David A. Naumann October 2, 2024

Professor of Computer Science, Stevens Institute of Technology, Hoboken NJ 07030.
`https://dnaumann.github.io/dnaumann/` `http://www.cs.stevens.edu/~naumann/`

Education

University of Texas at Austin, Computer Science, BA 1982 and PhD 1992
Dissertation advisors: Professors Edsger W. Dijkstra and C.A.R. Hoare

Appointments

Department Chair, Department of Computer Science, Stevens Institute of Technology, since Feb. 2022.

Visiting Fellow, Department of Computer Science, Princeton University (DeepSpec project), Sept. 2017–May 2018 (on sabbatical)

Visiting Professor, Madrid Institute for Advanced Studies in Software Development Technologies (IMDEA), Mar.–May 2011 (on sabbatical)

Visiting Researcher, Microsoft Research Cambridge, Sept.–Dec. 2010 (on sabbatical)

Visiting Fellow, SRI International, Sept. 2003 (on sabbatical)

Professor of Computer Science, Stevens Institute of Technology, since 2008

Associate Professor of Computer Science, Stevens Institute of Technology, 2002–08

Assistant Professor of Computer Science, Stevens Institute of Technology, 1997–02

Assistant Professor of Mathematics and Computer Science, Southwestern University, 1991–97

Associate Scientist, International Software Systems, Austin, 1986–91

Consultant-programmer, Renaissance Systems, Austin, 1985–86

Programmer-designer, IBM, Austin, 1982–85

Journal Papers

§ indicates co-authors who were students at the time of submission.

1. A recursion theorem for predicate transformers on inductive data types. *Information Processing Letters* 50 (1994) 329–336.

2. Data refinement, call by value, and higher order programs. *Formal Aspects of Computing* 7 (1995) 652–662.

3. Predicate transformers and higher order programs. *Theoretical Computer Science* 150 (1995) 111–159.

4. A categorical model of higher order imperative programming. *Mathematical Structures in Computer Science* 8 (1998) 351–399.

5. A weakest precondition semantics for refinement of object-oriented programs (with Ana Cavalcanti). *IEEE Transactions on Software Engineering* 26 (2000) 713–728.

6. Calculating sharp adaptation rules. *Information Processing Letters* 77 (2001) 201–208.

7. Predicate transformer semantics of a higher order imperative language with record subtyping. *Science of Computer Programming* 41 (2001) 1–51.

8. Soundness of data refinement for a higher order imperative language. *Theoretical Computer Science* 278 (2002) 271–301.

9. Stack-based access control for secure information flow (with Anindya Banerjee). *Journal of Functional Programming* 15 (2005) 131–177.

10. Ownership confinement ensures representation independence for object-oriented programs (with Anindya Banerjee). *Journal of the ACM* 52 (2005) 894–960.

11. Towards imperative modules: Reasoning about invariants and sharing of mutable state (with Michael Barnett). *Theoretical Computer Science* 365 (2006) 143–168.

12. Observational purity and encapsulation. *Theoretical Computer Science* 376 (2007) 205–224.

13. On assertion-based encapsulation for object invariants and simulations. *Formal Aspects of Computing* 19 (2007) 205–224. (Coincidentally, same pages as previous item.)

14. Refactoring and representation independence for class hierarchies (with Augusto Sampaio and Leila Silva). *Theoretical Computer Science* 433 (2012) 60–97.

15. Local reasoning for global invariants, Part I: Region logic (with Anindya Banerjee and Stan Rosenberg§). *Journal of the ACM* 60 (2013) 18:1–18:56.

16. Local reasoning for global invariants, Part II: Dynamic boundaries (with Anindya Banerjee). *Journal of the ACM* 60 (2013) 19:1–19:73.

17. Guiding a general-purpose C verifier to prove cryptographic protocols (with François Dupressoir§, Andrew D. Gordon, and Jan Jürjens). *Journal of Computer Security* 22 (2014) 823–866.

18. Behavioral subtyping, specification inheritance, and modular reasoning (with Gary Leavens). *ACM Transactions on Programming Languages and Systems* 37 (2015) 13:1–13:88.

19. Towards patterns for heaps and imperative lambdas. *Journal of Logical and Algebraic Methods in Programming* 85 (2016) 1038–1056.

20. A logical analysis of framing for specifications with pure method calls (with Anindya Banerjee and Mohammad Nikouei§). *ACM Transactions on Programming Languages and Systems* 40 (2018) 6:1–6:90.

21. A relational program logic with data abstraction and dynamic framing (with Anindya Banerjee, Ramana Nagasamudram§, and Mohammad Nikouei§). *ACM Transactions on Programming Languages and Systems* 44 (2022) 25:1–25:136.

Proceedings of Refereed Conferences

1. Derivation of programs for freshmen (with R. Denman, W. Potter, and G. Richter). ACM *Conference of the Special Interest Group in Computer Science Education* SIGCSE (1994) 116–120.

2. Predicate transformer semantics of an Oberon-like language. *IFIP TC2 Working Conference on Programming Concepts, Methods and Calculi* (1994) 467–487.

3. On the essence of Oberon. *International Conference on Programming Languages and System Architecture*, Zürich, Jürg Gutknecht, ed. (1994) 313–327.

4. Beyond Fun: order and membership in polytypic imperative programming. *5th International Conference on Mathematics of Program Construction*, (1998) 286–314.

