ISSN: 2454 - 4248
16 – 22

Implementation of Multivariate Authentication Protocol (MAP) for Side Channel Attack Detection

Allwin D
Department of Computer Science, National College, Thiruvananthapuram -9, Kerala, India
allwindb@gmail.com, allwin_d@yahoo.com

G Suganthi
Department of Computer Science, Women's Christian College, Nagercoil-1, Tamil Nadu, India
dr_suganthi_wcc@yahoo.co.in

***Abstract*—** Cloud Computing offers an extensive variety of resources like computational power, computational storage and applications to clients by means of internet. Cloud Computing is empowering IT administrators to deliver resources to the users quicker in a great flexible way and at a cost effective model without having to restructuring or updating the basic infrastructure. With the expanding number of organizations falling back on utilize resources in the Cloud, there is a need for ensuring the security of the data of the clients using the cloud resources. The major challenged faced by cloud data centers to ensure security to its clients. According to the side channel attack the data privacy of the user is violated by observing the operation of the deduplication in the storage server of cloud, so this attack will easily allow the malicious user to access the data. The major contribution of this paper is to address the serious security issues related to side channel attacks. This paper proposes the design of a Multivariate Authentication Protocol (MAP) protocol against side channel attacks.

***Keywords*-** *MAP, Side Channel Attacks, Virtual Machine, DCNN*

*****

I. INTRODUCTION

In the Information Technology (IT) world the popular research area is Cloud computing and it is a successful technology adopted by companies to share the hardware resources efficiently to all their users and applications [1]. It will provide better scalability, accessibility and efficiency in cost but at the same time there are some security risks are also obtained [2]. In cloud computing systems, virtual machine is one of the basic component which expedite the utilization of hardware platforms to be more efficient and reduces the computing resources maintenance [3]. The hypervisor attack will access the information of the user, sharing resources and manipulate the VM illegally [4]. According to the side channel attack it will violate the data privacy of the user by observing the operation of the deduplication in the storage server of cloud, so this attack will easily allow the malicious user to access the data [5]. There are different kinds of methods are used to avoid these attacks such as game theoretical modelling, supervised learning method etc. [6], but the result is that by using these methods the side channel attack is not fully eliminated but it is reduced slightly.

II. COMMON TYPES OF SIDE CHANNEL ATTACKS

(a) Timing Attacks

A timing attack is a security concern that permits an attacker to find vulnerabilities in the security of a PC or on a network system by understanding to what extent it takes the framework to react to various inputs. Timing attributes will differ based on the encryption key in light of the fact that different systems take different time set to process different inputs. Factors incorporate performance optimizations, branching and conditional statements, processor instructions, RAM and cache hits. A timing attack takes into consideration how long it takes for a system to do something and use the statistical analysis to find the right decryption key and get access to the system. Timing assaults are additionally used to target devices, for example, smartcards and web servers that utilization OpenSSL. Web servers were accepted to be less powerless against timing assaults since system conditions could veil the differences in timing.

(b) Simple and Differential Power Analysis Attacks

Power Analysis is a type of side channel attack in which the attacker studies the power utilization of cryptographic equipment and devices, such as a smart card, tamper-resistant "black box", or integrated circuit. The assault can non-intrusively separate cryptographic keys and other confidential information from the device. Power Analysis includes outwardly deciphering power traces, or graphs of electrical movement after some time. Differential power examination (DPA) is a more propelled type of force investigation which can permit an attacker to process the intermediate values inside cryptographic calculations by factually breaking down information gathered from numerous cryptographic operations.

