Ryan Lehmkuhl

ryanleh@mit.edu | github.com/ryanleh

EDUCATION

MIT | *PhD in Computer Science*
2022 - Present

UC BERKELEY | *B.S. in Electrical Engineering and Computer Science*
Class of 2021 • GPA 3.9/4.0

AWARDS AND HONORS

2022:	NSF Graduate Fellowship	
2021:	NSF Graduate Fellowship Honorable Mention	
2020:	CRA Outstanding Undergraduate Researcher Finalist	*Top 32 undergraduate CS researchers in the U.S.*
2020:	UC Berkeley EECS Outstanding GSI Award	*Top 10% of student instructors*
2019:	UC Berkeley Summer Undergraduate Research Fellowship	*21 students selected (I was the only EECS major chosen)*
2017:	UC Berkeley Regents' and Chancellor's Scholarship	*Top <1% of incoming students*

PUBLICATIONS

[1] *Alessandro Chiesa, **Ryan Lehmkuhl** , Pratyush Mishra, and Yinuo Zhang. *”Eos: Efficient Private Delegation of zkSNARK Provers”.* USENIX Security '23.

[2] **Ryan Lehmkuhl** , Pratyush Mishra, Akshayaram Srinivasan, and Raluca Ada Popa. *”Muse: Secure Inference Resilient to Malicious Clients”.* USENIX Security '21.

[3] Pratyush Mishra, **Ryan Lehmkuhl** , Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. *”Delphi: A cryptographic inference service for neural networks”.* USENIX Security '20.

* *- Alphabetical author ordering*

TEACHING

2022:	Code Tenderloin Instructor	*Intro CS course for formerly incarerated or homeless individuals in SF*
2021:	ANova Curriculum Designer & Tutor	*CS course for students from under-resourced highschools in Oakland*
2020:	CS161 Co-instructor	*UC Berkeley's Computer Security course (Summer term)*
2020:	CS161 Teaching Assistant	*UC Berkeley's Computer Security course (Spring term)*
2019:	CS161 Teaching Assistant	*UC Berkeley's Computer Security course (Summer term)*

TALKS

EOS | *Efficient Private Delegation of zkSNARK Provers*

– ConsensusDay Workshop	**June 2023**
– Usenix Security	**August 2023**

MUSE | *Secure Inference Resilient to Malicious Clients*

– Usenix Security	**August 2021**
– CRYPTO Privacy-Preserving Machine Learning Workshop	**August 2021**

DELPHI | *A Cryptographic Inference Service for Neural Networks*

– CCS Privacy-Preserving Machine Learning in Practice Workshop	**November 2020**
– Theory and Practice of Multi-Party Computation Workshop	**May 2020**

EXPERIENCE

OPAQUE | *Software Engineer* **Spring 2021 – Fall 2022**

- Constructing efficient systems for private data analytics utilizing hardware enclaves.

CIRCADENCE | *Research and Development Intern* **Summers 2017, 2018**

- Researched and developed cellular network attacks utilizing software-defined radios

NAVWAR | *Research and Development Intern* **Summers 2015, 2016**

- Performed vulnerability analysis that helped earn over $200,000 in lab funding

PROJECTS

FAULT-TOLERANT DISTRIBUTED KEY-VALUE STORE | *Go* **Spring 2023**

- Built a fault-tolerant distributed key-value store using the Raft consensus protocol
- Enabled support for log compaction, sharding, and persistence

FSS | *Rust* **September 2022**

- Built a high-performance library for various function secret-sharing schemes
- Included extensions for instantiating a private information retrieval scheme

EOS | *Rust* **Fall 2020**

- Designed an asynchronous MPC system for handling computation on secret-shared polynomials
- Extended the poly-commit and Marlin libraries to support delegation
- Built a delegation framework for constructing zkSNARKS through a distributed network of workers

POLY-COMMIT | *Rust* **Summer 2020**

- Designed and implemented a multivariate polynomial commitment scheme for the poly-commit library

MUSE | *Rust, C++* **September 2019 - September 2020**

- Implemented an efficient modular reduction algorithm for garbled circuits
- Building a multi-threaded, asynchronous, two-party computation framework secure against malicious clients

DELPHI | *Rust, C++, Python* **September 2018 – September 2019**

- Developed new approaches for training convolutional neural networks that are performant with cryptographic techniques using Keras and RayTune
- Built a secure two-party protocol for convolution and matrix multiplication using fully homomorphic encryption with Microsoft's SEAL library
- Implemented a novel cryptographic protocol and inference engine (Source Code)

GENETIC SCHEDULE | *Python* **Winter 2019**

- Genetic algorithm for finding an optimal schedule given complex constraints

SCRYPTO | *Rust, Python* **Summer 2018**

- Password-protected authenticated file encryption using AES-GCM and PBKDF2

MALICIOUSLY-SECURE SHARED FILE STORE | *Python, Go* **Spring 2018**

- Fully encrypted database with hierarchical sharing/revocation and efficient updates using a Merkle Tree

SCADA NETWORK TCP SESSION HIJACKER | *Python* **Summer 2016**

- Concurrently executes ARP cache poisoning and TCP session hijacking to hack a Navy SCADA controller