CS 598CLF – Secure Processor Design

Fall 2017
Instructor: Chris Fletcher
Location: Siebel, Room 1103
Time: Tuesdays and Thursdays, 12:30pm to 1:45pm
Course website: http://cwfletcher.net/598clf.html

Prerequisites:
CS 433 (Computer Architecture) or consent of instructor

Helpful, not necessary:
CS 461 (Computer Security)
ECE 385 (Digital Systems Laboratory)
CS 423 (Operating Systems Design)

Catalog description:
With the emergence of systems such as ARM Trustzone and Intel Software Guard Extensions, secure processors have become one of the next frontiers in secure systems design. Secure processors allow emerging applications (e.g., computation outsourcing) to be realized with a significantly smaller trusted computing base and/or significantly reduced performance overheads, relative to a "pure software" solution.

This course will bring students to the cutting-edge in secure processor architecture by examining the interplay between hardware, software and applied cryptography in these systems. The first several classes will feature lectures from the instructor: to give background on secure hardware systems from the standpoints of Computer Architecture and Applied Cryptography. The body of the course will be readings and discussion of top papers in the field. Course assignments will give students hands-on experience with the Intel Software Guard Extensions (SGX) SDK, building secure applications and evaluating their security. The end of semester will culminate in an original research project.

Intended audience:
This class is primarily intended for students who would like to conduct secure systems research where hardware plays a first-class role. It will also appeal to students with a casual interest in this hot topic, or who want to do research in computer security generally.

Grading:
- Paper reading (2 papers / week) - 35%
  - Discussion lead for 1-2 papers – 10%
  - 500 word summary + a discussion question / paper – 20%
  - Participation – 5%
- SGX Lab - 15%
- Final project - 40%
  - Proposal – 10%
  - Checkpoint – 10%
  - Final writeup (+ artifact if applicable) – 25%
  - Final presentation – 5%

Topics:

- Secure processors in industry
  - Tamper resistance
  - Attestation/Secure boot
  - Isolation
- Intel SGX as a case study (3 papers)
  - Hardware internals for Intel SGX
  - How to write enclave programs
  - Foundations for enclave programming
  - Attacks on shared-memory enclaves (shared resource, virtual memory)
  - Programming experience with the Intel SGX SDK
- Shared resource attacks and defenses (6 papers)
  - Primitive attacks (prime+probe, flush+reload, DRAM-based, use of cache line attributes)
  - Turning primitive attacks into application-level attacks
  - Co-locating to the same physical machine in a cloud environment
  - HW defenses to shared resource attacks
  - SW defenses “”
  - PL-inspired defenses “”
- Bootstrapping trust / foundations of trusted execution (2 papers + 1 paper from SGX section)
  - Authentication (EPID, PUFs)
  - Native sources of randomness (from circuit metastability, memory state)
- Hardware isolation leading up to SGX (3 papers)
  - Based on Intel TXT (Flicker)
  - Precursors to Intel SGX (Bastion)
  - Systems inspired by Intel SGX (Sanctum)
- Attacks on program memory (3 papers)
  - Broad overview of area (types of code/data injection, software fault isolation via CFI/DFI, memory safety)
  - Lightweight to heavyweight hardware mechanisms for memory safety
- Physical attacks and defenses (5 papers)
  - Differential power/fault analysis
  - Hardware trojans (attacks and defenses)
  - Attacks on DRAM
  - External memory (DRAM or otherwise) protection
  - On-chip memory protection