Ryan Lehmkuhl

https://github.com/ryanleh
ryanleh@berkeley.edu

I am interested in using cryptography to build decentralized and privacy-preserving systems.

EDUCATION

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

RELEVANT COURSEWORK

CS294-153	Foundations of Probablistic Proofs
CS294-163	Decentralized Security: Theory and Systems
CS261	Systems Security
CS161	Computer Security
CS171	Cryptography
CS170	Efficient Algorithms & Intractable Problems
CS162	Operating Systems
CS188	Artificial Intelligence
EECS126	Probability and Random Processes
Math 113 & 114	Abstract Algebra I & II

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 nation*
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] **Ryan Lehmkuhl** , Pratyush Mishra, Akshayaram Srinivasan, and Raluca Ada Popa. *"Muse: Secure CNN inference for malicious clients".* USENIX Security '21.

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

TEACHING

Summer 2020	Co-instructor for CS161 (Computer Security)
Spring 2020	Teaching Assistant for CS161 (Computer Security)
Summer 2019	Teaching Assistant for CS161 (Computer Security)

RESEARCH

PRIVATE DELEGATION OF ZKSNARK PROVERS | *RISELab, UC Berkeley* **September 2020 - Present**

Working under Professor Alessandro Chiesa on efficient delegation of zero-knowledge, succinct, non-interactive arguments of knowledge (zkSNARKs). Our delegation scheme reduces a prover's computational overhead by up to 26× and memory cost by upwards of 256×. In submission USENIX Security 2022.

MUSE | *RISELab, UC Berkeley* **September 2019 – Present**

Worked under Professor Raluca Ada Popa on malicious-client secure inference. We demonstrate a devastating attack against many prior semi-honest secure inference protocols which allows a malicious client to perfectly extract the server's model upwards of 312× faster than prior attacks. Motivated by this, we design Muse, an efficient secure inference protocol secure against malicious clients. Muse outperforms existing works by up to 21× and uses up to 3.6× less communication.

DELPHI | *RISELab, UC Berkeley* **September 2018 – September 2019**

Worked under Professor Raluca Ada Popa and Pratyush Mishra on semi-honest secure inference. Through a careful co-design of cryptography, machine learning, and systems, Delphi is up to 100x faster, uses 40x less bandwidth, and scales to networks 10x larger than prior work.

PROJECTS

DELEGATED PROVING | *Efficient Delegation of SNARK Provers — Rust* **September 2020 - Present**

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

POLY-COMMIT | *Multivariate Polynomial Commitment Scheme — Rust* **August 2020 - Present**

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

MUSE | *Client-Malicious Secure Inference — Rust, C++* **September 2019 - Present**

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

DELPHI | *Semi-Honest Secure Inference — 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 | *Genetic Algorithm for Scheduling — Python* **Winter 2019**

- Finds an optimal auditioning schedule for DeCadence A Cappella (Source Code)

SCRYPTO | *Secure File Encryptor/Decryptor — Rust, Python* **Summer 2018**

- Password-protected authenticated file encryption using AES-GCM and PBKDF2 (Source Code)

SECURE FILE STORE (CS161) | *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 | *MITM exploit — Python* **Summer 2016**

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

EXPERIENCE

OPAQUE | *Software Engineer* **Spring 2021 – Present**

- 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

WORKSHOPS

DELPHI | *A Cryptographic Inference Service for Neural Networks*

- CCS Privacy-Preserving Machine Learning in Practice (PPMLP) **November 2020**
- Theory and Practice of Multi-Party Computation (TPMPC) **May 2020**

MUSE | *Secure Inference Resilient to Malicious Clients*

- CCS Privacy-Preserving Machine Learning in Practice (PPMLP) **August 2021**