

Conghao (Tom) Shen

Stanford, California, US | +1 (510) 387-8301 | tom.shen.cn@gmail.com
github.com/tsunrise | linkedin.com/in/conghao-shen/ | tomshen.io

PROFESSIONAL SUMMARY

First-year Master's student in Computer Science from Stanford University with a passion for building efficient, scalable, and secure systems. Over five years of experience in various areas, from building simple full-stack web applications to efficient, secure multi-party computation protocols.

EDUCATION

Master of Science, Computer Science, Stanford University

Expected Graduation June 2024

- Intended Specialization in Artificial Intelligence
- Coursework: Deep Learning, Parallel Computing

Bachelor of Arts, Computer Science, UC Berkeley

Graduated May 2022

- GPA 3.98/4.0, Dean's List, Honors to Date
- Coursework: Algorithms, Security, Programming Languages, Operating Systems, Machine Learning, Probability theory

WORK EXPERIENCE

Software Engineering Intern, Arista Networks

May 2021 - Aug 2021

- Developed backend in Golang to serve REST APIs to interact with ARISTA network switches, including configuration upload, zero-touch enrollment, system reboot, and log file lookups.
- Wrote unit tests for basic logic like response parsing, and automation tests to cover the above-mentioned features.
- Refactored part of legacy Java codebases for zero-touch enrollment (~1k lines of code) to Golang Backend with full backward compatibility and updated Python scripts on switches to accommodate this change.

PROJECTS

Coauthor, LISA (Lightweight Secure Aggregation for Federated Learning)

Paper Submitted August 2022

- Built a secure aggregation multi-party computation system for federated machine learning.
- Wrote a performant server in Rust that was required to handle 50k client connections simultaneously.
- Designed a customized application layer over TCP, using message queues to share one socket safely across threads.
- Used Rust asynchronous programming to reduce CPU idle time for an IO-heavy application.

Contributor, Arkworks (<https://github.com/arkworks-rs>)

May 2020 - May 2022

- A collection of libraries (crates) in Rust for building a zero-knowledge proof system.
- Implemented state-of-the-art cryptographic primitives like low-degree tests, cryptographic sponge, commitment scheme, and Merkle Tree ([Code Samples](#)).
- Built complex protocols like interactive oracle proof systems in Rust with proper documentation.

Contributor, Manta Network DeFi (<https://github.com/Manta-Network/manta-rs/>)

Sep 2021 - Present

- Built a server for "trusted setup", a ceremony for cryptographic protocols used by DeFi with around 1000 participants.
- Integrated cryptographic algorithms like Poseidon hash function to current codebases.
- Audited and reviewed over [30 pull requests](#) and provided constructive feedback when requesting changes.

Lisp Compiler in OCaml

Sep 2021 - Dec 2021

- Used OCaml to implement a lisp-style compiler to x86 assembly. Added support for arithmetic, variables, allocation, closure with capture, and optimizations like inlining, constant propagation, and common subexpression elimination.

SKILLS

Programming Languages: Rust, Python, C, Java, Golang.

Tech/Frameworks: PyTorch, Tokio-rs, WASM, Git/Gerrit, Linux

Cloud: Serverless, CDN, Cloudflare Workers, AWS