Cache based Side Channel Attack on AES in Cloud Computing Environment

D.Pratiba
Asst.Professor,
Dept. of CSE,
RVCE, VTU

G.Shobha, PhD
Professor,
Dept. of CSE,
RVCE, VTU

Sonali Tandon
B.E Student
Dept. of CSE
RVCE, VTU

Srushti S B
B.E Student
Dept. of CSE
RVCE, VTU

Vartika
B.E Student

ABSTRACT

As Cloud services become more pervasive, works in the recent past have uncovered vulnerabilities unique to such systems. The use of virtualization to isolate computational tasks from ones carried out by adversaries that co-reside with it is growing rapidly. This trend has been precipitated by the failure of today's operating systems to provide adequate isolation due to the growth of cloud facilities. Unlike mainstream computing, the infrastructure supporting a Cloud environment allows mutually distrusting customers to simultaneously access an underlying cache thus promoting a risk of information leakage across virtual machines via side channels. This paper attempts to set up a private cloud environment, demonstrates a cache based side channel attack and explores solutions to counterattack the same. An intense cache access pattern analysis is carried out, thus gathering information about the table lookup indices during one AES encryption to finally recover 128-bit full AES key.

Keywords

Cloud Computing, AES, Side channel attack

1. INTRODUCTION

Cloud is a network of computers hosted over the Internet. The various applications and services running on the systems over a distributed cloud network utilize virtualized resources, these can be utilized with common networking standards and Internet protocols. Cloud computing visualizes computing as a service where the cloud providers develop a pool of computing resources which can be configured to adhere to customer needs. The customers can dynamically attain and release the required resources according to their changing needs. Ubiquitous network, scalability, reduction in cost and flexibility are some of the features which make Cloud Computing the next generation architecture of IT enterprise.

The cloud services are unique as they have no customary boundaries. The cloud services are becoming common and the cloud itself is becoming a part of the global infrastructure. With the increase in usage of cloud, new vulnerabilities have been uncovered which are unique to such systems. The key to a cloud computing environment is Hardware Virtualization. This consists of multiple instances of virtual machines which utilize the resources on a physical machine. Overlapping usage of physical machine resources by the virtual machines.

Leads to improvement in the use of computing resources in terms of energy consumption and cost effectiveness. This sharing of resources can be used to create cache based side channels which promotes the risk of leakage of information across virtual machines. The Cloud supporting infrastructure is different from conventional computing as it allows the memory cache to be simultaneously accessed by mutually distrusting clients. This fulfils the requirement for a side channel attack.

Cache–based side channel attacks are currently believed to be most dangerous, among side-channel attacks. As the solutions to a side-channel are specific to the hardware medium being exploited, the scope of this paper is limited to side-channel attacks which exploit the CPU cache.

This paper demonstrates the cache based side channel attack between virtual machines where a malicious virtual machine owned by the attacker extracts AES encryption key from a victim virtual machine which has been spawned on the same physical machine. The software implementation of AES encryption algorithm uses many table lookup operations which in turn affect the cache. These lookup indices are closely related with the private key used for encryption and decryption. The leakage of this key can reveal lot of confidential information. The present paper suggests two techniques for mitigating this attack by disrupting the cache access patterns during the encryption of the algorithm.

2. RELATED WORK & MOTIVATION

A spy process analyzes the pattern of the cache accessed by an AES process is discussed in [1]. The side-channel vulnerabilities involving the CPU cache is detailed in [2]. The low-level implementation of the cache structure which results in an indirect interaction among the processes which run on the same processor that causes cross-process leakage of information is detailed in [3].The side channel attack on a symmetric multiprocessing (SMP) system which is virtualized using XEN is described in [4].