(c) Fault Attacks

In the smart card industry, fault attacks have been progressively contemplated since the distribution of Boneh, DeMillo and Lipton's paper [7] in 1996. There are numerous

ISSN: 2454 - 4248
16 – 21

methods for carrying out a fault assault. Fault assaults exploit the physical properties of devices. Hypothetical fault assaults depend on fault models, which in turn display physical conduct of attacked devices. Smartcard manufacturers have known about the risk of faults for quite a while, henceforth, they have built up a substantial amount of countermeasures. These countermeasures are generally particularly built for various methods for physical attacks. One noteworthy group is sensors and channels, which plan to identify attacks. Different countermeasures are to utilize excess, i.e., double rail rationale, where memory is multiplied, multiplied equipment, fit for figuring an outcome twice in parallel, or multiplied calculations, where a calculation is performed twice on similar equipment. On the off chance that two outcomes are computed, they are thought to be error free if both values tally with each other.

III. MULTIVARIATE AUTHENTICATION PROTOCOL (MAP)

The proposed malicious user via side channel attack detection scheme will be a combination of multivariate validation and information encryption. The new multivariate user authentication level verification model will ensure the security level hardening for the user to access or modify the sensitive data VMs present in the cloud datacenter. The confidentiality of the valid user will be achieved by using the configurable one-time authentication tokens exchange between the cloud controller and requested user. The idea behind this proposed protocol is to introduce and develop a security aware authentication, in which user's credential acts as a unique identity, which is utilized to check the validity of the user through secure hash algorithm.

(i) Registration Phase

The registration phase begins when a client C_i visits the registration authority RA_i invoked to obtain credentials form the clients such as user name, nationality and Date of Registration (DOR). The registration authority generates the unique-id key UK_i through the help of secure hash function , the client C_i his credentials are need to be converted into ASCII values for the purpose of secure hashing to generate unique-id of the client. The client's credentials are kept inside the cloud portal bounded by a hash function h_{fc}.

a) Computes, $UK_i = h_{fc}(ASCII_{DOR}, k)$ where UK_i a unique-id key of the client C_i, $ASCII_{DOB}$ is the ASCII conversion value of the client date of birth, k is the private cloud key of the system *by random number generator* (RNG), and h_{fc} is the hash function.

b) The registration authority RA_i additionally generate the fair password for the client C_i utilizing the secure hash function, the cloud system generated new strong password is given to the client C_i after completion of registration.

c) Computes, $PW_i = UK_i \oplus h_{fc}(ASCII_{UN} \oplus k)$, where $ASCII_{UN}$ is the ASCII conversion value of the client user name incorporated with private cloud system key $'k'$.

d) The above secure cloud system generated evidence information variables UK_i and PW_i are stored in the registration authority RA_i incorporated with the hash function h_{fc} for future authentications.

(ii) Login Phase

We extend our initial registration protocol into Login Protocol, which is going to be utilized throughout the authentication process. In this phase client C_i visits the authority Login page and input credential username and cloud system generated password through public internet, the client C_i inputted username alphanumeric values are automatically converted into ASCII value denoted as $ASCII_{UN}$. The credentials are authenticated from the public cloud service provider CSP_i.

a) Computes, $LV_i = [ASCII_{UN}, PW_i]$, validates whether LV_i equals to the stored credential in the public cloud service provider of the client C_i .If LV_i equals to store RA_i and also validate the IP address of public cloud service provider, the authority login page computes further, otherwise, terminate the session.

b) Computes $OP_i = h_{fc}(PW_i \oplus T_c)$, where, OP_i is the one-time password computing variable and T_c is the current timestamp.

c) The public cloud service provider CSP_i sends one time password OTP through RA_i to the client C_i for further authentication over a calculated timestamp. Computes $S_m = [IP_i, OP_i, T_c]$. The

ISSN: 2454 - 4248
16 – 21

client C_i input the correct OTP and sends the message to the public cloud, here the network channel is insecure.

d) Computes $(T_s - T_c) > \Delta T_c$, the public cloud service provider CSP_i compare the time stamp T_c format, in case of incorrect format, the cloud rejects the login request. If the time stamp T_c minus the current time stamp of the cloud server T_s is greater than the expected time interval ΔT_c of the system than the system also rejects the login request.

e) If the current time stamp is in a valid format and under the required time interval then the system grants access to the client C_i to access and utilize the cloud application.

