International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 10, October- 2014

Cache-Based Side-Channel Attack on AES in Cloud Computing Environment

Sonali Tandon
Srushti S B
Vartika Agrawal
Dept. of Computer Science and Engineering
R V College of Engineering
Bangalore, India

***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. A Cloud Computing Environment to host the attack and prevent it is set up using an open source software called OpenStack. The AES algorithm implemented uses table lookup operations to access cache, and these lookup table indices are closely related to the AES key. Accordingly, a robust first round cache driven attack is launched on the victim virtual machine by an attacker. 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. Novel and efficient techniques to mitigate the attack are implemented. These include cache flushing followed by randomization of access to lookup table indices used in the AES encryption algorithm.**

Keywords*— *Cloud Computing; AES; Side Channel Attack

I. 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.

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 paper suggests two techniques for mitigating this attack by disrupting the cache access patterns during the encryption of the algorithm.

II. BACKGROUND STUDIES

Robust First Two Rounds Access Driven Cache Timing Attack on AES is discussed in [1] which uses a spy process to get the pattern of the cache accessed by an AES process. There is a misalignment of AES lookup tables over the data cache which is studied in detail and the accessed lookup table indices are then deduced. There is a close relation between the lookup table indices and the key so they are used effectively to extract the 128-bit key of AES. The experiments conducted have shown that 350 samples are required to extract the whole AES key when only the first round of AES is considered and if it is a second round attack then only 80 samples are required.

A Server-Side Solution to Cache-Based Side-Channel Attacks in the Cloud is discussed in [2] which analyses the side-channel vulnerabilities involving the CPU cache. The traditional defenses in a cloud environment has its own shortcomings hence a new mitigation technique is developed

which doesn't interfere with the Cloud model. Server side solution to this problem is to ensure that there is no overlapping of the cache among the probing and target instances which is the main requirement of the sequential cache-based side channels. This solution is implemented, validated and compared against the current state.

Cache Attacks and Countermeasures: the Case of AES is discussed in [3] which details 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. The paper also classifies the method used by the attacker to know about the memory access pattern of the other process. One method is to investigate the cache state and then analyse its effect on the execution time of the encryption process, the other method is to investigate the cache state after or during the encryption process. One of the variant of the attack described is that this is made possible even without the attacker knowing the plaintext or the ciphertext.

Cross-VM Side Channels and Their Use to Extract Private Keys is discussed in [4] which describes the side channel attack on a symmetric multiprocessing (SMP) system which is virtualized using XEN. The paper demonstrates the difficulty of a process to spy on a victim VM in a SMP environment and hence for the attack to happen the attacker should alternate the execution with the victim in the same core. Few other challenges were also detailed like core migration and noise in the extracted information.

Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds is discussed in [5] which demonstrates the need for the attacker to share the same physical infrastructure of the victim for the attack to take place. This is ensured by cloud cartography that is mapping of the internal cloud infrastructure. Network probing is done to check the co-residency of the VMs and to know the public services hosted. Once the co-residency is tested, placement of the attacker VM can be done either by brute forcing placement or abusive placement locality. The attacker measures the utilization of the CPU cache to extract the information using the Prime+Trigger+Probe technique.

Challenges in Implementing Cache-based Side Channel Attacks on Modern Processors is discussed in [6] which briefs about the AES implementation based on the look-up tables. The trivial Prime+Trigger+Probe technique is modified to overcome the prefetching which involves the principle of spatial locality that would transfer several words to the cache in a single access. The modified technique is used to find the location of the AES lookup tables. The list of all the non-accessed blocks are found using the Evict+Time method. The paper also analysed instruction caches and unified caches which are other cache features that affected the attack.

Setting up of a Cloud Cyber Infrastructure using Xen Hypervisor is discussed in [7] which explains the procedure to set up a cloud. The cloud set up uses the XEN hypervisor and is according to Platform as a service (PAAS) model. The hardware required to set up the cloud is specified followed by the configuration guide. According to the paper, servers required are added and the resource pool is created with a shared storage system. Virtual machines can be created by the users or a group of users created by the administrator using the virtual machine images available. These users have to be authenticated and this is done by the authentication services. Each virtual machines can be accessed by the users using the VNC remote desktop software.

The Design of a Private Cloud Infrastructure Based on XEN is discussed in [8] which analyses the benefits of a private cloud. It provides a technical overview of the XEN based virtualization and the two virtual domains. Virtualization can be full-virtualization and para-virtualization. Domain-0 and Domain-U are the two most important domains. XEN based architectural model and the infrastructure is explained in detail.

Management of Symmetric Cryptographic Keys in Cloud Based Environment is discussed in [9] which deals with managing of the secret cryptographic key in the cloud environment set up using OpenStack. An effective and robust secret protocol is implemented for managing the key. This technique is based on secret splitting which is based on Shamir's algorithm. A secured storage scheme is used for sensitive data in public/private/hybrid cloud. Symmetric cryptographic key is provided as a cloud service which the user can use in other utilities.

Comparing Delta, Open Stack and Xen Cloud Platforms: A Survey on Open Source IaaS is discussed in [10] which evaluates the differences between OpenStack and OpenNebula. OpenStack can offer same services as Amazon free of cost. One can acquire the resources dynamically and use services like storage, networking etc. Nova services can be used to create instances using the virtual machine images available and for networking. Cinder service can be used to create volumes which provides resources to the instances when attached to it. The OpenStack dashboard is a graphical User Interface (UI) for the users to manage their projects and instances. OpenNebula is similar but implementation is more complex and is more suitable for research institutes.

