

# Alexandra E. Michael

PhD Student in Hardware Security and Programming Languages  
alexandramichael.com • aemichae@cs.washington.edu • linkedin.com/in/aemichael

## EDUCATION

---

### University of Washington

#### Paul G. Allen School of Computer Science & Engineering

*Ph.D. in Computer Science & Engineering (in progress)*

*M.S. in Computer Science & Engineering*

Advised by David Kohlbrenner and Dan Grossman.

Seattle, WA, USA

September 2022 - Present

June 2024

### University of California, San Diego

*B.S. in Computer Science (GPA: 4.0)*

La Jolla, CA, USA

December 2021

## SELECTED COURSEWORK

---

### University of Washington

- CSE 507: Computer-Aided Reasoning. *Automated reasoning with SAT and SMT solvers.*
- CSE 505: Programming Languages. *Functional programming and proofs with the Rocq (aka Coq) proof assistant.*
- CSE 550: Systems for All. *Systems overview including operating systems, networks, distributed systems, etc.*

### University of California, San Diego

Graduate Computer Security • JIT Security • Compilers • Programming Languages • Software Engineering  
Computer Architecture: A Software Perspective • Operating Systems • Cryptography • Theory of Computation

## PUBLICATIONS

---

### Avoiding Instruction-Centric Microarchitectural Timing Channels Via Binary-Code Transformations.

Michael Flanders, Reshabh K. Sharma, **Alexandra E. Michael**, Dan Grossman, and David Kohlbrenner.

*Intl. Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), April 2024.*

<https://doi.org/10.1145/3620665.3640400>

### MSWasm: Soundly Enforcing Memory-Safe Execution of Unsafe Code.

**Alexandra E. Michael**<sup>\*</sup>, Anitha Gollamudi<sup>\*</sup>, Jay Bosamiya, Evan Johnson, Aidan Denlinger, Craig Disselkoen, Conrad Watt, Bryan Parno, Marco Patrignani, Marco Vassena, and Deian Stefan.

*Symposium on Principles of Programming Languages (POPL), January 2023.*

<https://doi.org/10.1145/3571208>

## RESEARCH EXPERIENCE

---

### University of Washington

*Compiler-based mitigations for hardware side channels*

Seattle, WA, USA

September 2022 - Present

Recent work has found that previously proposed microarchitectural optimizations can open novel side channels, potentially leaking sensitive data to attackers. Some such optimizations are now being implemented in hardware, and we anticipate more appearing as time goes on. We are developing compiler-based approaches to mitigating the resulting vulnerabilities in security-critical code.

---

<sup>\*</sup>Equal contribution

- Wrote binary transformations to mitigate x86\_64 assembly instructions against specific timing side channels.
- Formally verified transformations for semantic equivalence to the original instruction and safety against the considered vulnerabilities, using the Rosette computer-aided reasoning tool.
- Helped build custom LLVM pass to apply transformations when compiling C code to x86\_64 assembly.
- Built evaluation pipeline to compile the libsodium cryptographic library with transformations, test the compiled code for correctness, and assess the run- and compile-time performance cost of the transformations.
- Coauthored and presented a paper on our work so far at ASPLOS 2024.

**University of California, San Diego**  
*Memory-Safe WebAssembly (MSWasm)*

La Jolla, CA, USA  
 September 2019 - September 2022

MSWasm is a memory safety extension to the language WebAssembly (Wasm). MSWasm protects programs compiled to Wasm from unsafe languages like C and C++ from memory safety vulnerabilities in the original source code.

- Worked with multiinstitutional team to build MSWasm, a provably memory-safe WebAssembly extension.
- Implemented MSWasm frontend for GraalVM, a JVM-based engine for running various languages.
- Created and presented a poster and short research paper on implementing the MSWasm compiler in GraalVM.
- Collaborated on LLVM backend for compiling C code to MSWasm, based on LLVM’s existing Wasm backend.
- Compiled subset of WASI Libc and PolyBench-C benchmarks to MSWasm to assess MSWasm’s performance.
- Co-first authored a paper on MSWasm that appeared in Principles of Programming Languages (POPL) 2023.