The need for the attacker to share the same physical infrastructure of the victim for the attack to take place is mentioned in [5]. A brief about the AES implementation based on the look-up tables is discussed in [6]. This also analyzed instruction caches and unified caches which are other cache features that affected the attack. The procedure to set up a cloud is shown by experiments in [7]. The cloud set up uses the XEN hypervisor and is according to Platform as a service (PAAS) model. The benefits of a private cloud are analyzed in [8]. It provides a technical overview of the XEN based virtualization and the two virtual domains. Managing of the secret cryptographic key in the cloud environment set up using OpenStack is dealt with in [9]. The differences between OpenStack and OpenNebula are described in [10].

However none of the papers above claimed that they can recover the full AES key through the first round attack efficiently in private cloud environment. The existing papers carry out a brief cache access pattern analysis and they do brute force numerous times to determine the AES encryption key. There is a need to study the cache hits and misses and relate the same to the required key.

The motivation behind the paper is two-fold. The first reason is that using the cloud infrastructure made available by the cloud providers increases the expenditure [11] Storing data on cloud is charged based on both the size of data and time for which data is stored. Being able to set up a private cloud using open source software reduces this expenditure and the cloud can be modified to provide features according to the requirements.

The second reason is that among the various side channel attacks, cache based side channel attack is considered to be the most dangerous [12]. This has been used by the attacker to harm the victim in several ways by extracting private data from the victim machine by analyzing the cache access patterns. The attacker reads the cache hits and misses to extract the encryption key. Once the key is determined, it can be used to read a lot of private data. Better understanding of an attack and how an attacker carries it out helps in coming up with better countermeasures for the attack. Understanding the attack leads to identification of loopholes in the whole setup and aligns the thought process for mitigation techniques in the right direction.

3. OUR ARRPOACH

3.1 Assumptions

- The plaintext that has been encrypted using AES algorithm is known to the attacker.
- The attacker knows where the victim’s lookup tables reside in memory.
- From the reduced group of affected cache sets, the attacker knows exact 16 cache sets affected by AES after permutation.

3.2 Design

- **Setting up Private Cloud**
  OpenStack is an open source software that is used for setting up Private cloud. This module forms the basis of the entire paper. The side channel attack is shown among the virtual machines created in this private cloud. The shared cache theory is based on the OpenStack architectural design. The services for the compute and controller nodes are enabled which are required for creating and running instances. The Compute and Controller nodes are configured and many services are enabled for the proper functioning of the cloud. The Virtual machines are created in the compute nodes based on the image available.

- **AES Encryption Algorithm**
  AES (Advanced Encryption Standard) is a symmetric key algorithm. The algorithm supports a key size of 128,192 or 256 bits. In the paper we use keys of size 128 bits. The round function is repeated fixed number of times usually 10 for key size of 128 bits. This is required to encrypt a plaintext of 128 bits to a cipher text of 128 bits. The 16 byte plaintext is represented as 4x4 array. Each round has four steps-Byte substitution, Row Shift, Column Mixing and a Round key operation. These operations are very expensive hence these are replaced by inexpensive lookup tables which increases the speed of encryption and decryption. There are four lookup tables say t0, t1, t2 and t3. In the function the initial 4 indices obtained by XORing the plaintext and the key is 4 bytes each. These are used in the first round of AES where 16 values of the lookup table corresponding to the 16 indices are accessed. From each lookup table maximum of 4 memory accesses are possible in the first round. These 16 memory accesses affect a maximum of 16 cache sets.

- **Cache Access Pattern Analysis**
  The attacker clears the cache and fills the whole cache by reading from contiguous array of size equal or more than the size of the cache. The number of clock cycles required to read the array is measured before and after filling the cache. The shared cache is affected when the victim runs the AES algorithm. Again when the array is re-read the clock cycle is measured which is now altered. Depending upon the variation in the clock cycles the cache access is analyzed. The initial number of 4096 cache sets is reduced based on the known lookup table addresses. The cache sets that are affected by running the AES algorithm is required for extraction of the key. All the cache sets which had a negative drop or which remained constant even after the victim runs the AES algorithm are rejected. Further, those which had a very small positive change in the clock cycles are rejected as the difference is less than what is responsible for cache miss. This is done multiple times to get accurate result.