(iii) Password reset phase

The password-reset phase is the second general protocol of our system. This phase starts when client C_i request a new password. The login procedure is the same as discussed above in the Login Phase. The client C_i access his smart application and input his credentials including username and password. After successful login authentication from the public cloud the system asks the user to input his new password and the user device performs the following.

a) Computes $LV_i = [ASCII_{UN}, PW_i]$, where, LV_i is the login computing variable.

b) Verifies LV_i is equal to the stored PW_i or not. If the login is successful the authority login page computes further, otherwise, terminates the session.

c) Computes $NP_i = PW_n \oplus h_{fc}(ASCII_{UN} \oplus k)$, where, NP_i is the new password computing variable, PW_n is the new password. After successful authentication, the cloud generates the new password incorporated with the hash chain and store it to the cloud database.

d) The system then stores and replaces the new password PW_n with the old one PW_i and terminates the phase successful.

(iv) DCNN based VM Risk Classification

With late progressing of Internet of Things (IoTs), it turns out to be exceptionally alluring to actualize the profound Deep Convolutional Neural systems (DCNNs) onto implanted/convenient frameworks. By and by, executing the product based DCNNs (Figure 3.1) requires elite server bunches by and by, limiting their across the board organization on the cell phones. To conquer this issue, extensive research endeavors have been led with regards to growing exceedingly parallel and particular DCNN equipment, using GPGPUs, FPGAs, and ASICs. Stochastic Computing (SC), which utilizes bit-stream to speak to a number inside [-1, 1] by including the quantity of ones the bit-stream, has a high potential for executing DCNNs with high adaptability and ultra-low equipment impression.

Specifically, in each time period, the Attacker Monitor utilizes a semi supervised learning module construct via multiple DCNN, which classify all VMs into three categories high risk (malicious), medium risk (indeterminate) and low risk (permissible) and modify the VM allocation process accordingly.

Finally, to eliminate the side channels, once the attacker monitor detects that one co-tenant VM has abnormal behaviour accurately when the cloud controller suddenly executes crypto signature algorithm, it will provide notification for side channel attacks and also migrate this malicious VM to a different processor socket or alternative cloud server to destruct the side channels. Further the cloud controller will report this incident to the cloud service provider for further actions, such as shut down the malicious VM or permanently block the attacker's account. The proposed model implemented in cloudsim platform and the performance will be evaluated with existing models with respect to detection accuracy and latency.

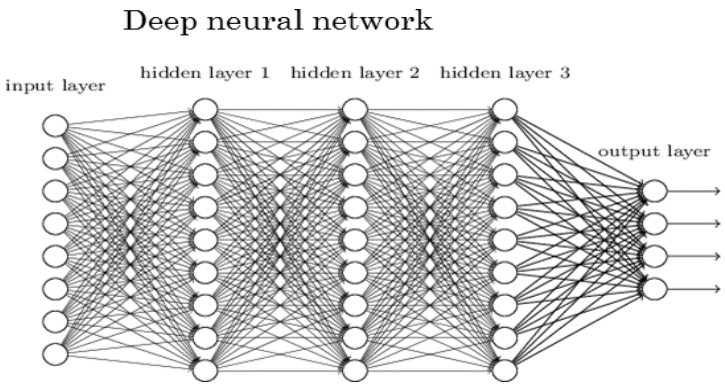

Figure 3.1 DCNN

(v) Virtual Machine

In figure 3.2, a virtual machine (VM) is an imitating of a PC framework. Virtual machines depend on PC structures and give usefulness of a physical PC. Their executions may include specific equipment, programming, or a blend. There are various types of virtual machines, each with various capacities.

