5925 Sycamore Canyon Blvd
Riverside CA 92507

Ju Chen

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

Short Bio

- Ju Chen is a research scientist at Deepbits Inc., a startup cybersecurity company. Ju Chen was a software engineer at Intel for 7 years and then got his Ph.D. in Computer Science from UC Riverside. Ju Chen has recent publications in top-tier CS conferences. On Github.com, Ju Chen received 175 stars for self-owned projects and 1300 stars for the projects he has contributed. Ju Chen has 14 years of experience in systems software and 7 years of experience in cybersecurity research.

Employment

- **2022 to the present:** Research Scientist, Deepbits Inc.
  - Software supply chain security
- **2019 - 2022:** Research Assistant, UC Riverside, Riverside CA
  - Automatic software vulnerability discovery
- **Summer 2019:** Research Intern, Baidu USA, Sunnyvale CA
  - Trusted computing infrastructure
- **2015 - 2019:** Research Assistant, Syracuse University, Syracuse NY
  - Hardware-assisted trusted computing and blockchain
- **2008 - 2015:** Software Engineer, Intel Corporation, Beijing, China
  - Linux Kernel development.

Education

Riverside, CA **University of California, Riverside** **2015 – 2022**
- Ph.D. in Computer Science

Beijing, China **Beihang University** **2005 – 2008**
- Master in Electrical Engineering

Beijing, China **Beihang University** **2001 – 2005**
- Bachelor in Electrical Engineering

Selected project experiences - in reverse chronological order

- **A time and space efficient symbolic executor**
  - Two orders of magnitude time faster than state-of-the-art tools and up to 1000x smaller memory footprint. Near-optimal performance in all source-based symbolic executors
  - Paper accepted to USENIX Security Symposium 2022 (top-tier security conference)
  - Open-source project (https://github.com/r-fuzz/symsan) received 135 stars.
  - Designed and implemented a scalable symbolic executor
  - Tech stack: C++/Rust/LLVM
- **An efficient and scalable path constraints solver**
  - Out-performed Z3, a popular constraints solver by up-to 85 times. Won the 1st place in Google's fuzzing benchmark
  - Paper accepted to IEEE Symposium on Security and Privacy 2022 (top-tier security conference)
  - Open-source project (https://github.com/r-fuzz/jigsaw) received 40 stars
  - Designed and implemented a super-fast path constraints solver.
  - Tech stack: C++/protobuf/Rust/LLVM/JIT
- **MesaTee: Baidu's trusted computing framework - a project participated as an intern in Baidu USA**
  - Designed and created MesaTee's secure storage feature.
  - Integrated to Baidu's internal branch
  - Tech stack: C++/Rust/Intel SGX
- **Cache-oblivious computing framework**
  - Paper accepted by SysTex 2017 at ACM SOSP
  - Designed and implemented a cache-oblivious computing frameworks
  - Secured popular data processing algorithms (resilient to side-channel attacks)
  - Tech stack: C++/Assembly.
- **Hardware-assisted trusted key-value storage**
  - Paper accepted by Middleware'21

  - Designed and implemented trusted storage system based of Intel SGX.
  - Tech stack: C/C++/key-value storage system/Intel SGX
- **Linux kernel development - 7 years at Intel as SWE**
  - Supported USB, Display, Audio and Graphics devices drivers
    - Member of the core software development team for Intel platforms
    - Owned USB/HDMI/Display modules
    - Bug fixing, new feature enabling, code refactoring
    - Post-Silicon validation
    - Linux Kernel up-streaming
    - Drivers proudly deployed to millions of Intel-powered devices globally
  - Led software prototyping for proof-of-concepts
    - USB-over-IP
    - WiFi-direct
    - Camera intelligence
  - Led OEM customer supporting - NEC, Dell, Asus and Lenovo
    - Communicated with customers that had limited technical background
    - Led cross-team collaboration to fix urgent issues
    - Won customers' recognition and trust
  - Tech stack: C/Linux Kernel

Publications

- **Ju Chen**, Wookhyun Han, Mingjun Yin, Haochen Zeng, Chengyu Song, Byoungyong Lee, Heng Yin, and Insik Shin. ”SymSan: Time and Space Efficient Concolic Execution via Dynamic Data-Flow Analysis”, 31st USENIX Security Symposium 2022
- **Ju Chen**, Jinghan Wang, Chengyu Song, Heng Yin, ”JIGSAW: Efficient and Scalable Path Constraints Fuzzing”, 43rd IEEE Symposium on Security and Privacy 2022
- Yuzhe Tang, K. Li, Q. Zhang, J. Xu, **Ju Chen**. “Authenticated Key-Value Stores with Hardware Enclaves”, ACM/IFIP Middleware 2021 (Industrial track)
- 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.