- **Extraction of AES key**
  There is a very close relationship among the Lookup table indices and the AES encryption key. The non-accessed indices can be found from the non-accessed cache sets and this can be obtained by the cache pattern analysis. The key is of 16 bytes and each byte can take 256 values. In the method used, for each byte all those not possible values are rejected. This is done repeatedly using different plaintext and more values for each byte are rejected. The key can be then extracted by applying brute force to the remaining possible values. The cache is cleared and a contiguous array of size equal or greater than that of the cache is read by the attacker in the Prime stage. The attacker then waits in a busy loop for the victim to run its AES. Once the AES is executed there is a sudden increase in the clock cycle values for all the selected cache sets which shows the occurrence of the attack. Then using the indices values the key is extracted.

- **Mitigation**
  The cache based side channel attack helps the attacker to extract private information about the victim lying in the same physical machine. This reduces the security of the virtual machines created in the cloud environment. A mitigation measure can be taken to ensure security of the cloud environment. Clearing of the cache before running the AES by victim can ensure that no cache based side channel can be created. This can be further improved by randomly accessing the lookup tables which causes a confusion for the attacker to analyze the cache access pattern. The cache is cleared by doing mathematical operations which do not involve any memory accesses. Also in the AES algorithm, the lookup tables are randomly accessed to ensure there is no side channel created.

3.3 Experiment Result and Analysis

The side channel attack on Advanced Encryption Standard algorithm has been implemented on OpenStack platform, set up on Ubuntu 12.04 LTS. The virtual machine created by the attacker is made co resident to that of the victim's virtual machine to create a cache based side channel. This channel is used to extract the private key that has been used for encryption. The experimental analysis is based mainly on the cache pattern analysis and the extraction of the AES key.The evaluation of the implemented paper is based on the way the cache is accessed by the virtual machines, the manner of key extraction and its efficiency, and how effective the mitigation technique is.

For the attack to occur, the attacker creates the side channel for communication with the victim. The attacker then analyses the way in which cache is affected due to execution of AES by the victim. This analysis is based on different criteria which include; mapping of addresses of lookup table indices, constant or negative drop in clock cycles, and small positive change in clock cycles. Many samples are taken under each criteria. These samples are evaluated and checked for consistency, which is further used for key extraction.

The possible key set determined for a particular key can be further reduced by using different plaintexts. Smaller the possible key set, better will be the efficiency of evaluation as the time taken to brute force is reduced. For the victim to ensure security from this attack, certain countermeasures are employed. The efficiency of this technique is evaluated. Efficiency is better when fewer accesses to random indices of lookup tables provide successful mitigation.

For the attacker to extract the AES key from the victim, the cache sets affected by running AES should be known. Each lookup table has 256 elements which correspond to particular cache sets in the cache memory. From the initial group of all cache sets, the possible affected cache sets are determined based on mapping of the lookup table addresses as shown in fig 1. More than one element in the lookup table can map to the same cache set.

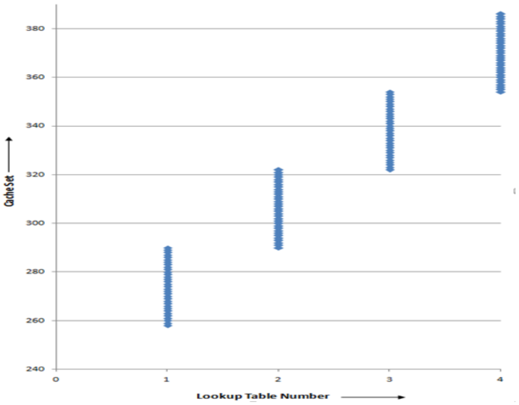

Figure 1 Mapping of lookup table addresses

The possible group of cache sets can be further reduced by analyzing the cache based on any constant or negative drop in the number of clock cycles as shown in fig 2. After the victim runs AES, the number of clock cycles taken by the attacker to access the cache sets affected by the AES is expected to increase. All those cache sets with the same or lesser number of clock cycles before and after running AES are the ones which have not been utilized by AES algorithm.

