International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Cross-VM Branch Prediction Analysis Attack: Scope Assessment and Simulation

Dhara H. Buch, Haresh S. Bhatt

Abstract*: Growing scope of cloud computing has made cloud security a challenging parameter. Among all the security parameters, virtualization security requires primary focus as it hides internal resource sharing details of the system. Side Channel Attack (SCA) is an attack that exploits the shared resource for extracting the private key of a cryptographic algorithm. Considering the significance of virtualization security, we need to analyze the SCA in a virtualization environment. In this paper, we target the Branch Prediction Analysis (BPA) Attack, one type of SCA. We have carried out an analysis to verify the scope of various BPA attack launching methods in virtualization environment along with the simulation. We have also analyzed the scope of existing solutions handling BPA attack.

***Index Terms*: Virtualization, Side Channel Attack, Branch Prediction Analysis Attack, Branch Target Buffer.**

I. INTRODUCTION

Emerging requirements of fast, accurate and efficient processing with tight security and authenticity has given rise to high-performance computing environments. Distributed systems like cluster, grid and cloud computing environments provide a large, scalable and robust operating platform. Such distributed systems are comprised of physically scattered but connected and integrated set of computing resources. Among them, on the fly and pay per use basis service offering by cloud computing has widened its area of application. At the same time, it becomes utmost important to analyze various cloud security issues for providing reliable services to cloud customers. Among different security issues, virtualization security is a major security concern as it forms the backbone of the cloud computing platform.

Isolation is a very important property of virtualization which prevents Virtual Machines (VMs) from accessing private resources of the other Virtual Machine. However, the shared resources open the door of many security breaches where one of the issues is the covert channel. By illicitly transferring information via shared resources, i.e., covert channel, an attacker can easily break the security policy of a system. In a virtualized environment, co-resident VMs share physical resources like CPU cache, a memory bus, disk bus, Branch Target Buffer (BTB) and Network Queue. The covert channel takes advantage of these shared resources to leak confidential information and gives rise to Side Channel Attack (SCA). [1]

Side Channel Attack is typically used to extract private key bits of cryptographic algorithms merely by measuring performance parameters. Attack launching methodology of SCA makes it very difficult to detect its presence in the system, which makes mitigation of SCA a very important parameter to consider. Although Side Channel Attack can be launched to exploit any of the shared resources, the majority of the research work has been carried out on exploitation of cache memory. However, Branch Prediction Analysis Attack is one type of SCA that also requires equal focus from the security perspective. Ounr et al. [2] have proposed Branch Prediction Analysis (BPA) attack where components of Branch Prediction Unit (BPU) like branch predictor or Branch Target Buffer (BTB) is exploited to extract private key of the asymmetric cryptographic algorithms like Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC). Onur suggested four different methodologies to launch BPA attack where each of the methods follows different procedures. In [2], Onur et al. have stated about the possibility of BPA attack in the presence of virtualization and sandboxing. However, we need to carry out a detailed analysis of how the attack can be carried out in a virtualization environment considering various underlying system configuration. In this paper, we give a brief overview of four BPA launching methods in section II. In section III, we analyze the scope of Cross-Virtual Machine (Cross-VM) BPA attack. Section IV discusses the existing work that also includes scope analysis of present solutions for handling BPA attack in virtualization. Simulation results are discussed in section V. Finally, a conclusion is provided in section VI.

II. BRANCH PREDICTION ANALYSIS ATTACK: AN OVERVIEW

BPA attack suggested by Onur [2] targets conditional branch instruction (line number 5 in Figure 1) of the Square & Multiply (S&M) algorithm used in asymmetric cryptographic algorithms like RSA and ECC. The execution of this instruction depends on the status of the key-bit. Hence, the execution flow of such algorithms is very much key-dependent and it also affects the execution time of each iteration. In the following subsections, we briefly discuss different attack methodologies to launch BPA attack.