5. Towards squiggly refinement algebra. *Proceedings, IFIP Programming Concepts and Methods* (1998) 346–365.

6. A weakest precondition semantics for refinement in an object-oriented language (with Ana Cavalcanti). *World Congress on Formal Methods* (1999) 1439–1459.

7. Ideal models for pointwise relational and state-free imperative programming. ACM *Principles and Practice of Declarative Programming* (2001) 4–15.

8. Representation independence, confinement, and access control (with Anindya Banerjee). *29th ACM Symposium on Principles of Programming Languages* (2002) 166–177.

9. Forward simulation for data refinement of classes (with Ana Cavalcanti). *International Symposium of Formal Methods Europe* (2002) 471–490.

10. On a specification-oriented model for object-orientation (with Ana Cavalcanti). *6th Brazilian Symposium on Programming Languages* (2002) 114–127.

11. Secure information flow and pointer confinement in a Java-like language (with Anindya Banerjee). *15th IEEE Computer Security Foundations Workshop*[1] (2002) 253–270.

[1] Nominally this was a refereed workshop, not a conference, but at the quality standard of a symposium—indeed, in 2007 it was renamed to the *20th* Computer Security Foundations *Symposium*.

12. Using access control for secure information flow in a Java-like language (with Anindya Banerjee). *16th IEEE Computer Security Foundations Workshop* (2003) 155–169.

13. CodeBLUE: a Bluetooth interactive dance club system (with Dennis Hromin[§], Michael Chladil[§], Natalie Vanatta[§], Susanne Wetzel, Farooq Anjum, and Ravi Jain). *IEEE Global Telecommunications Conference* (2003) 2814–2818.

14. Towards imperative modules: Reasoning about invariants and sharing of mutable state, extended abstract (with Mike Barnett). *19th IEEE Symposium on Logic in Computer Science* (2004) 313–323.

15. Friends need a bit more: Maintaining invariants over shared state (with Mike Barnett) *7th International Conference on Mathematics of Program Construction* (2004) 54–84.

16. Modular and constraint-based information flow inference for an object-oriented language (with Qi Sun[§] and Anindya Banerjee). *11th International Static Analysis Symposium* (2004) 84–99.

17. Assertion-based encapsulation, object invariants and simulations (invited survey paper), in post-proceedings, *Formal Methods for Components and Objects* (2004) 251–273.

18. Observational purity and encapsulation. *Fundamental Aspects of Software Engineering* (2005) 190–204. Awarded Best Software Sciences Paper of ETAPS 2005.

19. State based ownership, reentrance, and encapsulation (with Anindya Banerjee). *19th European Conference on Object-Oriented Programming* (2005) 387–411.

20. Verifying a secure information flow analyzer. *18th International Conference on Theorem Proving in Higher Order Logics* (2005) 211–226.

21. Deriving an information flow checker and certifying compiler for Java (with Gilles Barthe and Tamara Rezk[§]). *27th IEEE Symposium on Security and Privacy* (2006) 230–242.

22. From coupling relations to mated invariants for checking secure information flow. *11th European Symposium on Research in Computer Security* (2006) 279–296.

23. Closing internal timing channels by transformation (with Alejandro Russo[§], John Hughes, and Andrei Sabelfeld). In *11th Asian Computing Science Conference* (2006).

24. Allowing state changes in specifications (with Michael Barnett, Wolfram Schulte, and Qi Sun[§]). In *International Conference on Emerging Trends in Information and Communication Security* (2006) 321–336, invited paper.

25. Category theoretic models of data refinement (with Michael Johnson and John Power). In *Irish Conference on Mathmatical Foundations of Computer Science and Information Technology* (2006), invited paper. In Springer Electronic Notes in Theoretical Computer Science 225(2) (2009) 21–38.

26. Beyond stack inspection: a unified access-control and information-flow security model (with Marco Pistoia and Anindya Banerjee). In *28th IEEE Symposium on Security and Privacy* (2007) 149–163.

27. Modular verification of higher-order methods with mandatory calls specified by model programs (with Steve M. Shaner[§] and Gary T. Leavens). In *22d International Conference on Object Oriented Programming, Languages, and Systems* (2007) 351–368. Awarded Best Student Paper.

28. Expressive declassification policies and modular static enforcement (with Anindya Banerjee and Stan Rosenberg[§]). In *29th IEEE Symposium on Security and Privacy* (2008) 339–353.

29. Regional logic for local reasoning about global invariants (with Anindya Banerjee and Stan Rosenberg[§]). In *European Conference on Object Oriented Programming* (2008) 387–411. Awarded Distinguished Paper.

30. Boogie meets regions: a verification experience report (with Anindya Banerjee and Mike Barnett). In *Verified Software: Theories, Tools, Experiments* (2008) 177–191.

31. Dynamic boundaries: Information hiding by second order framing with first order assertions (with Anindya Banerjee). In the *Programming Languages and Systems, 19th European Symposium on Programming (ESOP)* (2010) 2–22.

32. Information flow monitor inlining (with Andrey Chudnov[§]). In *IEEE Computer Security Foundations Symposium* (2010) 200–214.

33. Local reasoning and dynamic framing for the composite pattern and its clients (with Stan Rosenberg[§]and Anindya Banerjee). In proceedings of 3d International Conference on *Verified Software: Theories, Tools, Experiments (VSTTE)* (2010) 183–198.

34. Guiding a general-purpose C verifier to prove cryptographic protocols (with François Dupressoir[§], Andrew D. Gordon, and Jan Jürjens). In *IEEE Computer Security Foundations Symposium* (2011) 3–17.