ISSN: 2454 - 4248
16 – 21

Framework virtual machines (likewise named full virtualization VMs) give a substitute to a genuine machine. They give usefulness expected to execute whole working frameworks. A hypervisor utilizes local execution to share and oversee equipment, taking into consideration different situations which are detached from each other, yet exist on the same physical machine. Current hypervisors utilize equipment helped virtualization, virtualization-particular equipment, essentially from the host CPUs.

Handle virtual machines are intended to execute PC programs in a stage autonomous environment

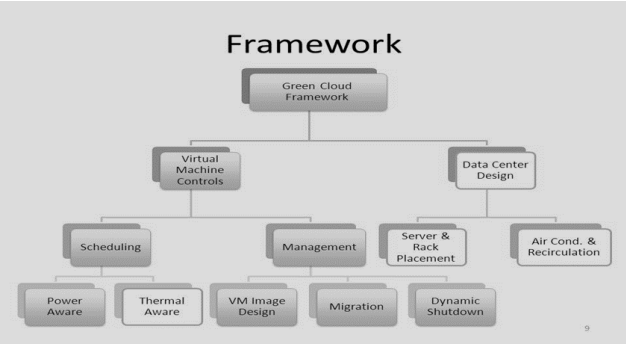

Figure 3.2 Virtualmachine framework

(vi) VM Classification

Distributed Computing is set of assets and administrations offered through the Internet. Cloud administrations are conveyed from server farms situated all through the world. Distributed computing encourages its shoppers by giving virtual assets by means of web. General case of cloud administrations is Google applications, gave by Google and Microsoft SharePoint. The quick development in field of "distributed computing" likewise increments serious security concerns. Security has remained a steady issue for Open Systems and web, when we are discussing security cloud truly endures. Absence of security is the main obstacle in wide selection of distributed computing. Distributed computing is encompassed by numerous security issues like securing information, and looking at the usage of cloud by the distributed computing sellers. The wide acknowledgment www has raised security hazards alongside the uncountable advantages, so is the situation with distributed computing. The blast in distributed computing has brought bunches of security difficulties for the buyers and specialist organizations. How the end clients of distributed computing realize that their data is not having any accessibility and security issues? Each one postures, Is their data secure? This review plans to distinguish the most powerless security dangers in distributed computing, which will empower both end clients and merchants to think about the key security dangers connected with distributed computing.

We calculated the following four metrics regarding the usage of VM. Each of these metrics corresponds to one of the features mentioned above:

1. *N* − the total number of VMs started (the larger the square is, the more VMs the account starts);
2. *T* − the average time interval between starting two VMs. Note that here the time interval means the time difference between starting the *i th* and the (*i* +1)*th* VMs, rather than stopping the *i th* and starting the (*i* + 1)*th* VMs;
3. *PA* − active percentage, which is equal to the proportion of time during which at least one VM is running, i.e., the active time, divided by the time since the account created its first VM;
4. *LEN*− the third quartile of the VMs' running time (note that we also tested the median value, and found that it was not effective at differentiating users – the median values for a large percentage of users were quite close).

5. Algorithm : DCNN based VM Classification
6. Initialize Position
7. While (number of iterations, or the stopping criteria is not met)
8. For p = 1 to number of locations
9. Divide training set data and testing set data
10. $$f_{train} \leftarrow 1/\left(1+\exp\left[-\sum_j^{n_h} \phi_{ji} v_{train,j} + p_i\right]\right)$$
11. $$f_{test} \leftarrow 1/\left(1+\exp\left[-\sum_j^{n_h} \phi_{ji} v_{test,j} + p_i\right]\right)$$
12. Initialize Super Parameter β
13. $$k(x_i, y_j) = \exp^{-r\|x_i - y_j\| X_p}$$
14. While (number of iterations, or the stopping criterion is not met)
15. For $i = 1$ to number of training data
16. $$z_i = \sum_{j=1}^{n} \beta_j y_j k(x_i, x_j)$$
17. $$\delta\beta_i = \eta(1 - z_i y_i)$$
18. $$If\ (\beta_i + \delta\beta_i) \le 0 \quad then\ \ \beta_i = 0$$
19. $$If\ (\beta_i + \delta\beta_i) > C \quad then\ \ \beta_i = C$$
20. $$If\ (\beta_i + \delta\beta_i) > 0 \quad then\ \ \beta_i = (\beta_i + \delta\beta_i)$$
21. Next i
22. Next iteration until criterion

ISSN: 2454 - 4248
16 – 21

23. For $i = 1$ to number of testing data

$$z_i = \sum_{j=1}^{n} \beta_j y_j k(x_i, x_j)$$

24.

