

# CHRISTOPHER W. WAGNER

(480) 376-4303

wagne279@purdue.edu

[azcwagner.github.io](https://azcwagner.github.io)

## RESEARCH INTERESTS

---

My goal is to use formal methods to create provably-correct and reliable software, protocols, and systems. I am interested in distributed systems, formal methods, programming languages, and information security. My current research focuses on modeling and verifying provably-correct distributed systems.

## EDUCATION

---

### Purdue University

PhD Student, Computer Science  
August 2018 - Present  
GPA: 3.95

### Utah State University

Bachelor of Computer Science  
December 2016  
GPA: 4.0

## HONORS AND AWARDS

---

### Professional Activities Grant

### Student Travel Fellowship

### Presidential Scholarship

### Eagle Scout Rank

OOPSLA 2021 (ACM SIGPLAN PAC)

CAV 2019

Utah State University

Boy Scouts of America

## PUBLICATIONS

---

### Peer-Reviewed Conferences

- N. Jaber, S. Jacobs, C. Wagner, M. Kulkarni, and R. Samanta, [Parameterized Verification of Systems with Global Synchronization and Guards](#). In *Computer Aided Verification (CAV)*, 2020.
- N. Jaber, C. Wagner, S. Jacobs, M. Kulkarni, and R. Samanta. QuickSilver: Modeling and Parameterized Verification for Distributed Agreement-Based Systems. In *Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*, 2021.
- Y. Bao, K. Sundararajah, R. Malik, Q. Ye, C. Wagner, N. Jaber, F. Wang, M. Ameri, D. Lu, A. Seto, B. Delaware, R. Samanta, A. Kate, C. Garman, J. Blocki, P. Letourneau, B. Meister, J. Springer, T. Rompf, and M. Kulkarni. HACCLE: Metaprogramming for Secure Multi-Party Computation. In *Generative Programming: Concepts & Experiences (GPCE)*, 2021.

## WORK EXPERIENCE

---

### Purdue University - Teaching Assistant

Jan 2020 - Present

- Assisted with course development in addition to typical TA duties
- Designed homework problems to assess student proficiency in first-order logic, temporal logic, model checking, and abstract interpretation

### Purdue University - Research Assistant

Aug 2018 - Present

- Working on the Discover[i] project for automated parameterized verification and synthesis of distributed systems
- Contributed to the Purdue HACCLE project aiming to increase the usability and performance of secure multi-party computation.
- Led an Abstraction-Guided Program Repair project targeting scalable and sound program repair for real-world programs

**Amazon Web Services - *Applied Scientist Intern***

Summer 2020

- Expanded the functionality of code contracts in CBMC to enable compositional reasoning in proofs involving functions with side effects. This entailed adding a construct to identify the memory locations which may be modified by a function and enforcing that constraint within the function body. This allowed some function calls to be replaced by their contract pre- and post-conditions surrounding a series of havocing statements, eliminating costly exploration of long call stacks.
- Proved functional correctness for portions of Amazon's FreeRTOS and s2n projects using CBMC

**Northrop Grumman - *Associate Software Engineer***

Mar 2017 - Jul 2018

- Engineered RTOS error inducement techniques for PowerPC programs
- Updated software test scripts and documentation for USGS Landsat 9 satellite
- Detected and identified root cause of bugs in embedded flight software

**Utah State University - *Computer Science Tutor***

Aug 2014 - Dec 2016

- Guided CS students in understanding and debugging coding assignments
- Taught problem-solving and programming fundamentals

**Micron Technology - *IT Software Engineer Intern***

Summer 2016

- Supported a scrum agile team in stand-ups, sprint planning, and retrospectives
- Established automated build and testing assets using HP UFT and Jenkins

**Hewlett Packard (Enterprise) - *Software Engineer Intern***

Summer 2015

- Engineered dynamic web apps using ASP.NET Razor
- Patched internal web framework by processing Bugzilla tickets

**RESEARCH PROJECTS**

---

**Discover[i]**

This project combines modular reasoning with parameterized model checking to push the boundaries of automated parameterized reasoning. Our Guarded Synchronization Protocols (GSP) model enables parameterized verification and forms the theoretical foundation for Quicksilver, our higher-level verification system, which supports high-level primitives for abstracting distributed consensus as an atomic program-level construct. In this project, I designed the initial phase-analysis used to generate guards for transitions in the GSP model. I also initiated the lifting of the GSP model's decidable fragment to the semantics of programs written in Mercury, the modeling language of Quicksilver. This allows us to leverage decidability and cutoff results for parameterized verification in the GSP model by performing static checks on Mercury programs, saving the nontrivial effort of translating these programs into the GSP model.

**HACCLE**

This IARPA-funded project aimed to make secure multi-party computation (MPC) more powerful and accessible. We combined MPC techniques such as garbled circuits and fully homomorphic encryption with a layered programming architecture, based on lightweight modular staging, to allow programmers with minimal security background to write secure distributed programs. In this project, my work focused on designing the syntax of Harpoon, our surface language. I also formalized a decomposition of secure functions which are invertible modulo an equivalence relation, which can improve performance by reducing the need for expensive cryptographic operations.

**Abstraction-Guided Program Repair**

The high-level goal of this project was to improve the scalability of fully-automated program repair. I

developed a framework to automatically generate Boolean (abstract) programs from C programs with respect to a set of Boolean predicates with the intention of using the Sketch Synthesis framework to complete undefined expressions in those Boolean programs. By generating abstract programs, and repairing them in their abstract form, we hoped to use details from the abstraction process to inform the construction and concretization of repairs.

## SERVICE AND MENTORING

---

### **Purdue CSGSA (Computer Science Graduate Student Association)**

- Treasurer 2021-2022

### **PurPL (Purdue PL Group)**

- Student Seminar Coordinator, 2018-2019

### **Reviewing**

- Adjunct Reviewer, CAV 2021

### **Mentoring**

- Mentored a visiting Summer 2019 intern on the Abstraction-Guided Program Repair project by building up student's technical background and supervising tool development.
- Student panel member, prospective PhD student visit day, 2019

## TALKS

---

PurPL Seminar: *Bounded Verification of Doubly-Unbounded Parameterized Systems*

PurPL Seminar: *Modeling and Verifying (Multi-)Parameterized Systems*

PurPL Seminar: *Discover[i]* (see Research Projects, above)

PurPL Seminar: *SQLizer: Query Synthesis from Natural Language* by N. Yaghmazadeh et al.

PurPL Seminar: *Modular and Verified Automatic Program Repair* by F. Logozzo and T. Ball

## FOR RECRUITERS

---

### **Languages**

Java, C, C++, C#, Python, Perl, PHP, MySQL, PostgreSQL, PowerPC Assembly  
Coq, Scheme, PostScript, Prolog

### **Tools**

CBMC, Z3, CPAchecker, Sketch, ANTLR, GNU Bison/Yacc  
Git, Subversion, Bugzilla, Jenkins, HP Unified Functional Testing (UFT)  
Amazon EC2, Wind River Simics

### **Work Authorization**

U.S. Citizen