35. Symbolic analysis for security of roaming protocols in mobile networks (with Chunyu Tang[§] and Susanne Wetzel). In *ICST Conference on Security and Privacy in Communication Networks* (2011) 480–490.

36. Decision procedures for region logic (with Stan Rosenberg[§] and Anindya Banerjee). In *13th International Conference on Verification, Model Checking, and Abstract Interpretation* (2012) 379–395.

37. Laws of programming for references (with Giovanny Lucero[§] and Augusto Sampaio). In *11th Asian Symposium on Programming Languages and Systems* (2013) 124–139.

38. Analysis of authentication and key establishment in inter-generational mobile telephony (with Chunyu Tang[§] and Susanne Wetzel). In *IEEE International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing* (2013) 1605–1614 (see www.iacr.org/2013/227).

39. A logical analysis of framing for specifications with pure method calls (with Anindya Banerjee). In post-proceedings of *Verified Software: Theories, Tools, Experiments* (2014) 3–20.

40. Information flow monitoring as abstract interpretation for relational logic (with Andrey Chudnov[§] and George Kuan). In *IEEE Computer Security Foundations Symposium* (2014) 48–62.

41. Inlined information flow monitoring for JavaScript (with Andrey Chudnov[§]). In *ACM Conference on Computer and Communication Security* (2015) 629–643.

42. Calculational design of information flow monitors (with Mounir Assaf). In *IEEE Computer Security Foundations Symposium* (2016) 210–224.

43. Specifying and verifying advanced control features (with Gary T. Leavens, Hridesh Rajan, Tomoyuki Aotani). In *7th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation* (2016).

44. Relational logic with framing and hypotheses (with Anindya Banerjee and Mohammad Nikouei[§]). In *36th Conference on Foundations of Software Technology and Theoretical Computer Science* (2016) 11:1–11:16.

45. Hypercollecting semantics and its application to static analysis of information flow (with Mounir Assaf, Julien Signoles, Éric Totel, Frédéric Tronel). In *44th ACM Symposium on Principles of Programming Languages* (2017) 874–887.

46. Assuming you know: epistemic semantics of relational annotations for expressive flow policies (with Andrey Chudnov). In *IEEE Computer Security Foundations Symposium* (2018) 189–203.

47. Whither specifications as programs (with Minh Ngo). In *Symposium on Unifying Theories of Programming* (2019) 39–61.

48. Verified sequential malloc/free (with Andrew W. Appel). In *International Symposium on Memory Management* (2020) 48–59.

49. Thirty-seven years of relational Hoare logic: Remarks on its principles and history. In *9th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation* (2020) 93–116.

50. Typed-based declassification for free (with Minh Ngo and Tamara Rezk). In *22d International Conference on Formal Engineering Methods* (2020) 181–197.

51. Alignment completeness for relational Hoare logics (with Ramana Nagasmudram[§]). In *36th Annual ACM/IEEE Symposium on Logic in Computer Science* (2021) 1–13.

52. An algebra of alignment for relational verification (with Timos Antonopoulos, Eric Koskinen, Ton Chanh Le, Ramana Nagasamudram[§], Minh Ngo). In *50th ACM Symposium on Principles of Programming Languages* (2023).

53. The WhyRel prototype for modular relational verification of pointer programs (with Ramana Nagasamudram[§]and Anindya Banerjee). In *29th International Conference on Tools and Algorithms for the Construction and Analysis of Systems* (2023). Nominated for an ETAPS best paper award.

54. Toward tool-independent summaries for symbolic execution (with Frederico Ramos, Nuno Sabino, Pedro Adão, and José Fragoso Santos). In *European Conference on Object-Oriented Programming* (2023).

55. Assume but verify: deductive verification of leaked information in concurrent applications (with Toby Murray, Mukesh Tiwari, and Gidon Ernst). In *ACM SIGSAC Conference on Computer and Communications Security* (2023) 1746–1760.

56. Verifying a C implementation of Derecho's coordination mechanism using VST and Coq (with Ramana Nagasamudram, Lennart Beringer, Ken Birman, and Mae Milano). In *NASA Formal Methods Symposium* (2024).

Book chapters

1. State Based Encapsulation for Modular Reasoning about Behavior-Preserving Refactorings (with Anindya Banerjee). Invited chapter in *Aliasing in Object-oriented Programming*, Dave Clarke and James Noble and Tobias Wrigstad, eds., Springer State-of-the-art Surveys (2013) 319–365.

2. A Simple Semantics and Static Analysis for Stack Inspection (with Anindya Banerjee). In *Semantics, Abstract Interpretation, and Reasoning about Programs: Essays Dedicated to David A. Schmidt on the Occasion of his Sixtieth Birthday*, EPTCS 129 (2013) 284–308 (DOI: 10.4204/EPTCS.129, ISSN: 2075-2180).

3. An Illustrated Guide to the Model Theory of Supertype Abstraction and Behavioral Subtyping (with Gary Leavens). In *Engineering Trustworthy Software Systems*, Jonathan P. Bowen and Zhiming Liu and Zili Zhang, eds., Springer (2018) 39–88.

Edited collections

1. Proceedings of Seminar 03411 on Language Based Security, 2005, Dagstuhl, Germany. Co-editor with Anindya Banerjee, Heiko Mantel, and Andrei Sabelfeld (http://www.dagstuhl.de/03411/).