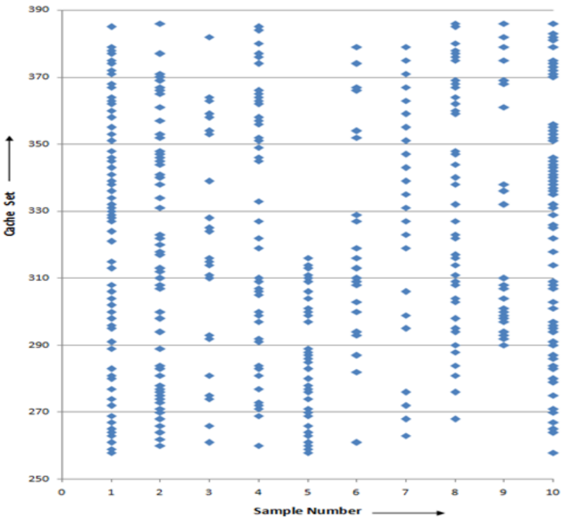

Figure 2 Constant or negative drop in clock cycles

When the victim runs AES algorithm, some cache sets are affected. Hence, when the attacker tries to access these cache sets, there is a cache miss. This will result in a difference of at least 250 clock cycles. All those cache sets which had difference less than 250 clock cycles were rejected as shown in fig 3.

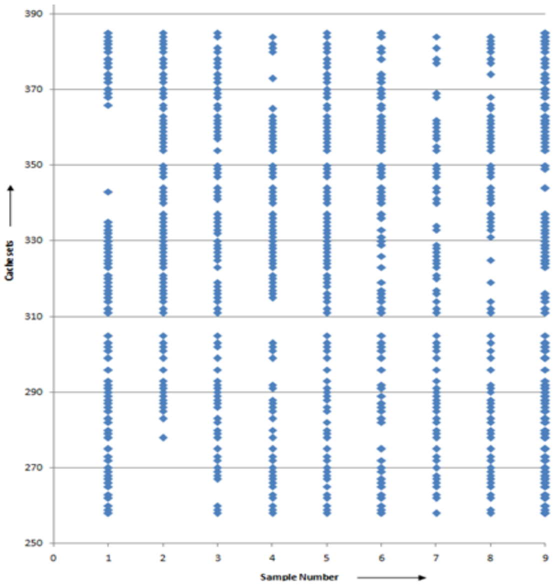

Figure 3 Small Positive Change in clock cycles

Once the group of non-accessed cache sets are known by the attacker, the corresponding non-accessed indices are found, these possible set of non-accessed indices are XORed with the plaintext to obtain the reduced possible set of values for each byte of the key. This is repeated for different plaintexts and the possible key set becomes smaller as shown in fig 4.

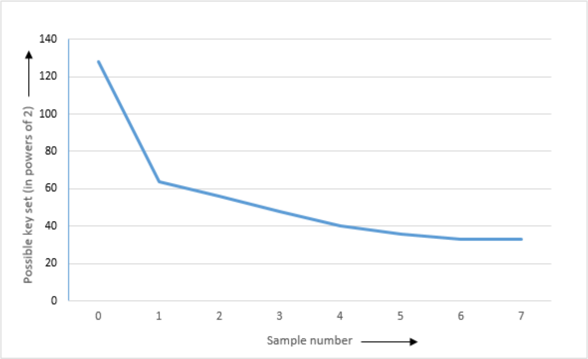

Figure 4 Reduction in possible key set

4. CONCLUSION

Unique security aspects of the Cloud have motivated this work. Primary among them is that the Cloud's architecture is particularly vulnerable to cache driven side channel attack. This paper is an earnest attempt to develop and implement novel techniques to prevent cache based side channel attack. In order to design a solution for counterattack, the details on how the attack is conducted to extract private information was required.