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Dhara Buch*, Department of Computer Engineering, Government Engineering College, Rajkot, Gujarat, 360005 India.
Haresh Bhatt, Space Application Center, ISRO, Ahmedabad, 380015, India.

```
Input:
C: Ciphertest
d: Private Key
n: Module
SqandMul (C, d,n)
1  M' ← 1
2  For i=1 to keylength
3     M'=MontMul(M',M',n);
4      If di = 1 then
5            M' = MontMul(M',C,n)
       End
   End
Return M'
```

Figure 1 Square & Multiply Algorithm

A. *Direct Timing Attack*

In DTA procedure, a large (around 1000) set of known ciphertext messages are chosen for simulation. There are mainly two phases, Offline and Online. In the Offline phase, the underlying cryptographic algorithm (RSA or ECC) is called for decryption for each of the message in the set. A trace is generated for the execution status of conditional branch instruction as taken or not-taken for the first i bits of the cryptographic algorithm, where those i bits are assumed be known. Further in the online phase, the generated trace of taken and not taken branches for a message is fed to the dynamic predictor for predicting the next unknown bit. Status of conditional branch instruction is predicted from the Branch Predictor from the generated trace for both possible bit values: 0 and 1. The actual execution status of the conditional branch instruction is also observed for the same two possibilities of the next bit. The output of the branch predictor is compared with the status observed from the actual execution. Result of comparison maps in either of the four cases depending on whether the actual behavior matches with the prediction of predictor or not for both the assumed values 0 and 1. The procedure is repeated for each message in the message set. If the four sets are named as M1, M2, M3 and M4, then an average of mispredicted branches of all the messages in a set is calculated. Let we denote the average as Avrg(MB(Mi)) for i=1 to 4. Finally, prediction of the next unknown bit is carried out from the following equation.

$$\begin{aligned} &if\ Avrg(MB(M_1)) > Avrg(MB(M_2)\ \text{and} \\ &\quad Avrg\big(MB(M_3)\big) < Avrg(MB(M_4))\ \text{then} \\ &\qquad next\_unknown\_bit = 1 \\ &else \\ &\qquad next\_unknown\_bit = 0 \end{aligned} \tag{1}$$

The above process is repeated until all the unknown bits are extracted.

B. *Asynchronous Attack*

Launching of Asynchronous attack is carried out by executing a dummy process that continuously clears BTB entries. Whenever the parallelly running cryptographic process requires to execute the target branch instruction, the required target address would not be present in BTB and the branch miss event is generated. The BPU will predict the branch as not taken because of the branch miss event. As the dummy process continuously clears BTB, every time when the key bit is 1, the misprediction event would be generated. Adversary can predict the key bit value by monitoring the branch misprediction event.

Clearance of BTB takes either of the three forms:

1. Total Eviction : Entire BTB is cleared
2. Partial Eviction : Part of BTB storing target address of target branch instruction is evicted
3. Single Eviction : Single BTB entry storing the target address of target branch is evicted

Total eviction is easy to implement, while the other two eviction methods can improve the success ratio of attack. Dummy process can predict the secret bits without any need of being synchronous to the cipher process and so the attack is known as Asynchronous attack.[2]

C. *Synchronous Attack*

Synchronous attack requires a synchronism between the dummy process clearing BTB entries and the cryptographic process. If the dummy process can keep pace with the crypto process by establishing synchronism, it clears the BTB just before the execution of the conditional branch instruction of S&M algorithm. For bit 1, the branch would be taken and an event of misprediction is generated. Accordingly, by clearing single BTB entry before conditional instruction of ith iteration, predicts di and generates the entire key accordingly.[2]

D. *Time Driven Attack*

In Time Driven Attack the spy process is a comprised of large number of conditional branch instructions. Total number of conditional instructions are selected to full the entire BTB or to full the BTB set where the victim crypto process is executing. The spy process starts executing before the victim crypto process executes. The spy process continuously executes and fills the target BTB entries. Hence, when the conditional branch instruction of RSA algorithm executes, corresponding target address would not be available in BTB and would be required to be fetched from memory. At the same time, the target address of target branch is entered in BTB for which one of the BTB entry is evicted. The evicted BTB entry belongs to the spy process.

