## PROPOSAL TITLE (CALIBRI, 22 pt) Entrant Names

## Department

## Institute

## Email IDs

# **Introduction and Problem Statement**

Many modern cryptosystems are vulnerable to side-channel attacks even after the recent advancement of cryptographic algorithms. These attacks compromise the secret key by the information gained through the physical implementation of the cryptosystem. One category of such attack includes Micro-architectural side-channel-attacks which can retrieve the secret key by observing micro-architectural functionalities of the processor implementation, like cache accesses, branch instructions, etc. [2, 3]. Modern microprocessors contain a set of special purpose registers to measure hardware related activities known as hardware performance counters, which leak valuable information regarding the encryption algorithm [1]. Some attacks [4] analyze these performance counters for compromising the security of the system.

There are many state-of-the-art countermeasures to prevent the side-channel attacks, but with the cost of an extra overhead of implementation. Implementation of these countermeasures are not feasible in resource constraint environment like IoT devices, Smart-phones etc. There has been some work [5, 6] to prevent these types of attacks by observing the performance counters of the adversary program, which it leaves behind while executing on the system. One possible way to detect and prevent these attacks is to analyze the hardware footprints of the system in real time using data analytics techniques and classify the state of the system as safe or unsafe. A safe state is when there is no existence of any side-channel-attack, and the unsafe state signifies the execution of any adversary program in the background. Most of these techniques state about preventing the cached-based side-channel attacks using some machine learning methods without generalizing the detection method. Moreover, some advanced techniques like *drammer* [7] exploit the row hammer hardware vulnerabilities on Android and iOS systems to take full control over the system. This is also a motivation to implement a detection method with negligible implementation overhead such that it can be used in any device with resource constraint.

Apart from side-channel-attacks there are some advanced malwares like FireEye which disrupts the normal flow of a system and gain authorized access and there are also malwares like ransomware, which encrypts victim’s data until a ransom is paid. We believe these kind of malwares also leave behind their executing footprints on the hardware events. This is also a motivation to implement a detection technique which will detect and prevent these types of attacks.

The data observed from hardware performance counters on both encryption and adversary processes form a time-series, which has been extensively studied in the context of economics, weather prediction, population dynamics, earthquake prediction, etc. Recently, time-series models as well as continuous-time models have been studied for prediction of events, such as retweets [8], opinions propagation [9], etc on online social networks. These models capture complex dependencies with past events, e.g. mutual excitation, non-local dependencies, periodic patterns, etc. Such models for temporal data processing can be helpful in learning dependencies between HPC values, e.g. a cache miss 5 time steps after a branch , which can form a sparse set of features prediction of attacks.

# **Proposal Description**

In this proposal, we present the following research goals –

1. We propose to implement a machine learning based countermeasure for side-channel attacks, which will detect and prevent any adversary program executing on a system in real-time with minimal false positives and false negatives. The observation of the performance counters given in Fig. 1. clearly discriminates between the states where there is no background processes and there are some background processes. This discrimination can be easily detected by some state-of-the-art classification algorithm. But, with this categorization, there is no way that we can say an attack code is running in the background. The background process could be an benign process.

So, our proposal is, we try to model a multivariate probability distribution for the programs running on a particular system as benign and as anomaly with the HPC values collected during runtime. The benign processes are those for which the performance counter values are uncorrelated with the keys. Whereas, in case of an anomaly (the adversary program), the performance counters will have a strong correlation with the key. The anomaly model will help us to detect the user activities which create false negatives and the benign model will help to identify those attacks which were not previously trained. So, we will be able to achieve low false positives and false negatives with this approach.