2. Proceedings of the ACM SIGPLAN Fourth Workshop on Programming Languages and Analysis for Security, 2009, Dublin, Ireland. Co-editor with Stephen Chong (Harvard). Published by ACM Press, 131 pages, ISBN 978-1-60558-645-8.
3. VSTTE 2010 Workshop Proceedings (119 pages). Co-editor with Rajeev Joshi (NASA), Tiziana Margaria (Potsdam), Peter Müller (ETH), and Hongseok Yang (U. London). Technical Reports 676, ETH Zurich, Computer Science.
4. Formal Methods: Foundations and Applications – Proceedings of 15th Brazilian Symposium, 2012. Co-editor with Rohit Gheyi.
5. Fifth International Symposium on Unifying Theories of Programming (UTP 2014), Revised Selected Papers. Editor.
6. Proceedings, 34th IEEE Symposium on Computer Security Foundations, 2021. Co-editor with Ralf Küsters.
7. Proceedings, 35th IEEE Symposium on Computer Security Foundations, 2022. Co-editor with Stefano Calzavaro.
8. Special issue of *Journal of Computer Security*, volume 31 number 5, for CSF 2022. Co-editor with Stefano Calzavaro.

Selected awards

Best Software Science Paper, ETAPS 2005 (European Association of Software Sciences and Technology).

Davis Memorial Award for Research Excellence, Stevens Institute of Technology, 2006.

Best Student Paper, OOPSLA 2008 (well, the student is Steven M. Shaner and the third coauthor is his advisor, Gary T. Leavens).

Distinguished Paper Award, ECOOP 2008 (with coauthors Anindya Banerjee and Stan Rosenberg).

Nominated for best paper award, ETAPS 2023 (with coauthors Ramana Nagasamudram and Anindya Banerjee).

Consulting

Microsoft Research; Vulcan Inc.; Galois Inc.

Patents

Patent US 10,904,291 B1, Jan. 26, 2021. Low-overhead software transformation to enforce information security policies. David Naumann, Andrey Chudnov[§], Aleksey Nogin, Pape Sylla.

Selected recent activities

Editorial board member:
ACM Transactions on Programming Languages and Systems
ACM Formal Aspects of Computing
Journal of Object Technology

Steering committee member: IEEE Computer Security Foundations Symposium 2021–2026.

Program co-chair: 34th & 35th IEEE Computer Security Foundations Symposium 2021 and 2022.

Chair of Panel of Experts reviewing 13 research teams in the area "Proofs and Verification", INRIA, France 2019.

Program chair: 5th International Symposium on Unifying Theories of Programming 2014, co-located with FM2014 in Singapore.

Program co-chair: 15th Brazilian Symposium on Formal Methods (SBMF) 2012.

Co-organizer: New Jersey Programming Languages Seminar, April 2010 (with Adriana Compagnoni and Dominic Duggan).

Program co-chair: 4th ACM Workshop on Programming Languages and Analysis for Security in connection with International Conference on *Programming Language Design and Implementation* 2009 (with Stephen Chong).

Co-organizer: IBM Programming Languages Day 2009 (with Rajesh Bordawekar, IBM, and Riccardo Pucella, Northeastern U.)

Co-chair of Theory Workshop in connection with International Conference on *Verified Software: Theory, Tools, Experiments* 2010 (with Hongseok Yang), and 2008 (with Peter O'Hearn).

Chair of Theory Working Group, Verified Software Initiative,[2] 2005–2007.

Co-editor: *Concurrency and Computation: Practice and Experience*, June 2004, special issue on formal methods for Java programs.

Co-organizer and co-editor of proceedings: first Dagstuhl seminar on Language Based Security (Seminar 03411, Oct. 5–10, 2003), with Anindya Banerjee, Heiko Mantel, and Andrei Sabelfeld.

Program committee member, refereed conferences:

European Conference on Object-Oriented Pogramming (ECOOP) 2024;
ACM Principles of Programming Languages (POPL) 2023, 2019, 2011;
European Symposium on Programming (ESOP) 2019, 2008, 2004;
IEEE Computer Security Foundations Symposium (CSF) 2022 (co-chair), 2021 (co-chair), 2020, 2019, 2015;

[2] http://vstte.ethz.ch/index.html and http://www.dagstuhl.de/06281/

International Symposium on Formal Methods (FM) 2021, 2015;
International Symposium on Unifying Theories of Programming (UTP) 2019, 2014, 2012, 2010, 2008, 2006. International Conference on Runtime Verification (RV) 2016, 2014;
4th Conference Principles of Security and Trust (POST) 2015;
Verified Software: Theories, Tools, Experiments (VSTTE) 2016, 2013, 2010, 2008;
International Conference on Mathematics of Program Construction (MPC) 2015, 2012, 2010, 2004, 2002, 2000;
ACM Computer and Communication Security (CCS) 2010;
Brazilian Symposium on Formal Methods (SBMF) 2018, 2017, 2016, 2015, 2014, 2013, 2012 (co-chair), 2011, 2010, 2009, 2008, 2007, 2004;
IFIP Formal Methods for Open Object-based Distributed Systems (FMOODS) 2008, 2007;
International Conference on Formal Engineering Methods (ICFEM) 2009, 2005;
TOOLS Europe 2008;
European Symposium on Research in Computer Security (ESORICS) 2007;
14th International Symposium on Formal Methods (FM) 2006;
7th IFIP International Conference on Theoretical Computer Science 2012;
Brazilian Symposium on Programming Languages 2008, 2007, 2006, 2005, 2004;
Brazilian Symposium on Software Engineering 2005, 2003, 2002;

Advisees and mentees

Postdocs: Mounir Assaf (2015–2017); Minh Ngo (2019).