25.

$$if\ z_i > 0\ \ then\ \ class_i = +1\ \ else\ \ class_i = -1$$

26.

$$if\ class_i = real\ class\ of\ testing\ data\ \ then$$

27. $$right = right + 1$$

28. Next i

29.

$$fitness_p = right / number\ of\ testing\ data$$

30. If the fitness of X_p is greater than the fitness of $x_{i,new}$

31. then Update $x_{i,new} = X_p$

32. For $k \in$ Neighborhood of X_p

33. If the fitness of X_k is greater than that of x_i then

34. Update $x_i = X_k$

35. Next k

36. Next generation until stopping criterion.

For malicious node classification, if $y_i = +1$ (i.e., the value of i^{th} training will be positive), then

$$\frac{\partial k}{\partial \varphi_j} = if \sum_{j=1}^{n} \phi_j x_i^j + c \geq 1\ then\ 0\ \ else\ \ -x_i^j$$

Moreover, if $y_i = 1$ (i.e., the value of i^{th} training will be negative),

$$\text{the } \frac{\partial k}{\partial \phi_j} = if\ \ \sum_{j=1}^{n} \phi_j x_i^j + c \leq -1\ then\ 0\ else\ -x_i^i$$

The two cases can be outlined as one, it embrace the value of y_i, as

$$\frac{\partial k}{\partial \phi_j} = if\ \ y_i\left(\sum_{j=1}^{n} \phi_j x_i^j + c \geq 1\right)\ then\ 0\ \ else\ \ -y_i x_i^j$$

IV. RELATED WORK

David Brumley et al. proposed Timing assaults are normally used to assault frail registering gadgets, for example, smartcards, to demonstrate that planning assaults apply to general programming frameworks. Specifically, devise a planning assault against OpenSSL. The tests demonstrate that extraction private keys from an OpenSSL-construct web server running with respect to a machine in the neighborhood network. The comes about show that planning assaults against system servers are viable and along these lines security frameworks ought to safeguard against them [8].

Markus Jacobson proposed [9] acquaint devices with model and portray phishing assaults, permitting a representation and measurement of the danger on a given complex arrangement of web services, use new model to depict some new phishing assaults, some of which have a place with another class of mishan dle presented in this: the setting mindful phishing attacks, to depict methods for utilizing the model present to measure the dangers of an assault by method for monetary examination, and strategies for shielding against the assaults depicted.

Peter Gutmann proposed method for defenseless against side channel assaults due to its strict prerequisites for outright mystery. In the product world, side channel assaults have here and there been expelled as illogical. Notwithstanding, new framework engineering elements, for example, bigger store sizes and multicore processors, have expanded the pervasiveness of side channels and nature of estimation accessible to an assailant. Programming engineers must know of the potential for side channel assaults what's more, arrangement suitably [10].

Werner Schindler proposed Timing assaults speak to a genuine risk which must be considered while executing open key cryptosystems The fundamental thought of any planning assault is to decide a mystery parameter from differences in running circumstances required for different info values they treat two variations of timing assaults presented in and debilitate assumptions indicate the separate scientific models what's more, improve these assaults for this apply factual choice hypothesis [11].