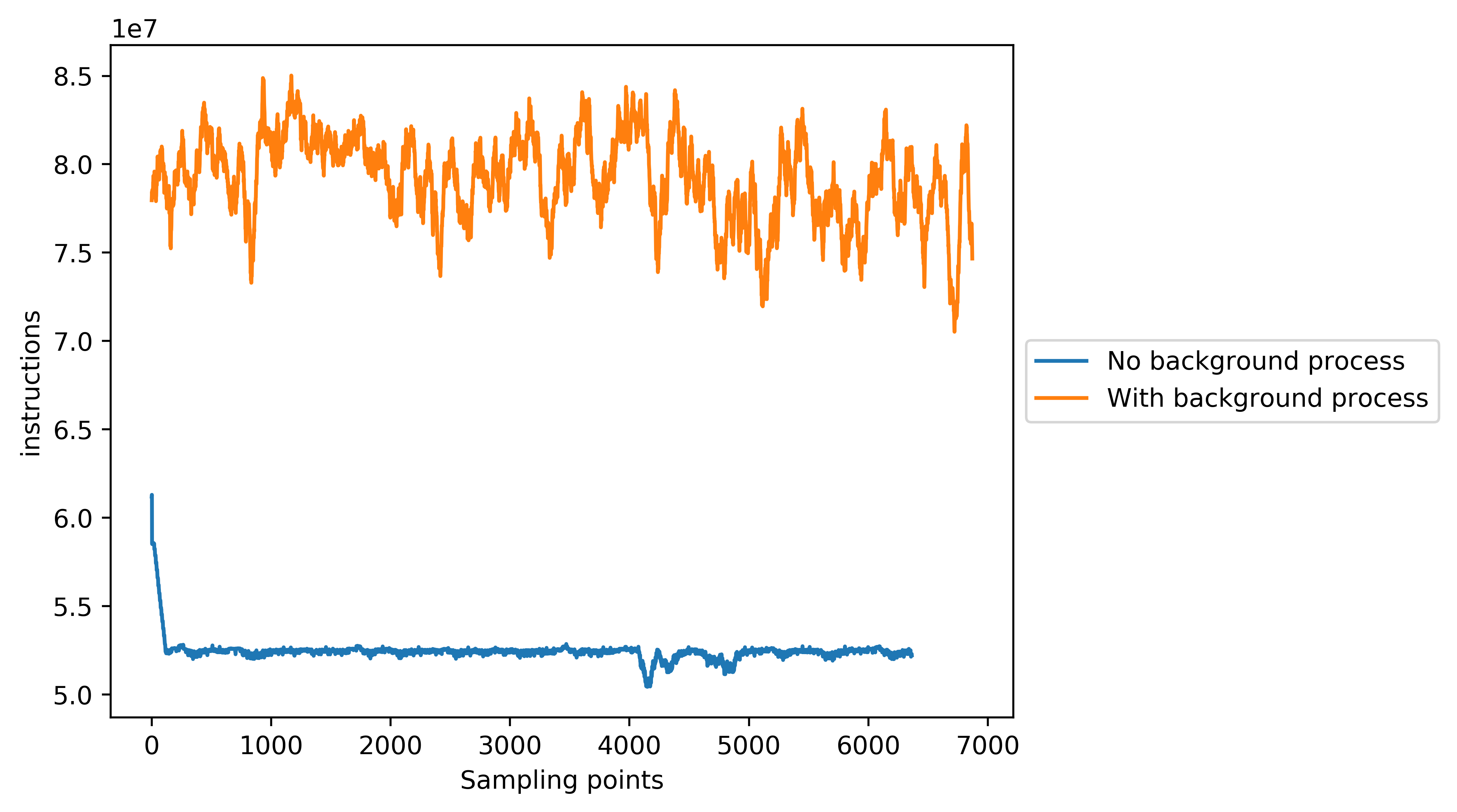


Fig. 1. : Number of instructions executed, sampled over time, for two states of a system. When there is no background process and there exists any background process.

To generate the dataset for modeling the detection method we will consider different hardware performance counter values as different features. In modern Intel based architecture we can measure at most 8 performance counters in parallel, because of that we will use feature selection to effectively select the hardware events for generating the dataset for training.

1. We would like to implement this online machine learning based detection algorithm with negligible overhead cost. So that it can be used in any system with resource constraint. We would also like to prevent the exploits like drammer in the Android or iOS devices using our model. The main objective of this online detection technique will be to ensure the security of a device with resource constraints from as much adversarial attacks as possible.
2. The recent increase of polymorphic malwares also motivates us to design a generalized detection technique for general systems and also for Android/iOS devices. We would like to extend the detection methods for malware detection also. Malwares generally harm a system by some system calls. We can track the system calls in the run-time and combined with the performance counters we will be able to distinguish a malicious application from a benign one with some state-of-the-art classification techniques.

# **Project Plan**

The project milestones of our proposal are –

1. We will first try to build an efficient machine learning based online detection method to prevent adversarial attacks such as micro-architectural attacks, memory attacks and malicious softwares. We will also try to implement this method with as much lower overhead cost as possible.
2. We would like to extend our work for the detection of malwares. For that, we will try to build a system which will be able to track system calls in the run-time and will be able to classify malicious applications with dynamic analysis.
3. After that we will try to improve our detection technique by incorporating time series analysis of the data which can be used to learn the dependencies between the HPC values.

# **REFERENCES**

1. Leif Uhsadel; Andy Georges; Ingrid Verbauwhede. Exploiting Hardware Performance Counters. 2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography, August 2008.
2. J. Bonneau and I. Mironov. Cache-Collision Timing Attacks Against AES. In L. Goubin and M. Matsui, editors, CHES, volume 4249 of Lecture Notes in Computer Science, pages 201-215. Springer, 2006.
3. Onur AcıiçmezÇetin Kaya KoçJean-Pierre Seifert. Predicting Secret Keys Via Branch Prediction. Topics in Cryptology -- CT-RSA 2007: The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA. February, 2007.
4. Sarani Bhattacharya and Debdeep Mukhopadhyay. Who Watches the Watchmen?: Utilizing Performance Monitors for Compromising Keys of RSA on Intel Platforms. Cryptographic Hardware and Embedded Systems -- CHES 2015: 17th International Workshop, Saint-Malo, France. September, 2015
5. Tianwei Zhang; Yinqian Zhang; Ruby B. Lee. CloudRadar: A Real-Time Side-Channel Attack Detection System in Clouds. Research in Attacks, Intrusions, and Defenses: 19th International Symposium, RAID 2016, Paris, France. September, 2016.
6. Marco Chiappetta; Erkay Savas; Cemal Yilmaz. Real time detection of cache-based side-channel attacks using hardware performance counters. Applied Soft Computing, Volume 49, December 2016.
7. V. van der Veen, Y. Fratantonio, M. Lindorfer, D. Gruss, C. Maurice, G. Vigna, H. Bos, K. Razavi, C. Giuffrida. Drammer: Deterministic Rowhammer Attacks on Mobile Platforms, in: CCS, 2016.
8. Huawei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. 2014. Modeling and predicting popularity dynamics via reinforced Poisson processes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14). AAAI Press 291-297.
9. Abir De, Isabel Valera, Niloy Ganguly, Sourangshu Bhattacharya and Manuel Gomez Rodriguez. Learning and Forecasting Opinion Dynamics in Social Networks. *Annual Conference on Neural Information Processing Systems (NIPS) 2016.*