to overcome the identified weakness in PLcache and RPcache	MIPS simulations		1. To secure the PLcache, one probable way out would be a pre-loading and locking all important information right before the crypto operation 2. when a procedure is not active, it will not lock its information in the data cache so that other process will not undergo from the reduced cache capacity	

In [3]	create a secret per-page memory mapping from virtual address offsets to physical address offsets	No implementation only gives method	Average overhead of only 2.5%.	Remove shortcomings in PLcache and RPcache	Reduced overhead
In [4]	Provide counter measures against Power analysis, traffic analysis, timing attack, fault analysis etc.	No implementation only gives counter measures	Not discussed	Not discussed	Increase overhead on system performance and computation
In [5]	Developed page coloring approach for the avoidance of cache based SCA	Cache is divided into Various parts. Each part is assigned a particular color. Each VM can access the cache part of assigned color for which it is allowed to access	Problem is that this security is gained at vast cost of performance deprivation and only up to certain boundary the performance degradation can be traded with protection	Authors solved the issue of cache interference and stop SCA that can be carried out to gain private data	Increase overhead and performance degradation
In [6]	Developed mitigation approach for access driven SCA in LLC	Linux	By this approach they avoid from the FLUSH+ RELOAD and Prime + Probe based attack in LLC with little performance overhead	Dynamically management for memory pages to disable sharing of LLC lines	small performance overhead
In [7] and [8]	Developed very light weight protection system CATalyst , for CSP and Clients	Virtual Machine (VM)	Much better then page coloring based scheme which provides an average degradation of 5.9%	1. Provide protection against SCA based on LLC. 2. Proposed technique is simpler as CAT, which isolate the cache partitions. 3. Page coloring scheme is miss memo ry deduplication [9] wh ich is available in CATalyst.	1. Cache capacity incurs very small performance degradation Approx 10% . 2. Drawback for benign applications that require the right to use to the high-resolution timers.

V. Conclusion and Future work

We have done work on this topic because of the cloud motivation in this research. First motivation was that cloud architecture is vulnerable to cache-based side channel attacks. And second was such attacks are not salved by any conventional method. So this cache-based side channel attack survey is much more important for any new researcher, which will interest to work in this domain in future.

REFERENCES

[1]. The Treacherous 12: Cloud Computing Top Threats in 2016, cloud Security Alliance in February 2016.

[2]. Dag Arne Osvik, Adi Shamir and Eran Tromer, Cache Attacks and Countermeasures: the Case of AES, 2005.

[3]. IDC. Enterprise it in the cloud computing era, 2008. ButlerW. Lampson. A note on the confinement problem. Commun. ACM ,16(10):613- 615, October 1973

[4]. IBM . Ibm mainframes, 2012, http://www.redbooks.ibm.com/redbooks/pdfs/sg247803.pdf.

[5]. Thomas Ristenpart, Eran Tromer, Hovav Shacham, Stefan Savage, Hey, You, Get Off of M y Cloud: Exploring Information Leakage in Third-Party Compute Clouds, CCS09, November 913, 2009, Chicago, Illinois, USA.

[6]. Yinqian Zhang, Ari Juels, M ichael K. Reiter, and Thomas Ristenpart. Cross-vm side channels and their use to extract private keys, CCS ’12, pages 305-316, New York, NY, USA, 2012. ACM .

[7]. Yukiyasu Tsunoo, Teruo Saito, Tomoyasu Suzaki, and M aki Shigeri. Cryptanalysis of des implemented on computers with cache. In Proc. of CHES 2003, Springer

[8]. LNCS, pages 62-76. Springer- Verlag, 2003.

[9]. Dawn Xiaodong Song, David Wagner, and Xuqing Tian. Timing analysis of keystrokes and timing attacks on ssh. In Proceedings of the 10th conference on USENIX Security Symposium - Volume 10, SSYM ’01, pages 25-25, Berkeley,CA, USA, 2001. USENIX Association.

[10]. Zhenghong Wang and Ruby B. Lee,"New Cache Designs for Thwarting Software cache-based Side Channel Attacks", ISCA’07, June 9–13, 2007, San Diego, California, USA,pp .494-505.

[11]. Jingfei Kong,Huiyang Zhou,Onur Aciicmez and JeanPierre Seifert, "Deconstructing New Cache Designs for Thwarting Software Cachebased Side Channel Attacks", CSAW’08, October 31, 2008, Fairfax, Virginia, U SA,pp .25-33.

[12]. Carles Hernandez, Jaume Abella, Andrea Gianarro, Jan Andersson and Francisco J.

[13]. Cazorlay,"Random M odulo: a New Processor Cache Design for Real-Time Critical Systems",ACM .

[14]. BorisKopf,LaurentM auborgneandM artin Ochoa,"Automatic Quantication of Cache Side-Channels",pp .1-20.

[15]. Tiri K., Acıiçmez O., Neve M ., Andersen F. (2007) An Analytical M odel for Time-

[16]. Driven Cache Attacks. In: Biryukov A. (eds) Fast Software Encryption. FSE 2007. Lecture Notes in Computer Science, vol 4593. Springer, Berlin, Heidelberg

[17]. Y. He, H. Guan, K. Chen and A. Liang, "A New Software Approach to Defend against Cache-Based Timing Attacks," 2009 International Conference on Information Engineering and Computer Science, Wuhan, 2009, pp . 1-4.

[18]. P. Zhou, T. Wang, G. Li, F. Zhang and X. Zhao, "Analysis on the parameter selection method for FLUSH+RELOAD based cache timing attack on RSA," in China Communications, vol. 12, no. 6, pp . 33-45, June 2015.

[19]. Yinqian Zhang, Ari Juels, M ichael K. Reiter, and Thomas Ristenpart. 2012. Cross-VM side channels and their use to extract private keys. In Proceedings of the 2012 ACM conference on Computer and communications security (CCS '12). ACM , New York, NY, USA, 305-316.

[20]. Acıiçmez O., Brumley B.B., Grabher P. (2010) New Results on Instruction Cache Attacks. In: M angard S., Standaert FX. (eds) Cryptographic Hardware and Embedded Systems, CHES 2010. CHES 2010. Lecture Notes in Computer Science, vol 6225.Springer, Berlin, Heidelberg

[21]. Acıiçmez O., Koç Ç.K. (2006) Trace-Driven Cache Attacks on AES (Short Paper). In: Ning P., Qing S., Li N. (eds) Information and Communications Security . ICICS 2006. Lecture Notes in Computer Science, vol 4307. Springer, Berlin, Heidelberg

[22]. Z. Ping, K. Yingzhan, C. Caisen and Z. Jilao, "Research on Key Technology for Data Cache Timing Attack on DSA," 2011 First International Conference on Instrumentation, M easurement, Computer, Communication and Control, Beijing, 2011, pp . 574-577.

[23]. CaiSen Chen, Tao Wang, YingZhan Kou, XiaoCen Chen, Xiong Li, Improvement of trace-driven I-Cache timing attack on the RSA algorithm, In Journal of Systems and Software, Volume 86, Issue 1, 2013, Pages 100-107, ISSN 0164-1212,

[24]. Xinjie Zhao, Shize Guo, Fan Zhang, Tao Wang, Zhijie Shi, Zhe Liu, Jean-François Gallais, A comprehensive study of multiple deductions-based algebraic trace driven cache attacks on AES, Computers & Security, Volume 39, Part B, 2013,Pages173-189