5925 Sycamore Canyon Blvd
Riverside CA 92507

Ju Chen

(585) 434-8060
chenju2k6@gmail.com
http://chenju2k6.github.io

Summary

- Ju Chen is currently a Ph.D. candidate in Computer Science at the University of California at Riverside, focusing on building scalable dynamic program analysis tools including symbolic execution and constraints solving to find bugs in software. Before his Ph.D. study, he was a software engineer at Intel Corporation, emphasizing Linux Kernel development. Ju Chen is enthusiastic about building efficient, secure, and widely-used software systems.

Employment

- **Summer 2019:** Research Intern, Baidu USA, Sunnyvale CA
- **Spring 2008 - Fall 2015:** Software Engineer, Intel Corporation, Beijing, China
- **2007:** Software Engineer Intern, Agilent Technologies, Beijing, China

Project Experiences - in reverse chronological order

- **A time and space efficient symbolic executor**.
  - I designed and implemented a scalable symbolic executor that is two orders of magnitude time faster than state-of-the-art tools and has a much smaller (up to 1000x) memory footprint.
  - Paper advanced to second round review in a top-tier security conference.
  - Tech stack: C++/Rust/LLVM
- **An efficient path constraints solver**.
  - I designed and implemented a super-fast constraints solver that is linearly scaled to multi-cores and beat popular constraint solvers.
  - Paper accepted to Oakland'22 (Updated April 7 2022)
  - Tech stack: C++/protobuf/Rust/LLVM/JIT
- **MesaTee: Baidu's secure computing framework - a project participated as an intern in Baidu USA**
  - I created a key-value store library prototype based on LevelDB and integrated it into the MesaTee framework (internal branch).
  - Tech stack: C++/Rust/Intel SGX
- **Cache-oblivious computing framework**
  - I designed and implemented a cache-oblivious computing framework. It manages a memory mapping inside the CPU cache and guarantees zero-cache-misses on writes/reads issued by upper-layer algorithms such as merge-sort, k-means, and sorting network.
  - Paper accepted by SysTex 2017 at ACM SOSP
  - Tech stack: C++/Assembly.
- **Secure key-value store using Intel SGX**
  - I refactored LevelDB to be a secure storage system using Intel SGX.
  - Paper accepted by Middleware'21
  - Tech stack: C/C++/key-value storage system/Intel SGX
- **Linux device drivers development**
  - I was a member of the core development team for Intel peripherals such as USB and graphical devices (display controller and 3D engine). Our device drivers are directly used or referenced by NEC, Dell, Lenovo, and many more.
  - I fixed numerous urgent software issues raised by Intel's OEM customers and received positive feedbacks.
  - I was the primary maintainer for the display controller driver.
  - I was the primary maintainer for the USB-client and USB-over-IP driver.
  - Tech stack: C/Linux Kernel/Hardware specs

Education

Riverside, CA — **University of California, Riverside** — **Fall 2019 – Present**
- Ph.D. candidate in Computer Science (anticipated graduation date: September 2022)

Syracuse, NY — **Syracuse University** — **Fall 2015 – 2019**
- Ph.D. candidate in Computer Science
- Graduate Coursework: Computer Security; Cloud Computing; Operating systems; Applied Cryptography

Beijing, China — **Beihang University** — **Fall 2001 – Spring 2008**

- Bachelor and Master in Electrical Engineering

Teaching

Teaching Assistant **University of California, Riverside** **Spring 2020 - Present**
- CS153 - Design of Operating Systems (course website)

Teaching Assistant **Syracuse University** **Spring 2017 - May 2019**
- CIS655 - Advanced Computer Architecture (course website)
- CIS/FIN600 - Blockchain and Cryptocurrencies (course website)

Research Experiences

- **2019-present** Finding software vulnerabilities in binaries and open-source projects using fuzz testing and symbolic execution. The research results are in the peer-review process in the top CS conferences.
- **2015-2018** Enabling secure key-value storage systems using hardware-assisted trusted execution environment and enabling efficient confidential computing. The research results are published in SecureComm and Systex.

Publications

- Yuzhe Tang, **Ju Chen**, Kai Li, "Authenticated LSM Trees with Minimal Trust", SecureComm 2019
- Qiwu Zou, Yuzhe Tang, **Ju Chen**, Kai Li, Charles Kamoua, Kevin Kwiat, Laurent Njilla. "ChainFS: Blockchain-Secured Cloud Storage", IEEE Cloud 2018
- K. Areekijseree, Yuzhe Tang, **Ju Chen**, Shuang Wang, Arun Iyengar and B. Palanisamy. "Secure and Efficient Multi-Party Directory Publication for Privacy-Preserving Data Sharing." SecureComm 2018, AR=30.6%
- Yuzhe (Richard) Tang, Zihao Xing, **Ju Chen**, Cheng Xu and Jianliang Xu. "Lightweight Logging over the Blockchain for Data-Intensive Applications", 2nd Workshop on Trusted Smart Contracts 2018 at Financial Cryptography (Workshop paper)
- **Ju Chen**, Yuzhe (Richard) Tang and Hao Zhou. "Strongly Secure and Efficient Data Shuffle on Hardware Enclaves", SysTex 2017 at ACM SOSP (Workshop paper)
- Yuzhe Tang and **Ju Chen** "Log-structured Authenticated Cloud storage with minimal trust using Intel SGX", Technical Report (https://eprint.iacr.org/2016/1063.pdf)
- John Ye, **Jason Chen**, Tianzhou Chen and Qinsong Shi, "Conflict-Free Code Block Scheduling to Hide SpMT Inter-Core Register Sync Delay", PDCAT '14
- John Ye, **Jason Chen**, Tianzhou Chen, Minghui Wu and Li Liu, "Offline Data Dependence Analysis to Facilitate Runtime Parallelism Extraction", CSE '14
- **Ju Chen**, Qi Zhao and Jinming Dong, "Research on kernel encoding function of H.264 CODEC JM8.6", Computer Engineering and Design 2008-17

Awards and services

- **2017:** iDash 2017 Student Travel Grant
- **2009 2010:** Intel Division Recognition Award
- **Conference Reviewer:** TKDE and ICPADS

Skills

- **Main focus:** Operating systems-level programming (e.g. IPC, forking and signal handling), Just-in-Time compilation, program instrumentation, performance profiling, and performance optimization.
- **Programming Languages:** Proficient in C/C++/Rust/Python. Familiar with Java/Lua/Matlab/HTML/Assembly.
- **Tools, libraries and open-source code-bases:** Proficient in GDB/Git/CMake/Gperftools/Valgrind/protobuf/LevelDB/AFL.