

Steve Zdancewic

Stephan A. Zdancewic, *Curriculum Vitae*

April 29, 2024

stevez@cis.upenn.edu
<http://www.cis.upenn.edu/~stevez>
Tel: 215-898-2661 Fax: 215-898-0587

Department of Computer Science
University of Pennsylvania
Philadelphia, PA 19104

Education

- **Ph.D. Computer Science**
Cornell University, August, 2002.
Dissertation title: *Programming Languages for Information Security*
Advisor: Andrew C. Myers
- **M.S. Computer Science**
Cornell University, August, 2000.
- **B.S. Computer Science and Mathematics**
Carnegie Mellon University, May, 1996.

Employment

- July 2021–present:** University of Pennsylvania
Schlein Family President’s Distinguished Professor of Computer and Information Science
- July 2014–July 2021:** University of Pennsylvania
Professor of Computer and Information Science
- June 2018–Aug. 2019:** Galois
Visiting Scientist (sabbatical)
- July 2008–July 2014:** University of Pennsylvania
Associate Professor of Computer and Information Science
- December 2009–July 2010:** Cambridge Computing Laboratory, UK
Visiting researcher (sabbatical)
- September 2009–December 2009:** Microsoft Research, Cambridge, UK
Visiting researcher (sabbatical)
- July 2002–July 2008:** University of Pennsylvania
Assistant Professor of Computer and Information Science
- June–July 1999:** Lucent Technologies, Bell Labs Innovations
Summer Intern

Research Interests

- **Programming languages:** semantics, type systems, functional programming, logics, concurrency, proof assistants, mechanized metatheory
- **Security:** programming language-based security, information-flow policies, downgrading, authorization logics and policies, auditing mechanisms

Awards

Distinguished Paper Award for “Semantics of Noninterference with Interaction Trees”, ECOOP, 2023
Distinguished Paper Award for “Interaction Trees”, POPL 2020
Christian R. and Mary F. Lindback Foundation Award for Distinguished Teaching, 2018
Micro “Top Picks” paper, 2013 (joint with Santosh Nagarakatte and Milo M. K. Martin)
Alfred P. Sloan Research Fellow, 2009
NSF CAREER Award, 2004
Best paper award at the Symposium on Operating Systems Principles (SOSP), 2001
Intel Foundation Graduate Student Fellowship, 2001
Best paper award at the Conference on Principles, Logics and Implementations of High-level Programming Languages (ICFP/PPDP), 1999
NSF Graduate Student Fellowship, 1996

University Experience & Service

Post Docs Advised

Limin Jia, 2008–2009 (now Associate Research Professor of ECE at CMU)
Benôit Valiron, 2011–2013 (now Assistant Professor at CentraleSupélec)
William Mansky, 2014–2016 (now Assistant Professor of CS at University of Illinois at Chicago)
Christine Rizkallah, 2016–18 (now Lecturer in CSE at University of New South Wales)
Yannick Zakowski, 2018–2020 (now a researcher at Inria)
Lucas Silver, 2023 (now a researcher at Johns Hopkins University Applied Physics Lab)

Ph.D. Students Advised

Stephen Tse, Ph.D. August 2007. *Dynamic Security Policies*
Peng Li, Ph.D. August 2008. *Programmable Concurrency in Pure and Lazy Languages*
Jeff Vaughan, Ph.D. December 2009. *Aura: Programming with Authorization and Audit*
Karl Mazurak, Ph.D. May 2013. *Linear Types, Protocols, and Concurrency in Classical F^o*
Jianzhou Zhao, Ph.D. August 2013. *Formalizing an SSA-Based Compiler for Verified Advanced Program Transformations*
Peter-Michael Osera, Ph.D. August 2016. *Program Synthesis with Types*
Jennifer Paykin, Ph.D. June 2018. *Linear/non-linear types for embedded domain-specific languages*
Robert Rand, Ph.D. December 2018. *Formally Verified Quantum Programming*
Li-yao Xia, Ph.D. August 2022. (cosupervised by Benjamin Pierce) *Executable Denotational Semantics with Interaction Trees*
Yishuai Li, Ph.D. May 2022. (cosupervised by Benjamin Pierce) *Testing by Dualization*
Lucas Silver, Ph.D. August 2023. *Interaction Trees and Formal Specifications*
Irene Yoon, Ph.D. December 2023. *Modular Semantics and Metatheory for LLVM IR*
Calvin Beck
Paul He
Nicholas Rioux
Lawrence Dunn (cosupervised by Val Tannen)
Stephen Mell (cosupervised by Osbert Bastani)
Joey Velez-Ginorio (cosupervised by Konrad Kording)

Masters Students Advised

Solomon Maina, MSE 2020.

Dong-ho Lee, 2018-2019.
Dmitri Garbuzov, MS 2017
David Malley, MSE June 2016. *The French Press Javascript Virtual Machine*
Rohan Shah, MSE June 2014. *Type-Directed Program Synthesis with Record Types*.

Undergraduate Sr. Thesis / Sr. Projects Supervised

2023–2024: Hanxi (Gary) Chen
2022–2023: Seungmin Han and Nathan Hauglund and Aisha Olapade
2021–2022: Alexander Kassouni and Daniel Pfrommer
2018–2019: Olek Gierczak (BAS thesis)
2015–2016: Philip Del Vecchio
2015–2016: Terry Sun and Sam Rossi
2014–2015: Haolin (Kevin) Lu, Fan Yin, Yukuan Zhang
2013–2014: Ceasar Bautista, Adi Dahiya, Kyle Hardgrave, and David Xu
2012–2013: Nate Close, Amalia Hawkins, and Rupi Sureshkumar
2010–2011: Marissa Krupen (EAS 499)
2008–2009: Luke Zarko
2007–2008: Roman Shor
2006–2007: Gerraud Champion, Michael O'Connor
2004–2005: Christopher Low, Steven Richter, Paul Shied
2003–2004: Michael Christensen, Jonathan Jin, Christopher Lam, Corey Pierson
2002–2003: Robert Battle, David Kolas, Matthew Russak

Undergraduate Summer Research Experiences Supervised

Santiago Rodriguez, 2023 (REPL REU)
Lana Semenova, 2023 (REPL REU)
Riely Shahar, 2023 (REPL REU)
Eduardo Gonzalez, 2022
Zakaria Sines, 2022
Nathan Sobotka, 2022
Hanxi (Gary) Chen, 2021, 2022
Caleb Gupta, 2021
Christa Simaan, 2021
Pia Kochar, 2016
Olek Gierczak, 2017

Teaching

- **CIS 120/1200: Programming Languages and Techniques I**
University of Pennsylvania—Fall 2010, 2011, 2012, 2014, 2016, 2017, 2019, 2020, 2021, 2024 Spring 2023
A freshmen level undergraduate course that introduces the basics of programming and computer science.
- **CSE 331: Introduction to Networks and Security**
University of Pennsylvania—Fall 2002, 2003, 2004, 2006
A junior and senior level undergraduate course that introduces the fundamentals of network and computer security, basic cryptographic protocols, and secure system design.
- **CIS 341/3410: Compilers**
University of Pennsylvania—Fall 2008, Spring 2011, 2013, 2015, 2017, 2018, 2020, 2022, 2024
A junior and senior undergraduate level course that introduces compiler design and implementation.

