Programming Assignment 1
(Cache Side Channel Attack: Demonstration and Countermeasure)
CS 665: Secure Memory Systems
2018-2019 – Semester I
Computer Science and Engineering
Indian Institute of Technology Kanpur
Due Date: Submit all the Task(s) by 16th September, 2018 11:55 PM

The programming assignment is designed to ensure that you are acquainted with the working principle of CPU Cache and its related security issues. In this assignment you are going to perform side channel attack on the CPU cache and last-level cache (LLC) and to exploit its vulnerability to leak information.

[You must be kidding: 3 points] Prior to perform your own attack, you are urged to emulate **"FLUSH+RELOAD"** attack to get familiarized with the methodologies. The details about the attack is available in the web-link https://github.com/IAIK/cache_template_attacks. You will find FLUSH+RELOAD attacks performed over multiple applications, viz. Automated keypress profiling on "gedit", OpenSSL AES T-table attack.

Once you have acquired the knowledge, begin with the following task lists.

A. **[Hmmm. makes sense: 4 points]** Perform **PRIME+PROBE** attack on L1 Dcache by running two threads in an SMT core. Mimic the similar approach used in **"FLUSH+RELOAD"** keypress profiling in "gedit". You can use your own PC/laptop for the hardware framework.

B. **[Hmmm. makes sense: 6 points]** In **Task A** it is limited to L1 and/or L2 cache that happens to be in the same core for a particular CPU. Overcome the limitation of the attack by performing over **LLC** that is shared across multiple cores. Try attacking an application called GnuPG (version 1.4.13).

C. **[It's easy right: 3 points]** Mount the **Task B** on the gem5 simulator. Get visualization of the attack vector and its effect on LLC latency. Also, show the cache block addresses of interest through which the information get leaked.

D. **[OK, no big deal: 4 points]** Propose a mitigation method to combat attack done in **Task A** and **Task B**. Biswa will talk about mitigation techniques in class. Implement any of them.

[Note: All the task(s) has to be be done on Linux Operating System. We will accept your submission via Canvas only, and other submission mode is strictly prohibited such as submitting via email or piazza. While using Piazza/email, use PA1 in the header]

Deliverables:

1. Source code for A and B, and gem5 patches for C and D. Document your code properly. Share your results (wait for Biswa’s lectures) and experiences in a pdf: why something worked/did not work. Any insights, any experiments that you did to validate your assumption/confusion. Biswa will talk about it in details.
2. In the pdf, provide division of labor: Who did what, Say Biswa did A, Biswa2 did B and 25% of C.
3. Feel free to discuss about the assignment among your friends and with Biswa and Saurabh. However, at the end of the day, we want you to do everything on your own.