Spy process continuously measures its execution time, so when one of its branch target address is thrown out, it finds considerable time difference.

Hence, when conditional branch instruction of RSA algorithms gets executed, it is reflected in the execution time of spy process and accordingly the presence of bit1 in the secret key is predicted by the attacker. [2][3] As discusses in previous sub-sections, BPA attack launching is carried out by different methods where in the first method i.e. DTA, spy process works independently from the victim process. For the last three methods, the spy process needs to observe the execution flow of the target process. Hence, both the spy and victim processes are required to run concurrently unlike the first method. We need to check whether these methods can work in virtualization environment considering the configuration of VMs. Related discussion is carried out in the next section.

III. CROSS-VM BRANCH PREDICTION ANALYSIS ATTACK

As per the original contribution in [2], BPA attack is launched by a spy process loaded on the same machine where the cryptographic algorithm is getting executed. An attacker can easily load a spy process on a machine running a cryptographic algorithm by exploiting any of the other services hosted on that machine. However, a VM in a virtualization environment does not host multiple services. Hence, it is not easy for the attacker to launch a spy process on a different VM which executes cryptographic library. Spy process on one VM should be able to launch a BPA attack on the other co-resident VM to make the attack launching possible in a virtualization environment. Applicability of such Cross-VM BPA attack also lies on the underlying hardware configuration as well as software sharing policies. Cryptographic library and BTB are the primary elements in all the four attack launching methods. In this section, we perform scope analysis of the four attack launching methods concerning the sharing configuration of library and BTB (i.e., COU core).

A. Cross-VM BPA Attack with Direct Timing Method

We consider an attack scenario in a virtualization environment, where VM1 and VM2 are co-resident VMs. The cryptographic library is installed on a shared VM and VM1 is accessing the RSA cryptographic library from there. With an assumption that VM2 manages to trap encrypted private key of VM1, VM2 launches Direct Timing Attack to extract the private key of RSA. The attack scenario is shown in Figure 2. As per the procedure of DTA, attack launching needs VM1 and VM2 to access the same cryptographic library. However, an isolated CPU core of both the VMs do not prevent the launching of DTA attack. Hence, we can state that Cross-VM DTA can be launched in the presence of shared cryptographic library irrespective of the sharing configuration of CPU cores.