III. 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.

IV. DESIGN

A. 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.

B. *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.

C. *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.

D. *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.

E. *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 analyse 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.

V. EXPERIMENTAL 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 beenused 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.

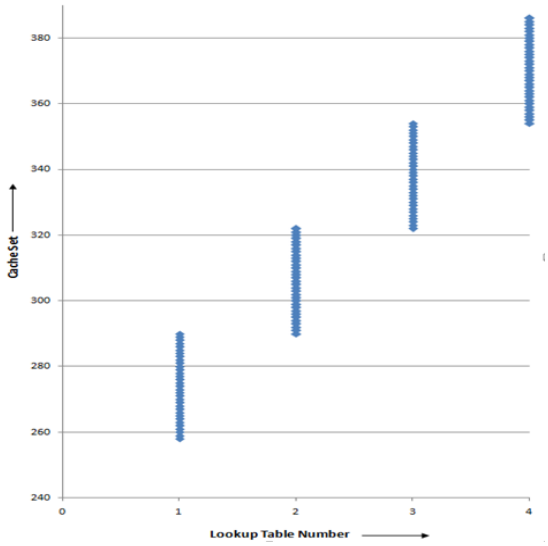

Fig. 1. Mapping of lookup table addresses

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.

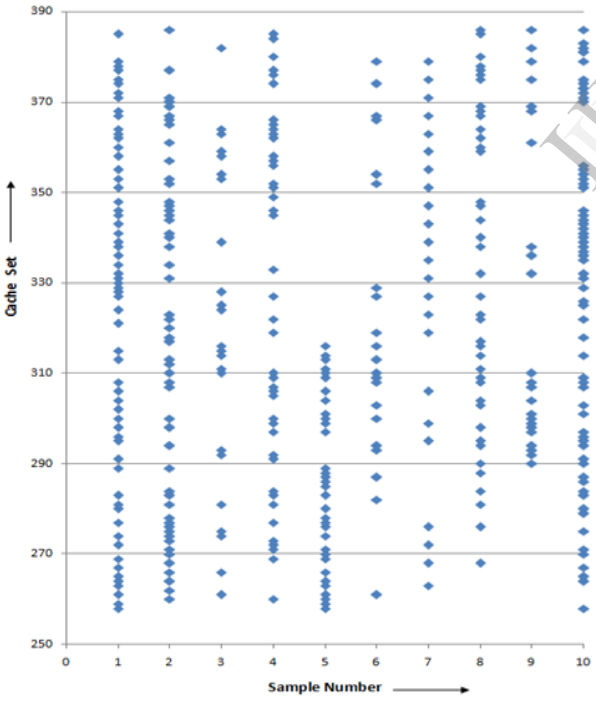

Fig. 2. Constant or negative drop in clock cycles

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

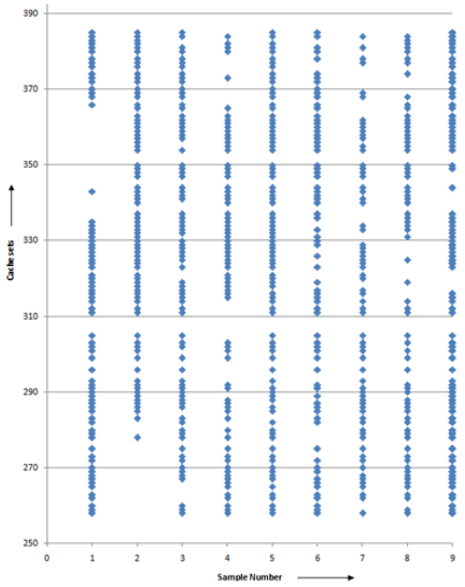

Fig. 3. Small Positive Change in clock cycles

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.

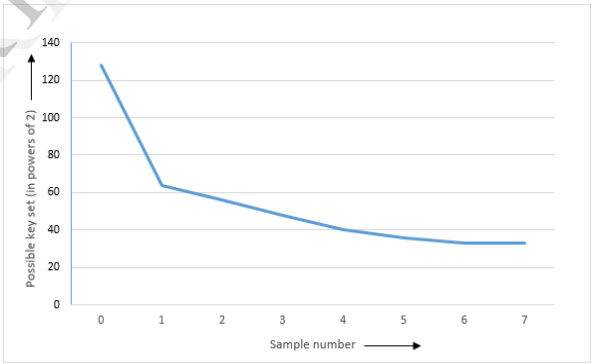

Fig. 4. Reduction in possible key set

VI. CONCLUSION

Unique security aspects of the Cloud have motivated this paper. 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 implementation is demonstrated. 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 analysing cache patterns followed by a brute force technique to fully recover 128 bit AES key. Cache

flushing and randomized lookup tables access avoids the creation of cache based channel thus providing a more secure cloud environment.

VII. 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.

REFERENCES

[1] Z. Xinjie, W. Tao, M. Dong, Z. Yuanyuan, and L. Zhaoyang, "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] M. Godfrey, M. Zulkernine, "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] E. Tromer, D. A. Osvik, and A.Shamir, "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] Y. Zhang, A. Juels, M.K. Reiter, and T. Ristenpart, "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] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "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] M. Terrell, N. Meghanathan, "Setting Up of a Cloud Cyber Infrastructure Using Xen Hypervisor," Proceeding of the 2013 Tenth International Conference 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, M.A Shibli, "Management of Symmetric Cryptographic Keys in cloud based environment," Advanced Communication Technology (ICACT), 2013 15th International Conference on 27-30 Jan. 2013, pp. 39-44.

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