Jonathan Protzenko

jonathan.protzenko@ens-lyon.org

I am a **Tech Lead Manager** in the Information Security Engineering division at Google. My research focuses on advancing the theory and practice of software verification, i.e. showing with mathematical certainty that a critical piece of code exhibits the intended behavior. My work has been covered by Quanta Magazine, IEEE Explore, Communications of the ACM, The Register, and has received the Internet Defense Prize as well as a SIGPLAN Research Highlight.

Selected Projects

- EverCrypt/HACL*: a no excuses, industrial-grade cryptographic library that combines C and assembly code to provide a fully verified collection of algorithms (used in Linux, Firefox, Python, Windows, the Tezos blockchain and many more).
- Aeneas: a new verification toolchain for programs written in safe Rust
- Eurydice: a compiler from Rust to readable C, to ease the transition to safe languages

My work straddles theory and practice: I advise PhD students, maintain research collaborations with several universities, but also drive concrete projects and hack on large amounts of code, verified or not.

Work Experience

Sep 2025 – present
Tech Lead manager at **Google** in the Information Security Engineering division (Seattle, WA).

Jan. 2024 – Sep 2025
Researcher at **Microsoft** in the Azure Research team (Redmond, WA).

Sept. 2016 – Dec. 2023
Researcher at **Microsoft** in the RiSE team (Redmond, WA).

Sept. 2014 – Sept. 2016
Post-doctoral researcher at **Microsoft** in the RiSE team (Redmond, WA).

Aug. 2010 – 2012
Core contributor and module owner for the Mozilla project, authoring patches and reviewing contributions for a core component of Mozilla Thunderbird. Two summer internships in Vancouver, BC working on improvements to the Thunderbird mail client, followed by contracting. I spoke about Mozilla or Thunderbird at numerous events (FOSDEM, 4+ Mozilla Summits) and I still maintain one the largest Thunderbird addons.

Aug. 2009
Three-month internship at LexiFi (http://www.lexifi.com), a finance software editor. I redesigned LexiFi's GUI technology using their custom version of OCaml, and worked on dynamic types with first-class modules.

Nov. 2005 – 2006
While in high school I wrote, with Benoît Picaud, under the supervision of Muriel Shan Sei Fan (editor) "Les Cahiers du Programmeur XUL", a French book about the XUL programming language, a Mozilla technology for building graphical user interfaces. The book was later translated to German.

Educational Background

Sept. 2010 – Sept. 2014
At **INRIA** (http://www.inria.fr), the French Research Institute for Computer Science.
Ph.D. in the Gallium team under the direction of François Pottier.

Sept. 2007 – June 2010
At **ÉNS Lyon** (http://www.ens-lyon.fr/) – a selective science university which trains future researchers and teachers.

- 2008-2010: Obtained a Master's degree in Computer Science with high honors. One semester was spent as an exchange student at the National University of Singapore and another semester doing a research internship at INRIA, in the Gallium team.

- 2007-2008: Obtained a Bachelor's degree in Computer Science, ranked 1st.

September 2005 – August 2007

At **Lycée Michel-Montaigne**, Bordeaux
Two years of top-level, maths and physics oriented, undergraduate courses in order to prepare for the highly selective entrance exams to higher studies.

SELECTED PUBLICATIONS (complete list available at https://jonathan.protzenko.fr)

On the practice of verified software:

- **TreeSync: Authenticated Group Management for Messaging Layer Security.** (Usenix Security'23). T. Wallez, J. Protzenko, B. Beurdouche, K. Bhargavan. ***Distinguished Paper Award, Internet Defense Prize***.
- **Noise*: A Library of Verified High-Performance Secure Channel Protocol Implementations.** (S&P'22). S. Ho, J. Protzenko, A. Bichhawat, K. Bhargavan.
- **EverCrypt: A Fast, Verified, Cross-Platform Cryptographic Provider**. (S&P'20). J. Protzenko et. al.
- **Formally Verified Cryptographic Web Applications in WebAssembly** (S&P'19). J. Protzenko et. al.
- **HACL*: A Verified Modern Cryptographic Library**. (CCS'17). J-K. Zinzindohoué, K. Bhargavan, J. Protzenko, B. Beurdouche.

On the theory of verified software:

- **Aeneas: Rust Verification by Functional Translation.** (ICFP'22). S. Ho, J. Protzenko
- **Verified Low-Level Programming Embedded in F*.** (ICFP'17). J. Protzenko et. al.

On language design and type systems:

- **Catala, a Programming Language for the Law.** (ICFP'21). D. Merigoux, N. Chataing, J. Protzenko
- **The Design and Formalization of Mezzo**, (TOPLAS'15). T. Balabonski, F. Pottier, J. Protzenko
- **Programming with permissions in Mezzo**, **(**ICFP'13). F. Pottier, J. Protzenko

PHD ADVISING

- Denis Merigoux (2018-2021), co-advised with K. Bhargavan at INRIA. ***Gilles Kahn PhD award***, 2022. ("Prix de thèse Gilles Kahn")
- Son Ho (2020-2024), co-advised with K. Bhargavan at INRIA. Shortlisted for ***GDR-GPL PhD award***, 2025.
- Théophile Wallez (2021-), co-advised with K. Bhargavan at INRIA

TEACHING

2024: Distinguished lecture at NUS on crypto and protocol verification
2019: Dagstuhl Summer School on Meta Programming (two lectures)
2018: invited lecture at the university of Cambridge on Meta Programming
Summer School on Formal Techniques at SRI
Low* tutorial @ PLDI
2017: invited lecture in the Cryptographic protocols formal and computational proofs class at MPRI (Paris)

SERVICE

Program Committees

2026: CCS, CSF, 2025: CC, CPP, CSF, 2024: POPL, CSF, ICFP, 2023: ICAIL, ESOP, CSF, TyDE@ICFP, OCaml@ICFP, 2022: CSF, CRCL, 2021: ICFP (external), OCaml@ICFP, 2020: ICFP, ML@ICFP, PriSC@POPL, CPP (sub-reviewer), 2019: ICFP (ERC), POPL (AEC), 2018: OCaml@ICFP, 2017: ESOP (sub-reviewer), 2016: Mobile!@OOPSLA, 2015: ML@ICFP

Chair / Organizer

2026: ICFP (Industrial Relations chair), 2024: VSTTE 2024 (co-chair), 2023: ProLaLa@POPL (co-chair), Dagstuhl Seminar on WebAssembly (co-organizer), 2022: ProLaLa@POPL (co-chair), PriSC@POPL (co-chair, with Marco Guarnieri), 2021: ML@ICFP (chair), PriSC@POPL (co-chair, with Deian Stefan), 2020: HASE@POPL (co-organizer)