---

## TEACHING EXPERIENCE

**University of Washington**  
**Paul G. Allen School of Computer Science & Engineering**  
*Teaching Assistant*

Seattle, WA, USA  
 Fall 2024

- CSE 351: The Hardware/Software Interface

Fall 2024

**University of California, San Diego**  
*Undergraduate Teaching Assistant (“CSE Tutor”)*

La Jolla, CA, USA  
 Spring 2019 - Fall 2021

- CSE 20: Discrete Mathematics
- CSE 95: Tutor Apprenticeship
- CSE 105: Theory of Computation

Spring ’19, Fall ’19, Winter ’20, Winter ’21, Fall ’21

Spring 2021

Spring 2020, Fall 2020

- Led course organizational meetings, assigned grading duties, and mentored newer TAs.
- Edited and graded assignments and exams.

- Developed weekly assignments and exercises to achieve learning goals in Tutor Apprentice course.
- Assisted students in office hours and lecture.

**The Harker School**  
*Teaching Assistant—Summer Programming & Advanced Programming*

San Jose, CA, USA  
 Summers 2016 - 2018

---

## WORK EXPERIENCE

**Meta**  
*Software Engineer Intern, WhatsApp Android Platforms*

Menlo Park, CA, USA  
 Summer 2022

- Investigated build speeds for WhatsApp Android app and experimented to identify areas for improvement.
- Designed and implemented proposed changes and assessed for effectiveness in improving build speeds.

*Software Engineer Intern, WhatsApp Business Integrity*

Summer 2021

- Designed and implemented extensible internal API in Haskell and PHP for querying business integrity data.

- Implemented UI tool for querying the API; collaborated with other teams to improve its privacy and usability.
- Presented API and tool to technical and non-technical stakeholders and iterated based on feedback.
- Project became the basis for a full customer-facing feature within a year of the internship.

**Lytix, Inc.**

Software Engineer Intern

San Diego, CA, USA

Summer 2020

- Implemented API endpoints in C# and T-SQL and new frontend features for client-facing web application.

HONORS & AWARDS

---

**NSF Graduate Research Fellowship**

National Science Foundation

March 2023

**Corin Anderson Endowed Fellowship**

University of Washington, Paul G. Allen School of Computer Science & Engineering

September 2022

**Master's Honorable Mention for Excellence in Research**

UC San Diego, Department of Computer Science & Engineering

June 2022

**Undergraduate Award for Excellence in Research**

UC San Diego, Department of Computer Science & Engineering

June 2022

**Summa Cum Laude**

UC San Diego

December 2021

**Tau Beta Pi—The Engineering Honor Society, Member**

Tau Beta Pi, California Psi Chapter

January 2021

**Provost Honors**

UC San Diego

Fall 2018 - Spring 2020, Winter - Spring 2021

SERVICE

---

**Department Service, University of Washington**

Graduate Student Housing Website Maintainer

Spring 2024 - Present

New Grad Mentor

Fall 2023 - Present

Pre-Application Mentorship Service (PAMS) Mentor

Fall 2023 - Present

Programming Languages & Software Engineering (PLSE) Blog Chair

Fall 2023 - Present

Security & Privacy Lab Outreach Co-coordinator

Winter 2023 - Present

PLSE Visit Days Co-coordinator

Winter 2023, 2024

PhD Applications Reader

Fall 2022

**Other Service**

Student Volunteer, Principles of Programming Languages (POPL)

2022, 2023

LANGUAGES & TOOLS

---

C • C++ • LLVM • Assembly • Bash • Rust • Java • Python • WebAssembly  
 Racket • Rosette • Haskell • Rocq\* • JavaScript/TypeScript • Kotlin • HTML • CSS/SASS • Vega-Altair

---

\*Formerly known as Coq