Conghao Shen

tomshen@stanford.edu, (510)-387-8301

GitHub: https://github.com/tsunrise, **LinkedIn:** https://www.linkedin.com/in/conghao-shen-b179b5169/
Personal Website: https://tomshen.io **Interests**: Systems, Security, Programming Language, Machine Learning

Education

Stanford University
Computer Science M.S., *Sep 2022 – Jun 2024*
University of California – Berkeley
Computer Science B.A., *Aug 2018 – May 2022, Graduated with GPA of 3.98/4*

Experience

Arista, Software Engineer (Intern), *May 2021 – Aug 2021*

- Developed backend to connect REST APIs to interact with low-level endpoints of network switches.
- Wrote programs in Go that works in clusters environment like Kubernetes.
- Designed automation tests to check the correctness of existing logic without time-consuming deployment.
- Utilized Gerrit code review system to communicate effectively with team members.

Manta Network, Pari-time Contributor (Open Source), *Sep 2021 – Present*

- Implement cryptography-related algorithms like Poseidon Hash and Trusted Setup for cryptocurrency-related Apps.
- Use Rust trait and constant generics to reduce code duplication without adding too much cognitive overhead.
- Do code review and make constructive comments using GitHub Pull Requests.
- Work on continuous integration (CI) scripts to ensure code quality and correctness.

Projects

Arkworks (https://github.com/arkworks-rs)**,** *May 2020 – Present*
Advised by Professor Alessandro Chiesa at UC Berkeley
Related Areas: Zero-knowledge proof, System Building, Cryptography, Open Source, Rust

- Implemented state-of-the-art cryptographic primitives such as the low-degree test using FRI, cryptographic sponge, polynomial commit scheme, and Merkle Tree.
- Write R1CS circuits of those crypto primitives so that they can be easily integrated into proof systems.
- Implement complex protocols like interactive oracle proof system [BCS16] in Rust, using my multiple previously written code as libraries.
- Build a relatively large system (~10k lines) from scratch. Learn how to maintain it with proper documentation.

LISA – Lightweight Secure Aggregation for Federated Learning, *August 2022*
Advised by Professor Raluca Ada Popa at UC Berkeley.
Related Areas: Zero-knowledge proof, System Building, Cryptography, Open Source, Rust

- Build a secure aggregation protocol for federated learning under distributed trust model.
- Design a protocol that performs machine learning without revealing clients' gradients to protect privacy.
- Work on server implementation that serves more than 10000 client connections simultaneously.
- Design customized application layer over TCP, allowing multiple tasks to share a single socket concurrently.
- Use mutex, multi-producer single consumer (MPSC) queue, and one-way channels to ensure synchronization.
- Use asynchronous programming to reduce CPU idle time when there is high demand for IO.

Highlighted Projects from Past Coursework

- End-to-end Encrypted Storage: Built cloud storage on untrusted servers that protects privacy and integrity even when a malicious party has full control. My submission was one of ten (out of 100+) whose submission survived all attacks.
- Lisp Compiler: Implement Compiler from lisp-style code to x86 assembly. Supports arithmetic, variables, heap allocation, closure with capture, and optimizations such as inlining, constant propagation, and common subexpression elimination.