My graduated PhD Students:

Qi Sun (defended October 2007, now at Google): *Constraint-based Secure Information Flow Inference for Object-oriented Programs*).

Stan Rosenberg (defended June 2011): *Region Logic: Local Reasoning for Java Programs and its Automation.*

Chunyu Tang (defended December 2013): *Modeling and Analysis of Mobile Telephony Protocols.*

Andrey Chudnov (defended April 2015, now at Galois Inc.): *Inlined Information Flow Monitoring for Web Applications in JavaScript.*

Mohammad Nikouei (defended November 2019): *A Logical Analysis of Relational Program Correctness.*

Other PhDs:

Opponent and PhD examiner for Leonid Mikhajlov, *Software reuse mechanisms and techniques: safety versus flexibility*, Turku University, Finland, supervisor Ralph Back, 1999.

PhD examiner for Hongseok Yang, *Local reasoning for stateful programs*, University Illinois, supervisor Uday Reddy, 2001 (student was supported by my award INT-9813854).

PhD examiner for Noah Torp-Smith, *Advances in Separation Logic*, IT University of Copenhagen, supervisor Lars Birkedal, 2005.

PhD examiner for Yannis Kassios, *A theory of object-oriented refinement*, University of Toronto, supervisor Eric Hehner, 2006.

PhD examiner for Mariela Pavlova, *Framework for formal verification of Java bytecode against functional and security policies*, University of Nice, supervisor Gilles Barthe, 2007.

Opponent and PhD examiner for Daniel Hedin, *Program analysis issues in language based security*, Chalmers University of Technology, Gothenberg, supervisor David Sands, 2008.

Opponent and PhD examiner for Musard Balliu, *Logics for information flow security: from specification to verification*, KTH Royal Institute of Technology, Stockholm, supervisor Mads Dam, 2014.

PhD examiner for José Fragoso Santos, *Enforcing secure information flow in client-side web applications*, University of Nice (INRIA), Sophia Antipolis, supervisors Tamara Rezk and Ana Almeida Matos, 2014.

PhD co-advisor for Giovanny Lucero, *Algebraic laws for object oriented programming with references*, Federal University of Pernambuco, Brazil, supervisor Augusto Sampaio, 2015.

PhD committee member for Yuyan Bao, *Reasoning about frame properties in object-oriented programs*, University of Central Florida, supervisor Gary Leavens, 2017.

PhD examiner for Michele Pasqua, *Hyper static analysis of programs: an abstract interpretation-based framework for hyperproperties verification.* University of Verona, supervisor Isabella Mastroeni, 2019.

PhD examiner for Mathias Pedersen, *Enforcement of Timing-Sensitive Security Policies in Runtime Systems.* Aarhus University, supervisor Aslan Askarov, 2019.

Rapporteur and jury member for Mohamad El Laz, *Provable Encryption Schemes for Distributed Systems.* Univelsité de Nice / INRIA Sophia-Antipolis, supervisor Tamara Rezk, March 2022.

PhD examiner for Jana Hofmann, *Logical Methods for the Hierarchy of Hyperlogics.* CISPA Helmholtz Center for Information Security, Saarbrücken, Germany, supervisor Bernd Finkbeiner, December 2022.

Supervised research internship (Oct 2010–Feb 2011) of Mehdi Bouaziz, *Symbolic execution for JavaScript*, École Normale Supérieure (ENS), Paris, supervisor Patrick Cousot.

Supervised research internship (Mar 2013–July 2013) Guillaume Bury, *Automating relational verification in Why3*, École Normale Supérieure (ENS), Paris, supervisor Xavier Rival.

PhD committee member: *At Stevens*: Mark Wolfskehl (1999), Tendü Yogurtcu (2000), Ricardo Medel (2007), Uma Batchu (2007), Ye Wu (2010), Yifei Bao (2013), Yidi Zhang (2014), Ruilin

Yang (2015), Walter Krawec (2015), Hesham Mansour (2015), Boxiang Dong (2016), Michael Engling (2017), Yuandong Cyrus Liu (2022). *Elsewhere*: Rohit Gheyi, Federal University of Pernambuco, Brazil (2007).

Selected external funding

Towards a Practical Calculus of Object-Oriented Programming. NSF-CNPq Collaborative Research Opportunities INT-9813854: $200,000 (1999–2001). Sole PI, with subcontractor Uday Reddy, University Illinois (linked to collaborative award to Paulo Borba, Augusto Sampaio, and Ana Cavalcanti, Federal University of Pernambuco, Brazil).

CodeBlue senior design team. Telcordia Technologies: approx. $14,000 (2001). Additional $15,000 granted 2003 for ongoing research on this case study.

Stanley Fellowship for PhD student Gerald Thompson: $15,000 stipend plus tuition (2001–2002).

Integrating Confinement and Access Control for Encapsulation. NSF Trusted Computing CCR-0208984: $168,727 (2002–2004). Sole PI (linked to collaborative award to Anindya Banerjee at Kansas State).

Formal Methods for Behavioral Subclassing and Callbacks. NSF Computing Processes and Artifacts CCF-0429894: $161,995 (2004–2006). Sole PI (linked to collaborative award to Gary Leavens at Iowa State).