OpenStack, a private cloud operating system, is set up to host the system and its requirements. A robust access driven side channel attack on AES is implemented. The approach in carrying out cache pattern analysis is unique and it aims to reduce the possible key byte set to the maximum extent before carrying out brute force. The possible key sets are reduced from 2^{128} to 2^{33} effectively using few samples for analyzing cache patterns followed by a brute force technique to fully recover 128 bit AES key. Cache flushing and randomized lookup table's access avoids the creation of cache based channel thus providing a more secure cloud environment.

5. FUTURE WORK

The attacker VM should locate the physical host of the victim VM and place a new VM co-resident to the victim. The key extraction should be possible when the base address of the lookup tables is not known. The possible key set may be reduced further, before applying brute force method.

6. REFERENCES

[1] Xinjie, Z.; Tao, W.; Dong, M.; Yuanyuan, Z.; and Zhaoyang, L., "Robust first two rounds access driven cache timing attack on aes", *Proceeding of the 2008 International Conference on Computer Science and Software Engineering*, Washington, DC, USA, vol. 3, 2008, pp. 785-788.

[2] Godfrey, M.;Zulkernine, M., "A Server-Side Solution to Cache-Based Side-Channel Attacks in the Cloud",*Proceeding of the 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD),*June 28 2013-July 3 2013, pp.163-170.

[3]Tromer, E.; Osvik, D. A.; and Shamir.A., "Cache Attacks and Countermeasures: the Case of AES" in Topics in Cryptology – CT-RSA 2006, Springer Berlin Heidelberg, 2006, ISBN: 978-3-540-32648-9.

[4] Zhang, Y.; Juels, A.; Reiter, M. K.; and Ristenpart, T., "Cross-VM Side Channels and Their Use to Extract Private Keys", *Proceeding of the CCS'12*, Raleigh, North Carolina, USA, October 16–18, 2012.

[5] Ristenpart, T.; Tromer, E.; Shacham, H.; and Savage, S., "Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds", *Proceeding of the 16th ACM Conference on Computer and Communications Security*, Chicago, Illinois, USA, 2009, pp. 199–212.

[6] Gajrani, Jyoti; Mazumdar, Pooja; Sharma, Sampreet; Menezes, Bernard, "Challenges in Implementing Cache-Based Side Channel Attacks on Modern Processors",*Proceeding of the 27th International Conference*, 5-9 Jan. 2014, pp.222-227.

[7] Terrell, M.; Meghanathan, N., "Setting Up of a Cloud Cyber Infrastructure Using Xen Hypervisor",*Proceeding of the 2013 Tenth InternationalConference on Information Technology: New Generations (ITNG),* 15-17 April 2013, pp. 648-652.

[8]Xinyu Miao; Jing Han, "The Design of a Private Cloud Infrastructure Based on XEN", *Proceeding of the 2011 Tenth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES)*, 14-17 Oct. 2011, pp. 160-164.

[9] Fakhar, F.;Shibli, M.A., "Management of Symmetric Cryptographic Keys in cloud based environment", *Proceeding of the15th International Conference onAdvanced Communication Technology (ICACT),* 27-30 Jan. 2013, pp. 39-44.

[10] Bist, M.; Wariya, M.; Agarwal, A., "Comparing delta, open stack and Xen Cloud Platforms: A survey on open source IaaS", *Proceeding of the 2013 IEEE 3rd International onAdvance Computing Conference (IACC)*, 22-23 Feb. 2013, pp. 96-100.

[11] Brandwacht, L.Meeuwissen, E.Van den Berg, H.Ivkovic, "Models and Guidelines for Dimensioning Private Clouds", Proceeding of the 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD), June 28 2013-July 3 2013, pp.880-886.

[12] Harnik, D.Pinkas, B.Shulman-Peleg,"Side Channels in Cloud Services: Deduplication in Cloud Storage", IEEE Security & Privacy, vol.8, no.6, Nov-Dec.2010, pp.40-47.