- **CIS 500/5000: Software Foundations**
University of Pennsylvania—Fall 2013, 2022, Spring 2016
A graduate level course that covers formal logic, programming language semantics, and reasoning about software.
- **CIS 551: Computer and Network and Security**
University of Pennsylvania—Spring 2005, 2006, 2007, 2008, 2009, 2012
A graduate level course that covers software, system, and network security, including: buffer-overflow attacks, denial of service attacks, cryptographic protocols, and countermeasures.
- **CIS 670: Advanced Topics in Programming Languages: Safety and Security**
University of Pennsylvania—Spring 2003
A graduate level course on the topics of advanced language design, type systems, and program analyses as they apply to safety and security of software.
- **CIS 670: Advanced Topics in Programming Languages**
University of Pennsylvania—Spring 2021
A graduate level course on the topics of advanced language design, and type systems, concentrating on: polymorphic lambda calculus, parametricity and logical relations, linear types, and modern applications of these ideas in programming languages like Rust.
- **CIS 700: Software and Compiler Verification**
University of Pennsylvania—Fall 2005
A graduate seminar that surveyed the historic and current approaches to verifying compilation, focusing on the programming language and compiler aspects of the problem.
- **CIS 700: π -calculus and the Foundations of Concurrent Systems**
University of Pennsylvania—Spring 2004
Co-taught with Benjamin Pierce
A graduate seminar that introduced Milner’s π -calculus as a tool for studying key features of concurrent systems, including synchronization and message passing.
- **CIS 7000: Compilers**
University of Pennsylvania—Spring 2024
Experimental offering of CIS 3410 as a masters-level course.

University Service

CIS Department Associate Chair (Jan. 2017–)
SEAS QISE committee member (2022–)
Penn Prize Graduate Teaching Awards Committee Chair (2024)
Penn Prize Graduate Teaching Awards Committee (2020, 2021)
Lindback and Provost Teaching Awards Committee (2018–2020)
SEAS Faculty Council (Fall 2015–Fall 2017)
CIS Department Undergraduate Chair (Fall 2010–2014)
SEAS UAC committee (Fall 2010–2014)
CIS Department Undergraduate Curriculum Committee (2008–2018)
Managed three CIS 399 “mini courses” (Python, Unix Skills, and C/C++) taught by graduate students (2005–2008)
CIS department web pages committee (2002–2003)
Led the successful application to have Penn be designated an NSA “Center of Academic Excellence in Information Assurance”, Fall 2002.

Professional Experience & Service

Programming Languages Mentoring Workshop (steering committee chair, 2021–2023)
Chair of IFIP Working Group 2.8 (Functional Programming) (2018–2023)

ACM SIGPLAN John C. Reynolds Dissertation Awards Committee (2018–2021)
 Programming Languages Mentoring Workshop (steering committee co-chair, 2020–2021)
 Programming Languages Mentoring Workshop (@POPL 2018, @POPL 2019, steering committee)
 Editorial Board: Journal of Mathematical Structures in Computer Science (2016–)
 Co-editor of a Special Issue in the Journal of Computer Security on *Computer Security Foundations*
 Summer School Co-organizer: ExCAPE Summer School on Program Synthesis, 2013, 2015
 Co-editor of a Special Issue of the Journal of Computer Security, 2014–2015.
 ACM SIGPLAN Executive Committee Member at Large, 2007–2009
 Steering Committee: Oregon Summer School on Programming Languages, 2005–present
 New Jersey Programming Languages Seminar organizer, 2005–2010
 Tutorial organizer: *Using Proof Assistants for Programming Language Research or, How to write your next POPL paper in Coq* (with B. Pierce and S. Weirich). Jan 2008.
 Co-editor of a Special Issue of the Journal of Information and Computation on *Computer Security: Foundations and Automated Reasoning*, 2007
 Workshop organizer: ACM Workshop on Mechanizing Metatheory (with Pierce and Weirich), 2006, 2007.
 Co-organizer (with David Walker): Summer School on Reliable and Fault Tolerant Software: July, 2005.
 Co-organizer (with Benjamin Pierce, Stephanie Weirich, and Penn graduate students) for the 7th annual ICFP programming competition, 2004.
 Co-organizer (with David Walker): Summer School on Software Security, June 17–25, 2004
 Journal Reviewing: Foundations and Trends in Programming Languages, Journal of Automated Reasoning, Higher Order Symbolic Computation, Journal of Computer Security, Journal of Functional Programming, Information Processing Letters, ACM Transactions on Programming Languages and Systems, ACM Transactions on Computer Systems, IEEE Transactions on Computers, ACM Transactions on Information System Security, Information and Computation
 Conference Reviewing: ACM: ICFP, OOPSLA, POPL, PLDI, SOSP, LCTES; IEEE: Security and Privacy, CSFW; USENIX OSDI; ICALP; ECOOP; FCS; ESOP

Conference and Program Committee Work

- ACM SIGPLAN POPL 2025 General Chair
- Program Committee Associate Chair POPL 2024
- Program Co-Chair: Certified Programs and Proofs (CPP), 2022 & 2023
- Co-organizer of the Programming Languages Mentoring Workshop (PLMW) @ POPL, 2019
- Co-organizer of the Deep Specifications at PLDI Workshop, 2018
- Co-organizer of the Programming Languages Mentoring Workshop (PLMW) @ POPL, 2018
- Program Chair: IEEE Computer Security Foundations Symposium, 2012
- Program Co-Chair: IEEE Computer Security Foundations Symposium, 2011
- Program Co-Chair: Foundations of Computer Security (FCS-ARSPA) 2007
- Program Co-Chair: Foundations of Computer Security (FCS-ARSPA) 2006
- Program Chair: Programming Languages and Analysis for Security (PLAS) 2006
- Program Chair: New Jersey Programming Languages Seminar, September 2003
- Program Committees:
 - Program Committee Member International Conference on Code Quality 2024 (ICCQ24)
 - International Conference on Functional Programming (ICFP) 2021
 - Principles of Programming Languages (POPL) 2020
 - Formal Structures for Computation and Deduction (FSCD) 2020
 - International Workshop on Coq for Programming Languages (CoqPL) 2019
 - Workshop on Software debloating and Delaying (SALAD) 2018
 - Programming Languages Design and Implementation (PLDI) External Review Committee 2018
 - European Symposium on Programming (ESOP 2018)