Stanley Fellowship for PhD student Stan Rosenberg: $15,000 stipend plus tuition 2005-06, renewed 2006-07.

Collaborative Research: Access Control and Downgrading in Information Flow Assurance. NSF CyberTrust CNS-0627338: $206,000 (2006–2009). Sole PI (linked to collaborative award to Anindya Banerjee at Kansas State).

Collaborative Research: A JML Community Infrastructure –Revitalizing Tools and Documentation to Aid Formal Methods Research. NSF Computing Research Infrastructure CNS-0708330: $100,000 (2007–2010). Sole PI (linked to collaborative awards to five other universities).

Innovation and Entrepreneurship Fellowship for PhD student Andrey Chudnov: full stipend and tuition for 2+1 years (2008–2011).

SFS Cybersecurity Scholars Program. NSF Scholarship for Service DUE-0830846: $1,000,601 plus supplements (2009–2014). Co-PI (with Wetzel (PI), Subbalakshmi, Chandramouli).

Wireless Network Modeling. US Government (Picatinny) $330,000 (2008–2009). Co-PI with PI Wetzel

Specification Language Foundations for Modular Reasoning Methodologies. NSF Computing and Communications Foundations CCF-0915611: $249,949 (2009–2012). Sole PI (linked to collaborative award to Gary Leavens at U. Central Florida).

Tunable Information Flow. Department of Homeland Security, Science and Technology Directorate (Sole PI on subcontract of contract to HRL Laboratories): $413,285.97 (2012-2014).

Flexible and Practical Information Flow Assurance for Mobile Apps. NSF Secure and Trustworthy Cyberspace CNS-1228930: $631,808 (2012–2015). Sole PI (linked to collaborative award to Gary Leavens at U. Central Florida).

Hyperproperty Abstraction for Information Flow Control. NSF Computing and Communication Foundations CCF-1649884: $104,790 (2016–2017). Sole PI.

Relational Verification for Information Assurance and Privacy. NSF Secure and Trustworthy Cyberspace CNS-1718713: $451,931 (2017–2020). Sole PI.

Provost Fellowship for PhD student Ramana Nagasamudram (2020–2022).

Epistemic Reasoning for Safety-Critical Systems. Siemens Corp.: $47,155 (2021). Sole PI.

Siemens Fellowship for High Assurance. Siemens Corp.: $105,331 (2021–2022). Sole PI.

Auto-active Hyperproperty Verification for Security. NSF Secure and Trustworthy Cyberspace CNS 2426414: $588,353 (2024–2027). Sole PI.

Selected talks since 2002

* = invited

I omit my talks in my department's seminar series and in the Lab for Secure Systems weekly series.

1. * **Lucent Bell Laboratories**, 9 Jan 02. Representation independence, confinement and access control.
2. **29th ACM Symposium on Principles of Programming Languages**, Portland, OR, 17 Jan 02. Representation independence, confinement and access control.
3. * **Java Verification Workshop, Oregon Graduate Center**, Portland, 12 Jan 02. Reasoning about modules: data refinement and simulation.
4. **Stevens Symposium on CyberSecurity and Trusted Software**, 15 Mar 02. Java, access control, and static analysis.
5. **6th Brazilian Symposium on Programming Languages**, Rio de Janeiro, 5 June 02. On a specification-oriented model for object-orientation.
6. * **6th Brazilian Symposium on Programming Languages**, Rio de Janeiro, 6 June 02. Representation independence, confinement and access control.

7. **15th IEEE Computer Security Foundations**, Cape Breton, Nova Scotia, 26 June 02. Secure information flow and pointer confinement in a Java-like language,

8. * **PointerFest, Queen Mary University of London**, 15 Aug 02. Representation independence, confinement and access control.

9. * **Dagstuhl Seminar on Reasoning about Shape**, Dagstuhl, Germany, 5 March 03. Pointer confinement and abstraction.

10. * **Microsoft Research, Redmond**, 12 March 03. Using access control for secure information flow.

11. * **Cornell University, Information Assurance Institute**, 20 March 03. Using access control for secure information flow in a Java-like language.

12. * **Princeton University, CS Seminar**, 12 May 03. Using access control for secure information flow in a Java-like language

13. * **SRI International** (Stanford Research Institute), Menlo Park, CA, 26 June 03. Using access control for secure information flow.

14. **ECOOP Workshop on Formal Techniques for Java-like Programs**, 21 July 03. Ownership: transfer, sharing, and encapsulation.

15. * **Cornell University, CS Seminar**, 29 Aug 03. Using access control for secure information flow—recent advances.

16. **Dagstuhl Seminar on Language Based Security**, 10 Oct 03. Summary talk outlining research agenda that emerged during this week-long event of which I was a co-organizer.

17. * **Microsoft Research, Cambridge, UK**, 19 Nov 03. Object invariants and owner transfer: issues and approaches.

18. * **INRIA Sophia-Antipolis, France**, 26 Nov 03. Object invariants and owner transfer: issues and approaches.

19. * **Microsoft Research, Redmond, WA**, 10 Dec 03. Object invariants and owner transfer: issues and approaches.

20. **Stevens Cybersecurity Symposium**, 26 March 04. After the buffer overflows are all patched: pre-deployment verification of behavioral contracts.

21. **19th IEEE Symposium on Logic in Computer Science**, Turku, Finland, 16 July 04. Towards imperative modules: Reasoning about invariants and sharing of mutable state.

