Ryan Lehmkuhl

https://ryanleh.me
ryanleh@mit.edu

EDUCATION

MIT | *M.S. Electrical Engineering and Computer Science*
Fall 2024

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

AWARDS AND HONORS

- **2022:** MIT Sunlin and Priscilla Chou Fellowship
- **2022:** NSF Graduate Fellowship
- **2021:** NSF Graduate Fellowship Honorable Mention
- **2020:** CRA Outstanding Undergraduate Researcher Finalist
- **2020:** UC Berkeley EECS Outstanding GSI Award
- **2019:** UC Berkeley Summer Undergraduate Research Fellowship
- **2017:** UC Berkeley Regents' and Chancellor's Scholarship

PUBLICATIONS

[1] **Ryan Lehmkuhl** , Alexandra Henzinger, and Henry Corrigan-Gibbs. *"Distributional private information retrieval."* USENIX Security '25.

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

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

[4] 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

Spring/Summer 2022	Code Tenderloin Instructor
Spring 2021	Berkeley ANova Instructor
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

PRIVATELY GATHERING AGGREGATE STATISTICS | *PDOS, MIT* **In Progress**

We are investigating new approaches for privately gathering aggregate statistics that reduce communication and computational overheads compared to existing works.

ATTACKS AGAINST PRIVATE SET MEMBERSHIP PROTOCOLS | *PDOS, MIT* **In Progress**

We show a number of attacks against existing protocols for private set membership such as Chrome's Safe Browsing and Certificate Transparency auditing.

DISTRIBUTIONAL PRIVATE INFORMATION RETRIEVAL | *PDOS, MIT* **USENIX Security '25**

We introduce a new type of private information retrieval (PIR) scheme, distributional PIR, that can run faster than classical PIR by explicitly taking the popularity of database entries into account. On popularity distributions built from real-world data, distributional PIR reduces compute costs by 5–77$\times$ compared to existing techniques.

EOS | *RISELab, UC Berkeley* **USENIX Security '23**

We design a cryptographic protocol for efficient delegation of zero-knowledge, succinct, non-interactive arguments of knowledge (zkSNARKs). In an end-to-end deplyoment, our delegation scheme reduces the prover's computational costs by up to 26× and memory overhead by upwards of 256×.

MUSE | *RISELab, UC Berkeley* **USENIX Security '21**

We demonstrate a devastating attack against many prior secure inference protocols: enabling a malicious client to perfectly extract the server's model upwards of 312× faster than prior attacks. Motivated by this, we design a new secure inference protocol secure against malicious clients. Our protocol outperforms existing alternatives by up to 21× and uses up to 3.6× less communication.

DELPHI | *RISELab, UC Berkeley* **USENIX Security '20**

We introduce a new secure inference protocol that incorporates a careful co-design of cryptography, machine learning, and systems. Our protocol is up to 100× faster, uses 40× less bandwidth, and scales to networks 10× larger than prior work.

CODING

CROWDSURF | *Distributional Private Information Retrieval — Go / C++* **2024**

- Implemented a GPU-accelerated end-to-end system for efficient private information retrieval
- Built a linearly homomorphic encryption with preprocessing scheme using Microsoft's SEAL library

FSS | *Function Secret Sharing — Rust* **2022**

- Implemented several types of function secret sharing schemes

DELEGATED PROVING | *Efficient Delegation of SNARK Provers — Rust* **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 | *Multivariate Polynomial Commitment Scheme — Rust* **2020**

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

MUSE | *Client-Malicious Secure Inference — Rust / C++* **2020**

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

DELPHI | *Semi-Honest Secure Inference — Rust / C++ / Python* **2019**

- Developed new approaches for training convolutional neural networks with polynomial activations
- Built a secure two-party protocol for convolution and matrix multiplication using Microsoft's SEAL library
- Implemented a novel cryptographic protocol and inference engine

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

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

EXPERIENCE

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

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

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

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

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

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