Certified Programs and Proofs (CPP 2017)
 Programming Languages Design and Implementation (PLDI) External Review Committee 2017
 European Symposium on Programming (ESOP 2017) [guest reviewer]
 Computer Security Foundations (CSF 2016)
 Mathematical Foundations of Program Semantics (MFPS 2016)
 CoqPL 2016
 Implementation and application of functional programming languages (IFL 2015)
 4th Workshop on Synthesis (SYNT 2015)
 OOPSLA External Review Committee 2014
 Programming Languages Design and Implementation (PLDI) External Review Committee 2014
 International Colloquium on Automata, Languages, and Programming (ICALP) 2014
 European Symposium on Programming (ESOP) 2014
 International Symposium on Engineering Secure Software and Systems (ESSOS) 2014
 Implementation and Application of Functional Languages (IFL) 2012
 Programming Language Design and Implementation (PLDI) 2012
 Hot Topics in Security (HotSEC) 2011
 Principles of Programming Languages (POPL) 2011
 European Symposium on Programming (ESOP) 2010
 Theory and Practice of Provenance (TAPP) 2009
 IEEE Symposium on Security & Privacy (Oakland) 2009
 Formal Methods in Security Engineering (FMSE) 2008
 Principles of Programming Languages (POPL) 2008
 IEEE Symposium on Security & Privacy (Oakland) 2008
 Hot Topics in Security (HotSec) 2007
 Workshop on Mechanized Metatheory (WMM) 2007
 Computer Security Foundations Symposium (CSF) 2007
 Mathematical Foundations of Program Semantics (MFPS) 2007
 International Conference on Functional Programming (ICFP) 2006
 Programming Languages and Analysis for Security (PLAS) 2006
 Formal Methods in Security Engineering (FMSE) 2006
 PASSWORD 2006
 Languages, Compilers, and Tools for Embedded Systems (LCTES) 2005
 Computer Security Foundations Workshop (CSFW) 2005
 Mathematical Foundations of Program Semantics (MFPS) 2005

Research Experience and Funding

Grants (bold text indicates those for which I am lead PI)

- **NSF SaTC: CORE: Medium: Secure and Formally-verified Low-level Languages \$1.2M 2023-2027, Steve Zdancewic PI**
- **NSF REU Site: Research Experience for undergraduates in Programming Languages (REPL) \$322,095 2023-2026, Steve Zdancewic PI**
- **ONR "Accountable Protocol Customization" \$7.5M (2019–2024) Boon Thau Loo (Co-PI) Benjamin Pierce (Co-PI) Andre Scedrov (Co-PI) plus others at Stanford and CMU, (Penn's portion is \$2.5M)**

- **ONR “REVOLVER: Recurrent EVOLution and Verification of Encapsulated Rights” \$853,600 (9/2017–2020), N00014-17-1-2930, Jonathan Smith (Co-PI)**
- DARPA: “Synthesizing Data Wranglers.” \$450,000 (2016). (Penn’s portion is \$145k), David Walker (Princeton), Kathleen Fisher (Tufts), Benjamin Pierce (PI), Steve Zdancewic (Co-PI)
- ARL MURI “SynCrypt: Automated Synthesis of Cryptographic Constructions.”, Andre Scedor (PI), Steve Zdancewic (Co-PI) (12/2015–2017)
- ONR MURI “Semantics, Formal Reasoning, and Tool Support for Quantum Programming”, Mike Mislove (Tulane) (PI), (12/2015–2017)
- NSF “Collaborative Research: Expeditions in Computing: The Science of Deep Specification”, NSF-1521539 \$10M (Penn’s portion \$3.35M) (12/15–11/20)
- **NSF CISE SHF Small: Nonstandard Computational Models of Linear Logic. NSF-1421193 Steve Zdancewic (PI). \$500,000 (09/14–09/17)**
- NSF “XPS: CLCCA: Improving Parallel Program Reliability Through Novel Approaches To Precise Data Race Detection”, NSF-1337174. University of Pennsylvania PI: Joseph Devietti, Co-PI Steve Zdancewic and Milo Martin \$700,000 (09/13 – 08/17)
- DARPA “Mission-Oriented Resilient Cloud Program (Mrc)”, University of Pennsylvania PI: Jonathan Smith, Co-PI Steve Zdancewic (and others) \$471,837 (09-11 – 09/15)
- NSF “Expeditions in Computer Augmented Program Engineering: ExCAPE: Harnessing Synthesis for Software Design”, NSF-1064279. University of Pennsylvania PI: Rajeev Alur, Co-PI Steve Zdancewic (and others) \$3.75m (07/11 – 6/14)
- NSF “Watchdog: Hardware-assisted Prevention of All Use-After-Free Security Vulnerabilities”, TC-1116682. University of Pennsylvania PI: Milo Martin, Co-PI: Steve Zdancewic \$500,000 (08/11 – 07/14)
- ONR “IRONCLAD C/C++: Enforcing Memory Safety to Prevent Low-level Security Vulnerabilities”, N00014-11-1-0596. University of Pennsylvania PI: Milo Martin, Co-PI: Steve Zdancewic \$349,000 (04/11 – 04/14)
- **NSF “Validating Program Transformations in a Mechanized LLVM”, CCF-1065166. University of Pennsylvania PI: Steve Zdancewic \$808,961 (07/11 – 07/15)**
- IARPA “Quantum Computer Science Program”, Sub on proposal in resp. to IARPA BAA-10-02. University of Pennsylvania PI: Jonathan M. Smith, Co-PI: Steve Zdancewic \$508,318 total DC&IC. (1/11 – 06/13)
- **NSF “Practical Linear Types for Safe Protocols” CCF-1017027. University of Pennsylvania PI: Steve Zdancewic. \$500,000 (09/10 – 09/13)**
- ONR Networks Opposing Botnets, PI: Jonathan Smith with B. Pierce, S. Zdancewic, B. Loo, S. Weirich (Penn) E. Felton, J. Rexford, D. Walker (Princeton) G. Morrisett, M. Welsh (Harvard), 2009-2012
- DARPA CS Study Group Phase 2 “Machine-checked Metatheory for Security-oriented Languages”. University of Pennsylvania PI: Stephanie Weirich, Co-PI: Steve Zdancewic. \$500,000 (5/08 – 4/10)
- NSF “CT-T: Collaborative Research: Manifest Security” CCF-0716469. University of Pennsylvania PI: Benjamin Pierce, Co-PIs: Stephanie Weirich, Steve Zdancewic. Carnegie Mellon University PI: Frank Pfennig, Co-PIs: Karl Cray, Robert Harper. \$1M (9/07 – 9/09)
- **NSF “CCF: Unifying Events and Threads: Language Support for Network Services” CCF-0541040. University of Pennsylvania PI: Steve Zdancewic. \$350,000 (8/06 – 7/09)**

- NSF “CRI: Machine Assistance for Programming Language Research” CNS-0551589. University of Pennsylvania PI: Stephanie Weirich, Co-PIs: Benjamin Pierce, Steve Zdancewic. \$200,000 (3/06 – 2/08)
- NSF “CT-T: Collaborative Research: Flexible, Decentralized Information-flow Control for Dynamic Environments” CCF-0524035. University of Pennsylvania PI: Steve Zdancewic, University of Maryland, College Park PI: Michael Hicks, Pennsylvania State University PI: Patrick McDaniel, University of Texas, San Antonio PI: William Winsborough. \$1.2M (9/05 – 8/08)
- NSF “CT-T: Resource-Guided Implementation of Secure Embedded Software” CNS-0524059. University of Pennsylvania PI: Steve Zdancewic, Co-PIs: Rajeev Alur, Andre Scedrov. \$1M (8/05 – 8/07)
- NSF “CAREER: Language-based Distributed System Security.” CNS-0346939. University of Pennsylvania PI: Steve Zdancewic. \$400,000 (6/04 – 5/09)
- NSF “Dynamic Security Policies.” CCR-0311204. University of Pennsylvania PI: Steve Zdancewic. \$300,000 (8/03 – 7/05)
- NSF “Software Security: Theory to Practice.” CCF-0438714. University of Pennsylvania PI: Steve Zdancewic, University of Oregon, PI: Zena Ariola, Princeton University, PI: David Walker. \$10,000 (8/04 – 1/05)