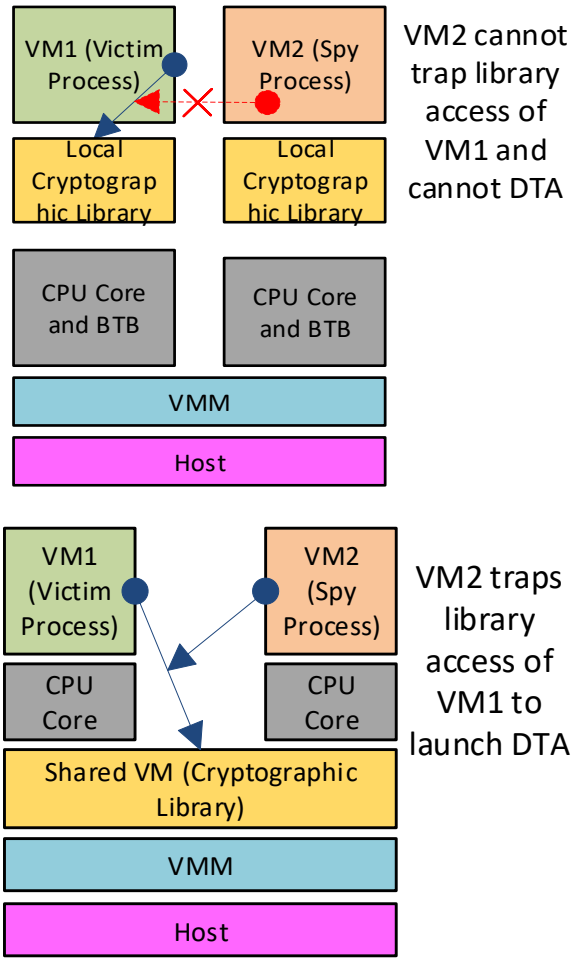

Figure 2 Cross-VM Direct Timing Attack – Attack Scenario

B. Cross-VM BPA Attack with Asynchronous Method

As discussed in section II, Asynchronous BPA attack is launched by the spy process that continuously clears the entire or part of BTB. BTB entries occupied by the target cryptographic algorithm is the main focus of the spy process. However, the spy process on one VM can access BTB of the victim process located on separate VM, provided both the VMs are sharing the CPU core. Thus, the Cross-VM Asynchronous attack can take place only among the VMs operated by the common CPU core irrespective of the sharing configuration of cryptographic library.

C. Cross-VM BPA Attack with Asynchronous Method

Only difference between the attack launching procedures of Asynchronous and Synchronous attack is that the spy process in Synchronous attack needs to keep pace with the victim process. It becomes possible for the spy process to clear BTB entries just before the conditional branch instruction only in the presence of common BTB. However, it is quite difficult to implement it in Cross-VM environment. Like Asynchronous attack, sharing configuration of the cryptographic library does not affect the success of BPA attack.

D. *Cross-VM BPA Attack with Time Driven Method*

Time Driven Attack works in a way which is similar to the Asynchronous attack with only one difference. In TDA, the spy process fills BTB entries to observe the execution time, unlike Asynchronous attack where BTB is cleared for the same purpose. However, both the attack procedure needs a common CPU core (and so common BTB) between the attacker and victim VMs. Attack scenario of Cross-VM BTB Attacks (Asynchronous, Synchronous and TDA) is presented in Figure 3.

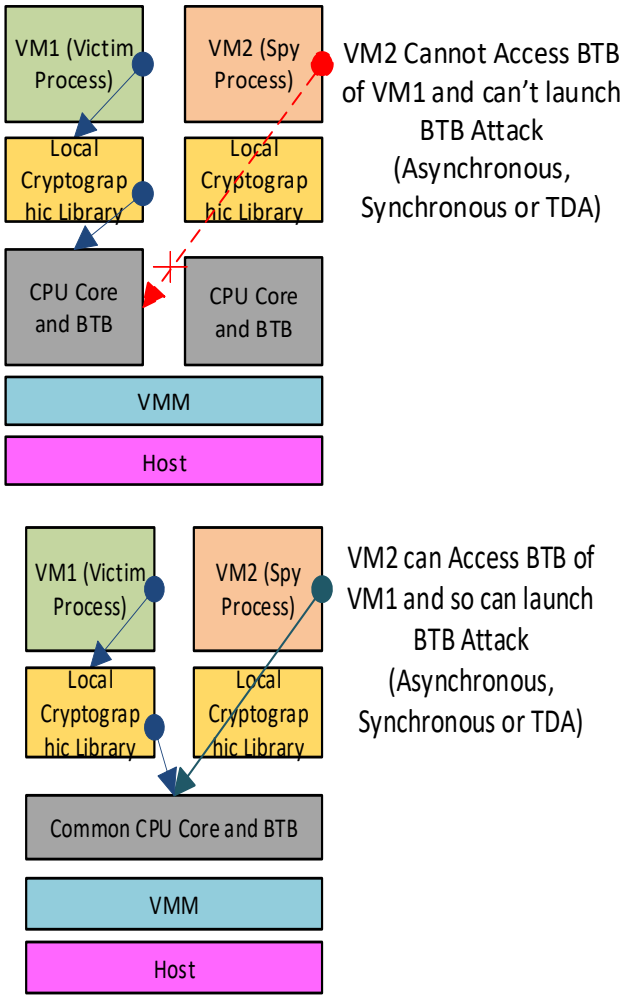

Figure 3 Cross-VM BTB Attacks – Attack Scenario

Summary of the above discussion is carried out in table I.