Mark Weiser proposed Pervasive figuring is the strategy for upgrading PC use by making numerous PCs accessible all through the physical environment, yet making them viably imperceptible to the client. Since this work at Xerox PARC in 1988, various specialists around the globe have started to work in the universal processing system. This clarifies what is new and distinctive about the PC science in omnipresent registering. It begins with a brief diagram of universal registering, and after that expounds through a progression of cases drawn from different sub disciplines of software

ISSN: 2454 - 4248
16 – 21

engineering: equipment parts (e.g. chips), organize conventions, association substrates (e.g. programming for screens also, pens), applications, protection, and computational techniques. Universal registering offers a system for new and energizing examination over the range of software engineering [12].

V. RESULTS AND ANALYSIS

The results of our proposed methodology is taken by varying the amount of file size being selected and encrypted from the VMs. The performance measures used in our paper for the evaluation of its efficiency are the Accuracy (Figure 5.1), Avalanche Effect (Figure 5.3) and Entropy factor (Figure 5.2). The obtained results are given in the form of graphs.

- Number of servers 10
- Each server has 10 VMs. Totally 100 VMs are taken.
- Each VM having the different file sizes

(i) Accuracy

Accuracy of the result provided by each virtual machines.

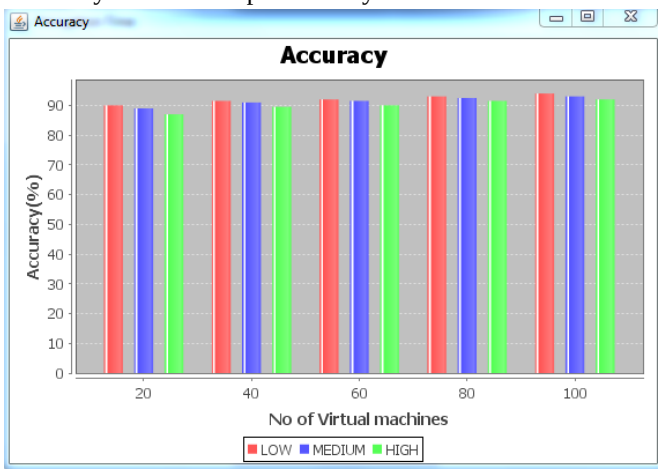

Figure 5.1 Accuracy

(ii) Entropy factor and Avalanche effect

There are two measures are used to detect the impact of single bit value on the whole encrypted text. We analyses the impact of single bit change in the plain text to that of the cipher text by the following measures. The measures are (a) Avalanche effect (b) Information Entropy. The following figures gives the result of above parameters for the different file sizes.