Invited Talks and Technical Presentations

1. *Vellom: Verifying LLVM IR Code*, Reed College, 5 Mar. 2024
2. *Vellom: Verifying LLVM IR Code*, University of Illinois at Chicago, 11 Dec. 2020
3. *Termination-Sensitive Program Specifications (Reasoning About Interaction Trees)*, IFIP Working Group 2.8 (Functional Programming), 9–13 March 2020
4. *Vellom: Verifying LLVM IR Code*, School of Computer and Cyber Sciences Colloquium, Augusta University, 14 Feb. 2020
5. *Interaction Trees: Representing Recursive and Impure Programs in Coq*, DeepSpec Workshop Invited Talk, 23 June 2019
6. *Compositional Compiler Correctness in Coq*, IFIP Working Group 2.8 (Functional Programming), 20–24 May 2019
7. *Verified Software Correctness - The Science of Deep Specification*, University of Pennsylvania Lindback Lecture for the Philomathean Society, 21 Mar. 2019
8. *Vellom – Verifying the LLVM*. StrangeLoop. St. Louis, MO. 28 Sept. 2018.
9. *Vellom – Verifying the LLVM*. Dagstuhl Seminar on Secure Compilation. Wadern, Germany. 15 May 2018.
10. *Vellom – Verifying the LLVM*. 4th International Conference on Tools and Methods of Program Analysis, TMPA-2017, (Keynote) Moscow, Russia. 3–4 Mar. 2017,
11. *Vellom II: Semantics and Verification for LLVM*, Principles in Practice (PiP), 21 Jan. 2017
12. *SSA \subseteq CBPV* IFIP Working Group 2.8 (Functional Programming), 17–22 July. 2016.
13. *Type- and Example-Driven Program Synthesis*. Symp. on TFP (Keynote) 8–10 Jun. 2016.
14. *Vellom: A Verified LLVM*. At the Mathematical Foundations of Program Semantics, 23–26 May 2016.
15. *Curry-Howard for GUIs: classical linear temporal logic* IFIP Working Group 2.8, 14–29 May. 2015.
16. *Vellom: Verifying Safety in the LLVM IR* Max Planck Institute, 9 Oct. 2014.
17. *Vellom: Verifying Transformations of the LLVM IR* Reliably Secure Software Systems (RS³) Annual Meeting Keynote Talk, 8 Oct. 2014
18. *Type- and Example-Driven Program Synthesis* Chalmers University, 6 Oct. 2014.
19. *Type- and Example-Driven Program Synthesis* IFIP Working Group 2.8, 12 Aug. 2014.
20. *Vellom: Verifying Transformations of the LLVM IR* Certification of High- and Low-level Programs Workshop, 7–10 July 2014.
21. *Vellom: Verifying Transformations of the LLVM IR* IFIP Working Group 2.8, 14 Oct. 2013.

22. *Vellom: Verifying Transformations of the LLVM IR Syntax and Semantics of Low-Level Languages (LOLA)*, 29 Jun. 2013.
23. *Linear Logic and Linear Algebra*. IFIP Working Group 2.8, 1 Nov. 2012.
24. *Mechanized Verification of Computing Dominators for Formalizing Compilers*. Certified Programs and Proofs. 13 Dec. 2012.
25. *Why Information-flow is Different From—and harder than—Verifying other kinds of Properties* NSF/CCC Workshop about Semiconductor Verification. 15 Jan. 2013.
26. *Work–Life Balance for Computer Scientists*. Programming Languages Mentoring Workshop , Philadelphia, PA. 24 Jan. 2012
27. *Protocol Types in a Classical Linear Logic*, Chalmers University Security Workshop, Gothenburg, Sweden. 29 Aug. 2011.
28. *Protocol Types in a Classical Linear Logic*, Mathematical Foundations of Program Semantics (MFPS) , Pittsburgh, PA. 27 May 2011.
29. *Lightweight Linear F*, School of Informatics, University of Edinburgh, Scotland. 20 Jul. 2010.
30. *AURA: A programming language with authorization and audit*, INRIA . 14 Jun. 2010.
31. *AURA: A programming language with authorization and audit*, Strathclyde University. 29 Apr. 2010.
32. *AURA: A programming language with authorization and audit*, Cambridge Computing Laboratory Wednesday Seminar. 10 Mar. 2010.
33. *F^o: Lightweight Linear F*, Cambridge Computing Labs Semantics Lunch. 12 Oct. 2009.
34. *AURA: A programming language with authorization and audit*, CUNY CS Colloquium. 12 Mar. 2009.
35. *AURA: A programming language with authorization and audit*, University of Iowa CS Colloquium. 21 Nov. 2008.
36. *AURA: A programming language with authorization and audit*, IFIP Working Group 2.8 (Functional Programming). Park City, Utah. 18 June 2008.
37. *AURA: A programming language with authorization and audit*, High Confidence Software and Systems Conference (HCSS). Maritime Institute. Linthicum, Maryland. 7 Mar. 2008
38. *Application-level concurrency in Haskell: Combining Events and Threads*, Cornell University CS Colloquium. Ithaca, NY. 11 Oct. 2007.
39. *Combining Access Control and Information Flow in DCC*, IFIP Working Group 2.8 (Functional Programming). Iceland. 17 Jul. 2007.
40. *Combining Access Control and Information Flow in DCC*, Dagstuhl Seminar on Mobility, Ubiquity and Security. Wadern, Germany. 26 Feb. 2007.
41. *Application-level Concurrency: Combining Events and Threads*, Declarative Aspects of Multicore Programming (DAMP). Nice, France. 16 Jan. 2007.
42. *Dynamic Information-Flow Policies in Java 5*, IBM T.J. Watson Research Center. 10 Jan. 2007
43. *Encoding Information Flow in Haskell*, IFIP Working Group 2.8 (Functional Programming). Boston, Massachusetts. 17 Jul. 2006.
44. *Encoding Information Flow in Haskell*, Mathematical Foundations of Program Semantics (MFPS) . Genoa, Italy. 26 May 2006.
45. *Encoding Information Flow in Haskell*, Department of CS, Chalmers University 22 May 2006.
46. *Language-based Information Security*, Computer Information Assurance and Security Invited Talk Series. University of Texas, San Antonio. 16 Dec. 2005.
47. *Deriving Noninterference Results from Parametricity*, Mathematical Foundations of Program Semantics (MFPS). Birmingham, England. 19 May 2005.
48. *Language Based Security and Secure Program Partitioning*, Department of Computer Science Seminar. University of Massachusetts, Amherst, Massachusetts. 15 Nov. 2004.
49. Keynote address: *Programming Language Tools for Security*, First ACM Workshop on Business Driven Security Engineering (BIZSEC), Fairfax Virginia. 31 Oct.2003.
50. *Dynamic Principals and the Decentralized Label Model*, Dagstuhl Seminar on Language-based Security, Schloss Dagstuhl, Germany. 6 Oct. 2003.
51. *Jif and Secure Program Partitioning*, City University of NY Graduate Center. 25 Sept. 2003.