IV. RELATED WORK

Proposal of BPA attack by Onur et al. [2] was followed by a solution to eliminate the scope of the attack [4] where replacement of S&M algorithm was suggested by Montgomery Ladder Algorithm [5]. However, it requires the replacement of the S&M algorithm in each library where S&M algorithm is used. Additionally, S. Bhattacharya in [6] and [7] has shown the possibility of BPA attack even in the presence of Ladder algorithm and Chinese Remainder Theorem. There are some research works that provide solutions to handle BPA attack and we need to check their applicability in a virtualization environment. Among them, Agosta et al. [8] suggested the elimination of conditional branch instructions to prevent the effect of execution time difference. The suggested approach can work irrespective of the type of environment, but elimination of branch instruction from each vulnerable algorithm from each library is difficult to implement.

The solution suggested by Y. Tan [9] works effectively if the Cross-VM BPA attack is launched by a method other than DTA. However, it may result in performance degradation for legitimate processes occupying BTB with a high ratio. Blacklisting approach proposed by Julie [10] considered a virtualization environment, which may fail if the white-listed processes like Secure Hypertext Transfer Protocol (HTTPS) and Secure File Transfer Protocol (SFTP) are compromised.

S. Bhattacharya [11] proposed a DTA handling mechanism where a randomization module is executed just to change the state of BPU. Changed status of BPU prevents the spy process from getting correct branch prediction information and fails to launch the BPA attack. Although the proposed work can be applied in virtualization also, the performance of some legitimate processes may also get affected.

Analysis of existing solutions reveals that there is a need to propose a solution that can handle Cross-VM BPA attack without affecting the performance of any other method.

As a part of practical analysis of Cross-VM BPA attack, we have simulated two of the attack methodologies: DTA and TDA. Implementation details and results of the simulation are provided in the next section.

V. ATTACK SIMULATION

A. *Simulating Direct Timing Attack*

Simulation of Cross-VM DTA requires a total of three VMs: an attacker (VM1), a victim (VM2) and a shared VM (VM3). VM1 accesses the RSA cryptographic library located on shared VM3 We initiated DTA from the victim attack VM2 where we assume that VM2 has already trapped encrypted private key packet of VM1. As per the procedure of DTA, number of mispredicted branches are calculated for each message in each of the four sets as per the discussion in sub-section A of section II. As per equation 1, prediction of a secret key-bit is carried out based on the comparison results among the calculated average values of each message set, namely Avrg(MB(M1)), Avrg(MB(M2)), Avrg(MB(M3)) and Avrg(MB(M4)). In Figure 4, obtained results of above-discussed parameters are plotted based on which secret value of one bit is predicted. The plot reflects the results observed during the prediction of 45th bit and the procedure is repeated for predicting each of the unknown bits. The results of plot reveal that average values of MB(M1) and MB(M4) are higher than MB(M2) and MB(M3) respectively which correctly predicts the next bit as 1.

Retrieval Number: B3553078219/19©BEIESP
DOI: 10.35940/ijrte.B3553.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