22. **New Jersey Programming Languages Seminar**, 1 Oct 04. Towards imperative modules: reasoning about invariants and sharing of mutable state.

23. * **Lucent Bell Laboratories**, Jan 05 (lost exact date; hosted by K. Namjoshi). Towards imperative modules.

24. * **European Joint Conferences on Theory and Practice of Software: Grand Challenge Workshop on Software Verification, Edinburgh**, 3 April 05. Using auxiliary state to express modular structure.

25. **European Joint Conferences on Theory and Practice of Software: Conference on Fundamental Aspects of Software Engineering**, 7 April 05. Observational purity and encapsulation.

26. * **Microsoft Research, Redmond**, 9 June 05. Observational purity.

27. **19th European Conference on Object-Oriented Programming**, Glasgow, Scotland, 29 July 05. State based ownership, reentrance, and encapsulation.

28. **18th International Conference on Theorem Proving in Higher Order Logics**, Oxford, UK, 25 Aug 05. Verifying a secure information flow analyzer.

29. * **IT University of Copenhagen**, 6 Oct 05. State based ownership, reentrance, and encapsulation.

30. **SRI International**, Menlo Park, CA, 3 April 06. Verified Software Theory Panel Report.

31. **SRI International**, Washington DC, 26 April 06. Verified Software Theory Panel Report.

32. * **IBM Research, Distinguished Speaker**, 27 April 06. Types and contracts for specifying and checking information flow.

33. * **Center for Informatics, Federal University of Pernambuco**, Recife, Brazil, 12 June 07. Pointer Confinement, Encapsulation, and Data Refinement.

34. **Dagstuhl seminar: Challenge of Software Verification**, Dagstuhl, Germany, 13 July 06. Verified Software Theory Panel Report.

35. * **ProSec Security Group, Chalmers University of Technology**, Gothenberg, Sweden, 24 July 06. From coupling relations to mated invariants for checking information flow.

36. **Northeast Verification Seminar, New York University**, 13 Oct 06. Secure information flow by mating invariants.

37. **11th European Symposium on Research in Computer Security**, Hamburg, Germany, 19 Sep 06. From coupling relations and mated invariants for checking secure information flow.

38. * **University of Pennsylvania, Computer Security Seminar**, 26 Oct 06. Specifying and verifying information flow policies using relational Hoare logic.

39. **Stevens/Microsoft Workshop for High School Teachers on Teaching CS**, 23 Mar 07. Using DrJava for introductory programming.

40. **Columbia/IBM/Stevens Security and Privacy Day, Columbia U.**, 1 June 07. Information flow verification and declassification.

41. * **Center for Informatics, Federal University of Pernambuco**, Recife, Brazil, 31 July 07. Assertion-based confinement for encapsulation.

42. * **IBM Research**, Hawthorne NY, 6 Sept 2007. Expressive declassification policies and modular static enforcement.

43. * **Harvard University, programming languages seminar**, 17 Oct 2007. Regional logic for local reasoning about global invariants.

44. * **New England Programming Languages Seminar**, Worchester Polytechnic University, 18 Oct 2007. Expressive declassification policies and modular static enforcement.

45. **Mid-Atlantic Programming Language Seminar (MAPLS) in conjunction with NJPLS**, U. Maryland, College Park, 30 Nov 2007. Expressive declassification policies and modular static enforcement.

46. * **Seminar on Types, Logics and Semantics for State**, (3–8 February, 2008), Dagstuhl, Germany, 5 Feb 2008. Regional logic: local reasoning, global invariants.

47. * **Computer Science Seminar**, Stony Brook University, 29 Feb 2008. Information flow with reclassification in object-oriented programs: specification, semantics, and modular verification

48. * **Lehman College Math Seminar**, 27 Feb 2008. Can we make software secure and even leakproof?

49. * **Microsoft Research**, Redmond WA, 14 May 2008. Regional logic: local reasoning, global invariants.

50. **IEEE Symposium on Security and Privacy**, 21 May 2008. Expressive declassification policies and modular static enforcement.

51. * **SRI International, formal methods outreach workshop**, 10 June 2008. Some thoughts on formal methods education.

52. **Automated Formal Methods**, workshop of CAV at Princeton, 14 July 2008. Core JML in PVS.

53. * **Computer Science Seminar, Chalmers University of Technology**, Gothenberg, Sweden, 19 Sept 2008. Expressive declassification policies and modular static enforcement.

54. * **PhD Fall School on Logics and Semantics of State, IT University**, Copenhagen, 20–24 October 2008. Short course on verification of object-oriented programs.

55. **The Open Group, Real-Time and Embedded Systems Forum**, San Diego, 2 Feb 2009. Naïve and flexible declassification with scalable enforcement.

56. **The Java Modeling Language (JML)**, Dagstuhl Seminar 09292, Germany, 14 July 2009. CoreJML in PVS.

57. **Typing, Analysis and Verification of Heap-Manipulating Programs**, Dagstuhl Seminar 09301, Germany, 21 July 2009. Dynamic encapsulation boundaries: a second order frame rule for region logic.

58. * **New Jersey Programming Languages Seminar**, Lehigh University, 2 Oct 2009. Region Logic and Dynamic Encapsulation Boundaries.

59. * **Princeton University Computer Science Department Colloquium**, 16 Dec 2009. Smart Assertions for Dumb Provers.