52. *Information Flow Security*, Stevens Institute of Technology Laboratory for Secure Systems Seminar, 10 April 2003.
53. *Programming Languages for Information Security*, [A job interview talk] presented at: Oregon Graduate Institute, University of Oregon, University of Pennsylvania, University of Virginia, Northeastern University, Harvard University, University of California, San Diego, University of Michigan, Purdue University, University of Texas at Austin, Rice University, University of Maryland, MIT, University of Colorado, Boulder, Pennsylvania State University.
54. *A Syntactic Account of Type Abstraction*, CMU's Principles of Programming seminar, 6 Nov. 1998.

Lecture Series / Invited Workshop Talks

1. *Formal Verification of Monadic Computations*, Summer School on Types, Logic and Verification, University of Oregon, Eugene Oregon. July 2022.
2. *Vellvm: Verifying the LLVM*, DeepSpec Summer School, University of Pennsylvania, Philadelphia Pennsylvania July 2017.
3. *Verifying LLVM Optimizations in Coq*, Summer School on Types, Logic and Verification, University of Oregon, Eugene Oregon. August 2013.
4. *Language-based Security*, International School on Foundations of Security Analysis and Design (FOSAD), Bertinoro, Italy. August 2008.
5. *Three Lectures on Stack Inspection and the Java Security model*, Summer School on Software Security: Theory to Practice, University of Oregon, Eugene Oregon. June 2004.
6. *Three Lectures on Language-based Information Flow Security*, Summer School on Foundations of Security, University of Oregon, Eugene Oregon. 24 & 25 June 2003.

Open Source Software

1. The Vellvm - Verified LLVM IR project <https://github.com/vellvm/vellvm>
2. Interaction Trees Library <https://github.com/DeepSpec/InteractionTrees>

Publications

Journal Papers

1. Stephen Mell, Steve Zdancewic, and Osbert Bastani. Optimal program synthesis via abstract interpretation. 8(POPL), 2024.
2. Nick Rioux, Xuejing Huang, Bruno C. d. S. Oliviera, and Steve Zdancewic. A Bowtie for a Beast: Overloading, Eta Expansion, and Extensible Data Types in $F\bowtie$. *Proceedings of the ACM on Programming Languages*, 7(POPL), 2023.
3. Nicolas Chappe, Paul He, Ludovic Henrio, Yannick Zakowski, and Steve Zdancewic. Choice trees: Representing nondeterministic, recursive, and impure programs in coq. *Proceedings of the ACM on Programming Languages*, 7(POPL), 2023.
4. Irene Yoon, Yannick Zakowski, and Steve Zdancewic. Formal reasoning about layered monadic interpreters. *Proceedings of the ACM on Programming Languages*, 6(ICFP), 2022.

5. Mohsen Lesani, Li-Yao Xia, Anders Kaseorg, Christian J. Bell, Adam Chlipala, Benjamin C. Pierce, and Steve Zdancewic. C4: Verified transactional objects. *Proceedings of the ACM on Programming Languages*, OOPSLA, 2022.
6. Paul He, Eddy Westbrook, Brent Carmer, Chris Phifer, Valentin Robert, Karl Smeltzer, Andrei Andrei Ștefănescu, Aaron Tomb, Adam Wick, Matthew Yacavone, and Steve Zdancewic. A type system for extracting functional specifications from memory-safe imperative programs. *Proceedings of the ACM on Programming Languages*, OOPSLA, 2021.
7. Yannick Zakowski, Calvin Beck, Irene Yoon, Ilya Zaichuk, Vadim Zaliva, and Steve Zdancewic. Modular, compositional, and executable formal semantics for llvm ir. *Proceedings of the ACM on Programming Languages*, 5(ICFP), 2021.
8. Lucas Silver and Steve Zdancewic. Dijkstra monads forever: Termination-sensitive specifications for interaction trees. *Proceedings of the ACM on Programming Languages*, 5(POPL), January 2021.
9. Nick Rioux and Steve Zdancewic. Computation focusing. *Proceedings of the ACM on Programming Languages*, 5(ICFP), 2020.
10. Li-yao Xia, Yannick Zakowski, Paul He, Chung-Kil Hur, Gregory Malecha, Benjamin C. Pierce, and Steve Zdancewic. Interaction trees. *Proceedings of the ACM on Programming Languages*, 4(POPL), January 2020.
11. Anders Miltner, Solomon Maina, Kathleen Fisher, Benjamin C. Pierce, David Walker, and Steve Zdancewic. Synthesizing symmetric lenses. *Proceedings of the ACM on Programming Languages*, 3(ICFP), 2019.
12. Solomon Maina, Anders Miltner, Kathleen Fisher, Benjamin C. Pierce, David Walker, and Steve Zdancewic. Synthesizing quotient lenses. *Proceedings of the ACM on Programming Languages*, 2(ICFP), 2018.
13. Anders Miltner, Kathleen Fisher, Benjamin C. Pierce, David Walker, and Steve Zdancewic. Synthesizing Bijective Lenses. *Proceedings of the ACM on Programming Languages*, 2(POPL), January 2018.
14. Andrew W. Appel, Lennart Beringer, Adam Chlipala, Benjamin C. Pierce, Zhong Shao, Stephanie Weirich, and Steve Zdancewic. Position paper: The Science of Deep Specification. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 375(2104), 2017.
15. B. Valiron and S. Zdancewic. Modeling simply-typed lambda calculi in the category of finite vector spaces. *Scientific Annals of Computer Science*, 24(2):325–368, 2014.
16. Peng Li and Steve Zdancewic. Arrows for Secure Information Flow. *Theoretical Computer Science*, 411(19):1974–1994, 2010.
17. Jay Ligatti, David Walker, and Steve Zdancewic. A type-theoretic interpretation of pointcuts and advice. *Science of Computer Programming: Special Issue on Foundations of Aspect-Oriented Programming*, pages 240–266, 2006.
18. Andrew C. Myers, Andrei Sabelfeld, and Steve Zdancewic. Enforcing robust declassification and qualified robustness. *Journal of Computer Security*, 14(2):157–196, 2006.
19. Stephen Tse and Steve Zdancewic. Run-time principals in information-flow type systems. *Transactions on Programming Languages and Systems*, 30(1):6, 2008.
20. Steve Zdancewic and Andrew C. Myers. Secure Information Flow via Linear Continuations. *Higher Order and Symbolic Computation*, 15(2/3):209–234, 2002.
21. Steve Zdancewic, Lantian Zheng, Nathaniel Nystrom, and Andrew C. Myers. Secure Program Partitioning. *Transactions on Computer Systems*, 20(3):283–328, 2002.

22. Dan Grossman, Greg Morrisett, and Steve Zdancewic. Syntactic Type Abstraction. *Transactions on Programming Languages and Systems*, 22(6):1037–1080, November 2000.

Volumes Edited

1. Pierpaolo Degano, Ralf Küsters, Luca Viganò, and Steve Zdancewic, editors. *Joint workshop on foundations of computer security and automated reasoning for security protocol analysis (FCS-ARSPA '06)*, volume 206 of *Information and Computation*. Elsevier, 2008.
2. Vugranam C. Shreedhar and Steve Zdancewic, editors. *Proceedings of the 2006 Workshop on Programming Languages and Analysis for Security (PLAS)*. ACM, 2006.