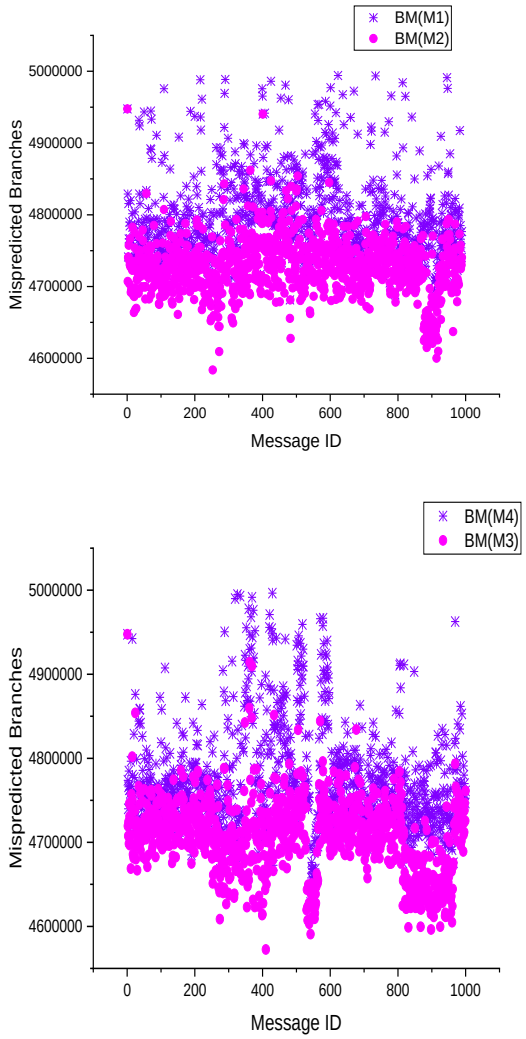

Figure 4. Distribution of MB (M1) and MB (M2) Distribution of MB (M3) and MB (M4)

B. *Simulating Time Driven Attack*

Cross-VM TDA requires both the attacker and the victim VMs to be operated by a common core. We implemented the environment with Gem5 simulator, where we had set the total number of BTB entries to 2048. A spy process with large number of conditional branch instruction to full BTB was initiated on VM1, the attacker VM. VM2, the victim VM, was set to run a victim process that calls RSA algorithm. Both the spy and victim processes were parallelly executed to measure execution time for the means of launching TDA. We observed the CPU clock cycles to find the difference between successive iterations. Observed results, shown in Figure 5, reveal that a considerable value of CPU clock cycle was observed at each iteration where the corresponding bit was 1 in the private asymmetric key. In Figure 5, we have shown values of CPU clock cycles for a group of private key-bits among the entire key-bit sequence.

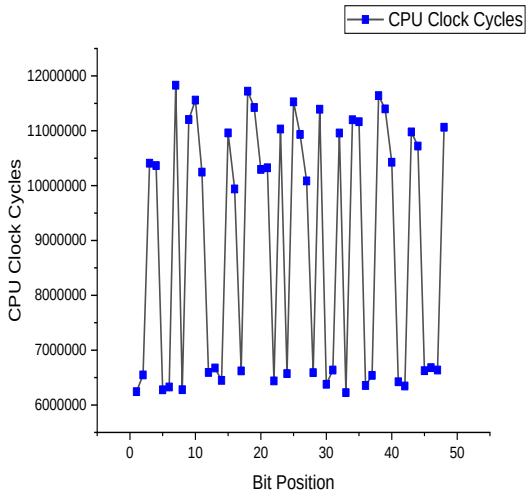

Figure 5 Simulation Results of Cross-VM Time Driven Attack

VI. CONCLUSION

BPA (Branch Prediction Analysis) attack is one such Side Channel Attack that can extract private key without explicitly performing any malicious action. The attack can be successfully launched by expoiting one of the services among multiple services hosted by a single machine. However, we need to analyze the scope of BPA attack in virtualization environment looking to its spread in current technology. In this paper, we discuss Cross-VM BPA attack which may take place between the victim and the spy process residing on different co-resident VMs. We check the applicability of different BPA attack launching mechanisms with reference to underlying resource sharing configuration. Our analysis reveals that among four different attack launching methods, only Direct Timing Attack is possible if the CPU cores of the two VMs are not shared. At the same time, launching of DTA requires shared cryptographic library unlike other three methods. We practically analyze the behavior of BPA attack launched by Direct Timing attack and Time Driven attack

We also analysis the existing solutions handling BPA attack to assess their scope in virtualization environment. Scope analysis of existing solutions reveal that there is need to propose a solution that can handle Cross-VM BPA attack that overcomes the limitations of present approches.

REFERENCES

1. T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds,” in Proceedings of the 16th ACM conference on Computer and communications security, 2009, pp. 199–212.
2. J. Seifert, Ç. Koç, and O. Aciiçmez, “Predicting Secret Keys Via Branch Prediction,” Ct-Rsa, pp. 225–242, 2007.
3. O. Aciiçmez, Ç. K. Koç, and J.-P. Seifert, “On the power of simple branch prediction analysis,” in Proceedings of the 2nd ACM symposium on Information, computer and communications security, 2007, pp. 312–320.
4. O. Aciiçmez, S. Gueron, and J. Seifert, “New branch prediction vulnerabilities in OpenSSL and necessary software countermeasures,” Cryptography and Coding. pp. 1–16, 2007.