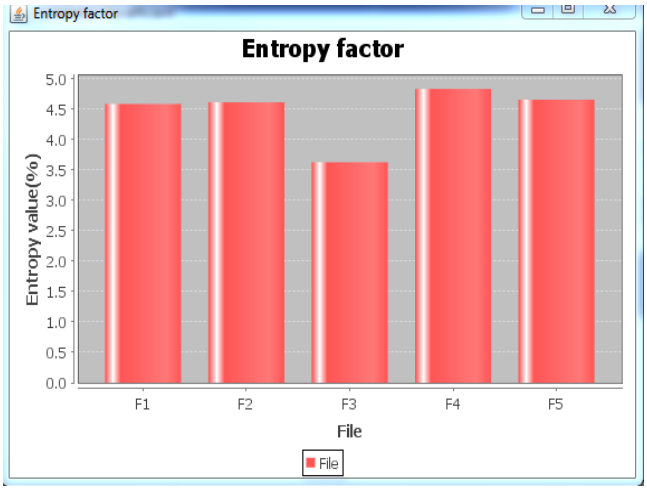

Figure 5.2 Entropy factor

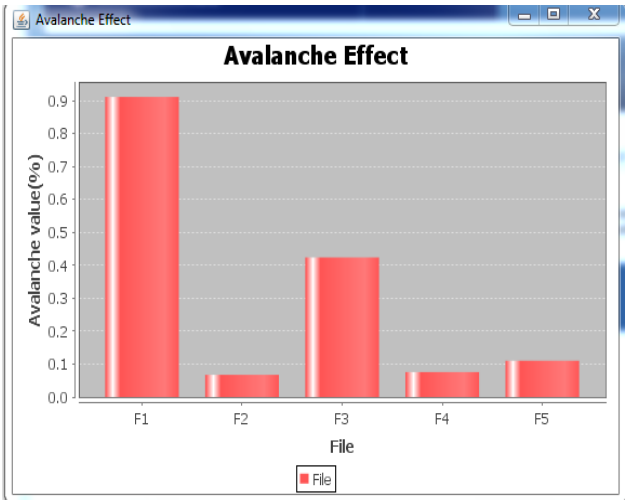

Figure 5.3 Avalanche effect

VI CONCLUSION

Side-channel attacks are hazardous in light of the fact that it can break systems that are thought to be secure through other security systems. The Cloud's architecture is particularly susceptible to side-channel attacks. Such attacks in the Cloud cannot be solved by conventional means without interfering with the Cloud model. To address these problems, we have developed and implemented a DCNN based VM Risk Classification model. The proposed models give better results when compared to the conventional methods for preventing side channel attacks.

REFERENCES

[1] Han, Yi, Jeffrey Chan, TansuAlpcan, and Christopher Leckie, "Using virtual machine allocation policies to defend against co-resident attacks in cloud computing." (2015).

[2] Han, Yi, TansuAlpcan, Jeffrey Chan, and Christopher Leckie, "Security games for virtual machine allocation in cloud computing.", Springer International Publishing, pp. 99-118, 2013.

ISSN: 2454 - 4248
16 – 21

[3] Han, Yi, TansuAlpcan, Jeffrey Chan, Christopher Leckie, and Benjamin IP Rubinstein, "A game theoretical approach to defend against co-resident attacks in cloud computing: Preventing co-residence using semi-supervised learning." IEEE Transactions on Information Forensics and Security, Vol. 11, No. 3, pp: 556-570, 2016.

[4] Tep, Kin Suntana, Ben Martini, Ray Hunt, and Kim-Kwang Raymond Choo. "A Taxonomy of Cloud Attack Consequences and Mitigation Strategies: The Role of Access Control and Privileged Access Management." In Trustcom/BigDataSE/ISPA, 2015 IEEE, vol. 1, pp. 1073-1080. IEEE, 2015.

[5] Sadique, UK Muhammed, and Divya James, "A Novel Approach to Prevent Cache-based Side-Channel Attack in the Cloud." Procedia Technology, Vol. 25, pp: 232-239, 2016.

[6] Zhang, Rui, Wen Qi, and Jianping Wang. "Cross-VM Covert Channel Risk Assessment for Cloud Computing: An Automated Capacity Profiler." In 2014 IEEE 22nd International Conference on Network Protocols, pp. 25-36.IEEE, 2014.

[7] D. Boneh, R.A. DeMillo, and R.J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In W. Fumy, editor, Advances in Cryptology – EUROCRYPT '97, volume 1233 of LNCS, pages 37–51. Springer-Verlag, 1997.

[8] David Brumley and Dan Boneh,Remote timing attacks are practical, (2005) pp 701–716 in crypto.stanford.

[9] Markus G. Kuhn," Eavesdropping attacks on computer displays", Research Gate 2009.

[10] Joseph Bonneau," Robust Final-Round Cache-Trace Attacks Against AES", International association for cryptographic research 2006.

[11] Werner Schindler," On the Optimization of Side-Channel Attacks by Advanced Stochastic Methods" International Workshop on Public Key Cryptography,pp-83-105,2005.

[12] Miroslav Kne_zevi_c, Ventzislav Nikov, and Peter Rombouts," Low-Latency Encryption Is \Lightweight = Light + Wait"??" International Workshop on Cryptographic Hardware and Embedded Systems, pp 426-446,2012.