Highly Selective Conference and Workshop Papers

1. Lawrence Dunn, Val Tannen, and Steve Zdancewic. Tealeaves: Structured Monads for Generic First-Order Abstract Syntax Infrastructure. In *14th International Conference on Interactive Theorem Proving (ITP)*, 2023.
2. Lawrence Dunn, Val Tannen, and Steve Zdancewic. Syntax monads for the working formal metatheorist. In *Proceedings of the 6th International Conference on Applied Category Theory (ACT)*, 2023.
3. Stephen Mell, Favyen Bastani, Steve Zdancewic, and Osbert Bastani. Synthesizing trajectory queries from examples. In *Computer Aided Verification - 35th International Conference, CAV, 2023*.
4. Lucas Silver, Paul He, Ethan Cecchetti, Andrew K. Hirsch, and Steve Zdancewic. Semantics for noninterference with interaction trees. In *Proceedings of the 37th Annual European Conference on Object-Oriented Programming (ECOOP 2023)*, 2023.
5. Stephen Mell, Osbert Bastani, and Steve Zdancewic. Ideograph: A language for expressing and manipulating structured data. In *Proceedings Twelfth International Workshop on Computing with Terms and Graphs (TERMGRAPH 2022)*, pages 65–84. Electronic Proceedings in Theoretical Computer Science, 2022.
6. George Tolkachev, Stephen Mell, Steve Zdancewic, and Osbert Bastani. Counterfactual explanations for natural language interfaces. In *60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022. (short paper).
7. Yishuai Li, Benjamin Pierce, and Steve Zdancewic. Model-based testing of networked applications. In *The ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)*, 2021.
8. Hengchu Zhang, Wolf Honoré, Nicolas Koh, Yao Li, Yishuai Li, Li-Yao Xia, Lennart Beringer, William Mansky, Benjamin Pierce, and Steve Zdancewic. Verifying an HTTP Key-Value Server with Interaction Trees and VST. In Liron Cohen and Cezary Kaliszyk, editors, *12th International Conference on Interactive Theorem Proving (ITP 2021)*, volume 193 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
9. Yannick Zakowski, Paul He, Chung-Kil Hur, and Steve Zdancewic. An equational theory for weak bisimulation via generalized parameterized coinduction. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP)*, January 2020.
10. Jennifer Paykin and Steve Zdancewic. A HoTT Quantum Equational Theory. In *The 16th International Conference on Quantum Physics and Logic (QPL)*, 2019. extended version available on arXiv.
11. Marcella Hastings, Brett Hemenway, Daniel Noble, and Steve Zdancewic. SoK: General Purpose Compilers for Secure Multi-Party Computation. In *IEEE 2019 Symposium on Security and Privacy (Oakland)*, 2019.

12. Nicolas Koh, Yao Li, Yishuai Li, Li yao Xia, Lennart Beringer, Wolf Honoré, William Mansky, Benjamin C. Pierce, and Steve Zdancewic. From C to Interaction Trees: Specifying, Verifying, and Testing a Networked Server. In *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP)*, January 2019.
13. Robert Rand, Jennifer Paykin, Dong-Ho Lee, and Steve Zdancewic. REQUIRE: Reasoning about reversible quantum circuits. In *The 15th International Conference on Quantum Physics and Logic (QPL)*, 2018.
14. Christine Rizkallah, Dmitri Garbuzov, and Steve Zdancewic. A Formal Equational Theory for Call-By-Push-Value. In *9th International Conference on Interactive Theorem Proving (ITP)*, 2018.
15. Jennifer Paykin and Steve Zdancewic. The Linearity Monad. In *Proceedings of the 10th ACM SIGPLAN International Haskell Symposium*, 2017.
16. Robert Rand, Jennifer Paykin, and Steve Zdancewic. QWIRE Practice: Formal Verification of Quantum Circuits in Coq. In *The 14th International Conference on Quantum Physics and Logic (QPL)*, 2017.
17. William Mansky, Yuanfeng Peng, Steve Zdancewic, and Joseph Devietti. Verifying dynamic race detection. In *The 6th ACM SIGPLAN Conference on Certified Programs and Proofs (CPP 2017)*, 2017.
18. Jennifer Paykin, Robert Rand, and Steve Zdancewic. QWire: A Core Language for Quantum Circuits. In *Proc. of the ACM Symposium on Principles of Programming Languages (POPL)*, 2017.
19. Jennifer Paykin and Steve Zdancewic. Linear $\lambda\mu$ is cp (more or less). In *A List of Successes to Change the World (Wadlerfest)*, 2016.
20. Jonathan Frankle, Peter-Michael Osera, David Walker, and Steve Zdancewic. Example-directed synthesis: A type-theoretic interpretation. In *Proc. of the ACM Symposium on Principles of Programming Languages (POPL)*, 2016.
21. Robert Rand and Steve Zdancewic. VPHL: A Verified Partial-Correctness Logic for Probabilistic Programs. In *Mathematical Foundations of Program Semantics (MFPS)*, 2015.
22. Santosh Nagarakatte, Milo M. K. Martin, and Steve Zdancewic. Everything you want to know about pointer-based checking. In *1st Summit on Advances in Programming Languages, SNAPL 2015, May 3-6, 2015, Asilomar, California, USA*, pages 190–208, 2015.
23. William Mansky, Dmitri Garbuzov, and Steve Zdancewic. An axiomatic specification for sequential memory models. In *Computer Aided Verification - 27th International Conference, CAV 2015*, 2015.
24. Jeehoon Kang, Chung-Kil Hur, William Mansky, Dmitri Garbuzov, Steve Zdancewic, and Viktor Vafeiadis. A formal C memory model supporting integer-pointer casts. In *Proc. 2015 ACM SIGPLAN Conference on Programming Languages Design and Implementation (PLDI)*, 2015.
25. Peter-Michael Osera and Steve Zdancewic. Type-and-Example-Directed Program Synthesis. In *Proc. 2015 ACM SIGPLAN Conference on Programming Languages Design and Implementation (PLDI)*, 2015.
26. Benoît Valiron and Steve Zdancewic. Finite vector spaces as model of simply-typed lambda-calculi. In *Proceedings of the 11th International Colloquium on Theoretical Aspects of Computing (ICTAC 14)*, 2014.
27. Aloïs Brunel, Marco Gaboardi, Damiano Mazza, and Steve Zdancewic. A core quantitative coefficient calculus. In *Proc. of the 23rd European Symposium on Programming (ESOP)*, volume 8410, pages 351–370, 2014.
28. Santosh Nagarakatte, Milo M. K. Martin, and Steve Zdancewic. Watchdoglite: Hardware-accelerated compiler-based pointer checking. In *Proceedings of Annual IEEE/ACM International Symposium on Code Generation and Optimization, CGO '14*, pages 175:175–175:184. ACM, 2014.