5. M. Joye and S.-M. Yen, "The Montgomery Powering Ladder," in International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), 2002, pp. 291–302.
6. S. Bhattacharya et al., "Fault Attack revealing Secret Keys of Exponentiation Algorithms from Branch Prediction Misses.," IACE Cryptogr. ePrint, p. 790, 2014.
7. S. Bhattacharya and D. Mukhopadhyay, "Who Watches the Watchmen?: Utilizing Performance Monitors for Compromising Keys of RSA on Intel Platforms," Springer, Berlin, Heidelberg, 2015, pp. 248–266.
8. G. Agosta, L. Breveglieri, G. Pelosi, and I. Koren, "Countermeasures against branch target buffer attacks," in Fault Diagnosis and Tolerance in Cryptography, 2007. FDTC 2007. Workshop on, 2007, pp. 75–79.
9. Y. Tan, J. Wei, and W. Guo, "The micro-architectural support countermeasures against the branch prediction analysis attack," Proc. - 2014 IEEE 13th Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2014, pp. 276–283, 2015.
10. J. Sebot and S. Gueron, "Mitigating branch prediction and other timing based side channel attacks," US Pat. 8,869,294, 2014.
11. S. Bhattacharya, S. Bhasin, and D. Mukhopadhyay, "Online Detection and Reactive Countermeasure for Leakage from BPU Using TVLA," in 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), 2018, pp. 155–160.

AUTHORS PROFILE

Dhara Buch was born in Rajkot City, Gujarat India in 1980. She received the B.E. and MTech degrees in Computer Engineering from Gujarat University in 2001 and SVNIT, Surat in 2012 respectively. She is currently perusing her Ph.D. from Gujarat Technological University, Ahmedabad. From 2001 to 2005, she was a Lecturer with MVM College, Rajkot. From 2005 to 2010, she was with AVPTI, Rajkot. Since, 2010, she has been Assistant Professor in Computer Engineering at Government Engineering College, Rajkot. She has published more than 3 research articles. Her research interest includes information security and cloud computing. Ms. Buch is an active member of Institutions of Engineers, India and Indian Society for Technical Education, India.

Haresh S Bhatt was born in Rajkot City, Gujarat, India in 1961. He received the B.Sc. and M.Sc. degrees in Physics from the Saurashtra University, Rajkot in 1981 and 1983, Advanced PG Diploma in Space Science and their Applications and the Ph.D. degree in Computer Science from Gujarat University, Ahmedabad, in 1984 and 2003. He is associated with Indian Space Research Organization since 1984. Currently, he is chairman of Information & Cyber Security Board and Mission Director, Information Security, at Space Applications Centre. His remarkable contribution in automatic cloud covers estimation and in solving on-board sensor calibration problems for IRS-1C was internationally acclaimed and was first of its kind. He has carried out pioneering work in the field of satellite-based grid computing. He was chairman of task team of Indian LRIT NDC implementation for tracking the ships sailing across countries to comply with International Law about Safety of Life at Sea. He has published 50+ papers in International Journals and conferences. He has actively participated in organizing various conferences. He has 2 patents from India and 1 patent from Singapore. His area of interest is information & cyber security, cloud computing and grid computing. Currently, 3 research students are pursuing PhD under him at various universities. Dr. Bhatt was a recipient of the UN/ESA long-term fellowship award of Office of Outer Space Affairs United Nations in 1994, ISRO team award for work related to disaster communication in 2008, CSI patron award in 2010, LRIT project got CSI appreciation award for national G2G in 2012, and CSI Fellow award in 2013.

Table I Sharing Configuration for Cross-VM BPA Attack

BPA Attack Mechanism	Required Sharing Configuration	
	Cryptographic Library	CPU Core and BTB
Direct Timing Attack	Shared	Shared / Separate
Asynchronous Attack	Shared / Separate	Shared
Synchronous Attack	Shared / Separate	Shared
Time Driven Attack	Shared / Separate	Shared