60. * **IFIP Working Group 2.1 on Algorithmic Languages and Calculi**, Braga Portugal, 26 Jan 2010. Dynamic boundaries: information hiding interfaces for object-based programming.

61. * **Keynote at European Symposium on Programming one of the European Joint Conferences on Theory and Practice of Software** (ETAPS), Paphos, Cyprus, 23 March 2010. Dynamic Boundaries: Global Invariants for Local Reasoning.

62. (Presented by my PhD student Andrey Chudnov) **IEEE Computer Security Foundations Symposium**, 18 July 2010. Information Flow Monitor Inlining.

63. * **Keynote at the 13th Brazilian Symposium on Formal Methods** (SBMF), Natal, Brazil, 12 Nov 2010. Dynamic Boundaries: Global Invariants for Local Reasoning.

64. **Microsoft Research Cambridge** programming languages group, 27 Nov 2010. What is Modular Verification?

65. * **IMDEA Software Institute**, Madrid, 26 May 2011. Dynamic Boundaries: Global Invariants for Local Reasoning and Information Hiding.

66. **New Jersey Programming Languages Seminar**, U. Penn, 15 Nov 2012. A Relational Program Logic for Data Abstraction and Security.

67. * **NSF retreat on Android security**, U. California, Riverside, 21 Feb 2013. The Flowspecs project: flexible and practical information flow assurance for mobile apps.

68. * **The Java Modeling Language, Shonan workshop**, Shonan Village Center, Japan, 15 May 2013. Exploring info flow extensions in a project that targets Android apps.

69. * **Keynote at annual meeting of Reliably Secure Software Systems priority program of German Research Foundation**, University of Saarbrucken, 7 Oct 2013. Specifying and verifying fine-grained information flow policies for mobile apps.

70. **Verified Software: Theories, Tools, Experiments**, Vienna, 18 July 2014. A logical analysis of framing for specifications with pure method calls.

71. (Presented by my PhD student Andrey Chudnov) **27th IEEE Computer Security Foundations Symposium**, 19 July 2014 (featured in joint session with CAV). Information flow monitoring as abstract interpretation for relational logic.

72. * **TCS Seminar, KTH Royal Institute of Technology**, Stockholm, 2 Oct 2014. Information flow monitoring as abstract interpretation for relational logic.

73. * **Workshop on Programming Languages and Verification**, INRIA, Sophia Antipolis, 9 Dec 2014. Information flow monitoring as abstract interpretation for relational logic.

74. (Presented by my PhD student Andrey Chudnov) **ACM Computer and Communications Security**, Denver, 14 Oct 2015. Inlined information flow monitoring for JavaScript.

75. * **Keynote at IBM Programming Languages Day**, T.J. Watson Research Center, 23 Nov 2015. Towards practical and highly assured information flow control for mobile apps.

76. **University of Texas at Austin**, Programming Languages Lunch Colloquium, 28 March 2016. Calculational design of information flow monitors.

77. * **Bell Labs Mathematics Seminar**, 14 July 2016. Calculational design of information flow monitors.

78. **Secure Compilation Meeting**, Paris, 15 Jan 2017. Can relational logic facilitate secure compilation?

79. * Panelist on Formal Methods for Reliable Software Security, at Reliably Secure Software Systems final project meeting, **Darmstadt U.**, Sept 2017

80. * **Galois Inc.**, 11 Nov 2017. Towards unbounded data and iteration in SAW (Static Analysis Workbench).

81. * **Dagstuhl Seminar 18151 on Program Equivalence**, April 2018. Relational Region Logic,

82. * **Dagstuhl Seminar 18201 on Secure Compilation**, May 2018. Relational Logic for Fine-grained Security Policy and Translation Validation.

83. **IEEE Computer Security Foundations Symposium**, July 2018. Assuming you know: Epistemic semantics of relational annotations for expressive flow policies.

84. **Aarhus University, LogSem seminar**, 21 Oct 2019. Data abstraction and relational program logic.

85. **Princeton programming languages seminar**, 12 Dec 2019. Data abstraction and relational program logic.

86. **ACM International Symposium on Memory Management**, 16 June 2020. Verified sequential malloc/free.

87. * **Keynote at Software Factory 4.0, Darmstadt U.**, 15 Oct 2020. Data Abstraction and Relational Program Logic.

88. **International Conference on Formal Engineering Methods**, 1 March 2021. Relaxed noninterference for free.

89. **ACM/IEEE Logic in Computer Science**, 29 June 2021. Alignment completeness for relational Hoare logics.

90. * **Vistas in Verified Software, Newton Institute for Mathematical Sciences, UK**, 12 July 2022. Towards algebraic foundations for alignment.

91. * **38th International Conference on Mathematical Foundations of Programming Semantics** (special session on Relational Verification and Formal Reasoning) 13 July 2022. Towards algebraic foundations for alignment.

92. * **INRIA Sophia-Antipolis research seminar**, 7 November 2022. Towards algebraic foundations for alignment.

93. * **NYU Formal Methods Seminar**, 1 December 2022. Towards algebraic foundations for alignment.

94. (Presented by my PhD student Ramana Nagasamudram) **50th Symposium on Principles of Programming Languages**, January 2023. An algebra of alignment for relational verification.

95. (Presented jointly with my PhD student Ramana Nagasamudram) **29th International Conference on Tools and Algorithms for the Construction and Analysis of Systems**, April 2023. The WhyRel prototype for modular relational verification of pointer programs.