29. Christian DeLozier, Richard Eisenberg, Santosh Nagarakatte, Peter-Michael Osera, Milo M. K. Martin, and Steve Zdancewic. Ironclad C++: A library-augmented type-safe subset of C++. In *Proceedings of the 28th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, (OOPSLA)*, 2013.
30. Santosh Nagarakatte, Milo M K Martin, and Steve Zdancewic. Hardware-enforced comprehensive memory safety. *IEEE MICRO's "Top Picks of Architecture Conferences of 2012" Issue (Micro Top Picks'2013)*, May/June 2013.
31. Jianzhou Zhao, Santosh Nagarakatte, Milo M. K. Martin, and Steve Zdancewic. Formal verification of SSA-based optimizations for LLVM. In *Proc. 2013 ACM SIGPLAN Conference on Programming Languages Design and Implementation (PLDI)*, 2013.
32. Jianzhou Zhao and Steve Zdancewic. Mechanized verification of computing dominators for formalizing compilers. In *The Second International Conference on Certified Programs and Proofs (CPP)*, Lecture Notes in Computer Science, pages 27–42, 2012.
33. Santosh Nagarakatte, Milo M. K. Martin, and Steve Zdancewic. Watchdog: Hardware for safe and secure manual memory management and full memory safety. In *Proceedings of the 39th International Symposium on Computer Architecture (ISCA)*, June 2012.
34. Jianzhou Zhao, Santosh Nagarakatte, Milo M. K. Martin, and Steve Zdancewic. Formalizing the LLVM Intermediate Representation for Verified Program Transformations. In *Proc. of the ACM Symposium on Principles of Programming Languages (POPL)*, 2012.
35. Stephanie Weirich, Dimitrios Vytiniotis, Simon Peyton Jones, and Steve Zdancewic. Generative type abstraction and type-level computation. In *Proc. of the ACM Symposium on Principles of Programming Languages (POPL)*, 2011.
36. Jianzhou Zhao, Qi Zhang, and Steve Zdancewic. Relational parametricity for polymorphic linear lambda calculus. In *Proceedings of the Eighth ASIAN Symposium on Programming Languages and Systems (APLAS)*, 2010.
37. Santosh Nagarakatte, Jianzhou Zhao, Milo M. K. Martin, and Steve Zdancewic. CETS: Compiler-enforced temporal safety for c. In *Proceedings of the ACM International Symposium on Memory Management (ISMM)*, 2010.
38. Karl Mazurak and Steve Zdancewic. Lollipop: to Concurrency from Classical Linear Logic via Curry-Howard and Control. In *Proc. of the 15th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, 2010.
39. Aaron Bohannon, Benjamin C. Pierce, Vilhelm Sjöberg, Stephanie Weirich, and Steve Zdancewic. Reactive noninterference. In *ACM Computer and Communications Security Conference (CCS)*, 2009.
40. J. Nathan Foster, Benjamin C. Pierce, and Steve Zdancewic. Updatable security views. In *Proc. of 22nd IEEE Computer Security Foundations Symposium (CSF)*, 2009.
41. Santosh Nagarakatte, Jianzhou Zhao, Milo M. K. Martin, and Steve Zdancewic. SoftBound: Highly Compatible and Complete Spatial Memory Safety for C. In *Proc. 2009 ACM SIGPLAN Conference on Programming Languages Design and Implementation (PLDI)*, 2009.
42. Limin Jia, Jeffrey A. Vaughan, Karl Mazurak, Jianzhou Zhao, Luke Zarko, Joseph Schorr, and Steve Zdancewic. AURA: A programming language for authorization and audit. In *Proc. of the 13th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, Victoria, British Columbia, Canada, September 2008.

43. Jeffrey A. Vaughan, Limin Jia, Karl Mazurak, and Steve Zdancewic. Evidence-based audit. In *Proc. of 21st IEEE Computer Security Foundations Symposium (CSF)*, pages 177–191. IEEE Computer Society Press, 2008.
44. Joe Devietti, Colin Blundell, Milo M.K. Martin, and Steve Zdancewic. Hardbound: Architectural support for spatial safety of the c programming language. In *International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, March 2008.
45. Jeffrey A. Vaughan and Steve Zdancewic. A cryptographic decentralized label model. In *IEEE 2007 Symposium on Security and Privacy (Oakland)*, pages 192–206, 2007.
46. Peng Li and Steve Zdancewic. Combining events and threads for scalable network services. In *Proc. 2007 ACM SIGPLAN Conference on Programming Languages Design and Implementation (PLDI)*, pages 189–199, 2007.
47. Rajeev Alur, Pavol Černý, and Steve Zdancewic. Preserving secrecy under refinement. In *Proc. of 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 107–118, 2006.
48. Peng Li and Steve Zdancewic. Encoding information flow in Haskell. In *Proc. of 19th IEEE Computer Security Foundations Workshop (CSFW)*, pages 16–27. IEEE Computer Society Press, 2006.
49. Nikhil Swamy, Michael Hicks, Stephen Tse, and Steve Zdancewic. Managing policy updates in security-typed languages. In *Proc. of 19th IEEE Computer Security Foundations Workshop (CSFW)*, pages 202–216. IEEE Computer Society Press, 2006.
50. Brian E. Aydemir, Aaron Bohannon, Matthew Fairbairn, J. Nathan Foster, Benjamin C. Pierce, Peter Sewell, Dimitrios Vytiniotis, Geoffrey Washburn, Stephanie Weirich, and Steve Zdancewic. Mechanized Metatheory for the Masses: The POPLMark Challenge. In *International Conference on Theorem Proving in Higher Order Logics (TPHOLs)*, pages 50–65, 2005.
51. Stephen Tse and Steve Zdancewic. Designing a Security-typed Language with Certificate-based De-classification. In *Proc. of the 14th European Symposium on Programming (ESOP)*, volume 3444, pages 279–294, 2005.
52. Peng Li and Steve Zdancewic. Practical Information-flow Control in Web-based Information Systems. In *Proc. of 18th IEEE Computer Security Foundations Workshop (CSFW)*, pages 2–15, 2005.
53. Peng Li and Steve Zdancewic. Downgrading Policies and Relaxed Noninterference. In *Proc. 32nd ACM Symp. on Principles of Programming Languages (POPL)*, pages 158–170, January 2005.
54. Andrew C. Myers, Andrei Sabelfeld, and Steve Zdancewic. Enforcing robust declassification. In *Proc. of 17th IEEE Computer Security Foundations Workshop (CSFW)*, pages 172–186, 2004.
55. Peng Li and Steve Zdancewic. Advanced Control Flow in Java Card Programming. In *Proceedings of the 2004 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES)*, pages 165–174, June 2004.
56. Stephen Tse and Steve Zdancewic. Run-time Principals in Information-flow Type Systems. In *IEEE 2004 Symposium on Security and Privacy (Oakland)*, pages 179–193. IEEE Computer Society Press, May 2004.
57. David Walker, Steve Zdancewic, and Jay Ligatti. A Theory of Aspects. In *Proc. of the 8th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, pages 127–139, Upsala, Sweden, August 2003.
58. Steve Zdancewic and Andrew C. Myers. Observational Determinism for Concurrent Program Security. In *Proc. of 16th IEEE Computer Security Foundations Workshop (CSFW)*, pages 29–45, Asilomar, CA, July 2003.

59. Lantian Zheng, Stephen Chong, Steve Zdancewic, and Andrew C. Myers. Building Secure Distributed Systems Using Replication and Partitioning. In *IEEE 2003 Symposium on Security and Privacy (Oakland)*, pages 236–250. IEEE Computer Society Press, 2003.
60. Steve Zdancewic, Lantian Zheng, Nathaniel Nystrom, and Andrew C. Myers. Untrusted Hosts and Confidentiality: Secure Program Partitioning. In *Proc. 18th ACM Symp. on Operating System Principles (SOSP)*, volume 35(5) of *Operating Systems Review*, pages 1–14, Banff, Canada, October 2001.
61. Steve Zdancewic and Andrew C. Myers. Robust Declassification. In *Proc. of 14th IEEE Computer Security Foundations Workshop (CSFW)*, pages 15–23, Cape Breton, Canada, June 2001.
62. Steve Zdancewic and Andrew C. Myers. Secure Information Flow and CPS. In *Proc. of the 10th European Symposium on Programming (ESOP)*, volume 2028 of *Lecture Notes in Computer Science*, pages 46–61, April 2001.
63. Steve Zdancewic, Dan Grossman, and Greg Morrisett. Principals in Programming Languages: A Syntactic Proof Technique. In *Proc. of the 4th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, pages 197–207, Paris, France, September 1999.
64. Stephen Tse and Steve Zdancewic. Translating Dependency into Parametricity. In *Proc. of the 9th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, 2004.

Invited Papers

1. Steve Zdancewic. Challenges for Information-flow Security. In *Proceedings of the 1st International Workshop on the Programming Language Interference and Dependence (PLID'04)*, 2004. (5 pages).
2. Steve Zdancewic. A Type System for Robust Declassification. In *Proceedings of the Nineteenth Conference on the Mathematical Foundations of Programming Semantics (MFPS)*. Electronic Notes in Theoretical Computer Science, March 2003. (16 pages).

Refereed Workshop Papers

1. Neelakantan R. Krishnaswami, Jennifer Paykin, and Steve Zdancewic. Curry-howard for guis. In *POPL Off the Beaten Track (OBT)*, 2015.
2. Jennifer Paykin and Steve Zdancewic. A linear/producer/consumer model of classical linear logic (extended abstract). In *Third International Workshop on Linearity, LINEARITY*, 2014.
3. Peter-Michael Osera, Vilhelm Sjöberg, and Steve Zdancewic. Dependent inoperability. In *The Sixth ACM SIGPLAN Workshop on Programming Languages meets Program Verification (PLPV)*, 2012.
4. Karl Mazurak, Jianzhou Zhao, and Steve Zdancewic. Lightweight linear types in System F°. In *ACM SIGPLAN International Workshop on Types in Languages Design and Implementation (TLDI)*, pages 77–88, 2010.
5. Michael J. May, Carl A. Gunter, Insup Lee, and Steve Zdancewic. Strong and weak policy relations. In *POLICY 2009, IEEE International Symposium on Policies for Distributed Systems and Networks*, pages 33–36, 2009.
6. Limin Jia and Steve Zdancewic. Encoding information flow in Aura. In *Proceedings of the 2009 Workshop on Programming Languages and Analysis for Security (PLAS)*, pages 17–29, 2009.
7. Karl Mazurak and Steve Zdancewic. ABash: Finding bugs in bash scripts. In *ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS)*, June 2007.

8. Michael Hicks, Stephen Tse, Boniface Hicks, and Steve Zdancewic. Dynamic updating of information-flow policies. In *Proc. of Foundations of Computer Security Workshop (FCS)*, 2005.
9. Peng Li and Steve Zdancewic. Unifying Confidentiality and Integrity in Downgrading Policies. In *Proc. of Foundations of Computer Security Workshop (FCS)*, 2005.
10. Peng Li, Yun Mao, and Steve Zdancewic. Information Integrity Policies. In *Proceedings of the Workshop on Formal Aspects in Security & Trust (FAST)*, September 2003.
11. Usa Sammapun, Raman Sharykin, Margaret Delap, Myong Kim, and Steve Zdancewic. Formalizing Java-MaC. In *Proceedings of the Third Runtime Verification Workshop*, pages 171–190. Electronic Notes in Theoretical Computer Science, July 2003.
12. Michael Greenwald, Carl A. Gunter, Björn Knutsson, Andre Scedrov, Jonathan M. Smith, and Steve Zdancewic. Computer Security is Not a Science (but it should be). In *Proceedings of the Large-Scale Network Security Workshop*, March 2003.
13. Greg Morrisett, Karl Crary, Neal Glew, Dan Grossman, Richard Samuels, Frederick Smith, David Walker, Stephanie Weirich, and Steve Zdancewic. TALx86: A Realistic Typed Assembly Language. In *2nd ACM SIGPLAN Workshop on Compiler Support for System Software*, pages 25–35, 1999.

Technical Reports, Works in Progress, and Unpublished Manuscripts

1. Jennifer Paykin and Steve Zdancewic. A HoTT Quantum Equational Theory (Extended Version). available on arXiv, 2019.
2. Jennifer Paykin and Steve Zdancewic. A linear/producer/consumer model of classical linear logic. Technical report, University of Pennsylvania, 2014.
3. Christian DeLozier, Richard Eisenberg, Santosh Nagarakatte, Peter-Michael Osera, Milo M.K. Martin, and Steve Zdancewic. Ironclad C++: A library-augmented type-safe subset of C++. Technical Report MS-CIS-13-05, University of Pennsylvania, March 2013.
4. Jianzhou Zhao, Qi Zhang, and Steve Zdancewic. Relational parametricity for polymorphic linear lambda calculus (extended tr). 2010.
5. Dmitri Garbuzov, William Mansky, Christine Rizkallah, and Steve Zdancewic. Structural operational semantics for control flow graph machines, 2018.
6. Brian Aydemir, Stephanie Weirich, and Steve Zdancewic. Abstracting Syntax. (15 pages), 2008.
7. Stephen Tse and Steve Zdancewic. Concise concrete syntax. Technical Report MS-CIS-08-11, University of Pennsylvania, 2008.
8. Limin Jia, Jeffrey A. Vaughan, Karl Mazurak, Jianzhou Zhao, Luke Zarko, Joseph Schorr, and Steve Zdancewic. AURA: preliminary technical results. Technical Report MS-CIS-08-10, University of Pennsylvania, 2008.
9. Jeffrey C. Vaughan, Limin Jia, Karl Mazurak, and Steve Zdancewic. Evidence-based audit, technical appendix. Technical Report MS-CIS-08-09, University of Pennsylvania, 2008.
10. Stephen Tse and Steve Zdancewic. Translating Dependency into Parametricity. (33 pages) Accepted to *Journal of Functional Programming*, pending revisions, 2006.
11. Stephen Tse and Steve Zdancewic. Designing a Security-typed Language with Certificate-based De-classification. Technical Report MIS-CIS-04-16, University of Pennsylvania, 2004.

12. Stephen Tse and Steve Zdancewic. Translating Dependency into Parametricity. Technical Report MIS-CIS-04-01, University of Pennsylvania, 2004.
13. Stephen Tse and Steve Zdancewic. Run-time Principals in Information-flow Type Systems. Technical Report MS-CIS-03-39, University of Pennsylvania, 2003. The conference version appears in IEEE Security and Privacy 2004.
14. Stephan A. Zdancewic. *Programming Languages for Information Security*. PhD thesis, Cornell University, August 2002.
15. Steve Zdancewic, Lantian Zheng, Nathaniel Nystrom, and Andrew C. Myers. Secure Program Partitioning. Technical Report 2001-1846, Computer Science Dept., Cornell University, 2001.
16. Steve Zdancewic and Andrew C. Myers. Confidentiality and Integrity with Untrusted Hosts. Technical Report 2000-1810, Computer Science Dept., Cornell University, 2000.
17. Steve Zdancewic and Dan Grossman. Principals in Programming Languages: Technical Results. Technical Report TR99-1752, Computer Science Dept., Cornell University, June 1999.