Vincent LAPORTE

Academic Positions and Employment

since 01/2020 Researcher (CRCN) at Inria, Nancy, in the Pesto team

09/2018–12/2019 Engineer at Inria, Rennes; working on the Coq proof assistant

01/2016–08/2018 Post-doc at IMDEA Software Institute, Madrid (España), supervised by Gilles Barthe

2012–2015 Ph.D. at University of Rennes 1, France, supervised by Sandrine Blazy and David Pichardie
Subject: verification of static analyses for low-level languages

09–11/2014 Internship at Microsoft Research, Cambridge (UK), supervised by Cédric Fournet
Subject: a CompCert cryptographic back-end for verifiable computation

2011–2012 Pre-Doc at Purdue University, Indiana, supervised by Jan Vitek
As a 4th year student of the *École Normale Supérieure de Cachan, Antenne de Bretagne*
Subject: formal verification of a Java compiler

02–06/2011 Internship at inria, Rennes
Supervised by Guillaume Hiet, Sandrine Blazy and David Pichardie
Subject: static analysis of x86 executables; certification and robustness analysis

06–08/2009 Internship at the imdea, Madrid, supervised by Gilles Barthe
Keywords: cryptography; formal proof; probability distributions; Coq; CertiCrypt

06–07/2008 Internship at the *Laboratoire de l'Informatique du Parallélisme*, inria, (éns Lyon, France)
Supervised by Jean Duprat
Subject: Coq–GeoGebra Interface; designing a tool demonstrated at Types 2009.
Keywords: formal proof; planar geometry; Coq; GeoGebra

Education and Qualifications

2017 *Qualification aux fonctions de maître de conférences, CNU section 27*

2012–2015 Ph.D. at University of Rennes 1, France, supervised by Sandrine Blazy and David Pichardie
Subject: verification of static analyses for low-level languages

2010–2011 5th year in 5yr. post-secondary diploma in Computer Science (*Master d'informatique*)
École Normale Supérieure de Cachan, Antenne de Bretagne; University of Rennes 1, France
With honors (*mention bien*)
Keywords: research; verification; machine learning

2008–2009 4th year in 5yr. post-secondary diploma in Computer Science (*Master d'informatique*)
École Normale Supérieure de Cachan, Antenne de Bretagne; University of Rennes 1, France
With honors (*mention bien*)
Keywords: research; signal processing; optics; electronics; compilation; semantics; formal methods; distributed computing

2007–2008 3yr. post-secondary diploma in Computer Science (*Licence d'informatique*)
École Normale Supérieure de Cachan, Antenne de Bretagne; University of Rennes 1, France
With honors (*mention très bien*)
Keywords: electromagnetism; computability; logic; algorithmics; architecture; data structures; operating systems; image processing

2005–2007 Competitive 2yr. cycle specialising in Mathematics. *Lycée Pierre de Fermat*, Toulouse, France

2005 *Baccalauréat* in Mathematics, English European Section; with honors (*mention très bien*)
Lycée Marcellin Berthelot, Toulouse, France

Publications

International Journals

1. Sandrine Blazy, Vincent Laporte, and David Pichardie. "Verified Abstract Interpretation Techniques for Disassembling Low-level Self-modifying Code". In: *Journal of Automated Reasoning* 56.3 (2016), pp. 283–308. DOI: 10.1007/s10817-015-9359-8.

2. Suresh Jagannathan, Vincent Laporte, Gustavo Petri, David Pichardie, and Jan Vitek. "Atomicity Refinement for Verified Compilation". In: *ACM Transactions on Programming Languages and Systems* (*TOPLAS*) (2014).

International Conferences

1. Basavesh Ammanaghatta Shivakumar, Gilles Barthe, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Swarn Priya, Peter Schwabe, and Lucas Tabary-Maujean. *Typing High-Speed Cryptography against Spectre v1.* Cryptology ePrint Archive, Paper 2022/1270. 2022. URL: https://eprint.iacr.org/2022/1270.

2. Basavesh Ammanaghatta Shivakumar, Gilles Barthe, Benjamin Grégoire, Vincent Laporte, and Swarn Priya. "Enforcing fine-grained constant-time policies". In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security,* (*CCS*). 2022. DOI: 10.1145/3548606.3560689.

3. Gilles Barthe, Benjamin Grégoire, Vincent Laporte, and Swarn Priya. "Structured Leakage and Applications to Cryptographic Constant-Time and Cost". In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security,* (*CCS*). 2021.

4. José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Vincent Laporte, and Tiago Oliveira. "Certified Compilation for Cryptography: Extended x86 Instructions and Constant-Time Verification". In: *21st International Conference on Cryptology in India.* 2020.

5. José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Benjamin Grégoire, Adrien Koutsos, Vincent Laporte, Tiago Oliveira, and Pierre-Yves Strub. "The Last Mile: High-Assurance and High-Speed Cryptographic Implementations". In: *41st IEEE Symposium on Security and Privacy,* (*S&P*). 2020.

6. Gilles Barthe, Sandrine Blazy, Benjamin Grégoire, Rémi Hutin, Vincent Laporte, David Pichardie, and Alix Trieu. "Formal Verification of a Constant-Time Preserving C Compiler". In: *Proceedings of the ACM on Programming Languages* (*POPL*) (2020).

7. José Bacelar Almeida, Cécile Baritel-Ruet, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Alley Stoughton, and Pierre-Yves Strub. "Machine-Checked Proofs for Cryptographic Standards: Indifferentiability of Sponge and Secure High-Assurance Implementations of SHA-3". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security,* (*CCS*). 2019.

8. Gilles Barthe, Benjamin Grégoire, and Vincent Laporte. "Secure compilation of side-channel countermeasures: the case of cryptographic 'constant-time'". In: *31st IEEE Computer Security Foundations Symposium,* (*CSF*). **Distinguished paper**. 2018.

9. José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Arthur Blot, Benjamin Grégoire, Vincent Laporte, Tiago Oliveira, Hugo Pacheco, Benedikt Schmidt, and Pierre-Yves Strub. "Jasmin: High-Assurance and High-Speed Cryptography". In: *Proceedings of the 24th ACM Conference on Computer and Communications Security,* (*CCS*). 2017.

10. José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, François Dupressoir, Benjamin Grégoire, Vincent Laporte, and Vitor Pereira. "A Fast and Verified Software Stack for Secure Function Evaluation". In: *Proceedings of the 24th ACM Conference on Computer and Communications Security,* (*CCS*). 2017.

11. Gilles Barthe, Sandrine Blazy, Vincent Laporte, David Pichardie, and Alix Trieu. "Verified Translation-Validation of Static Analyses". In: *30th IEEE Computer Security Foundations Symposium,* (*CSF*). 2017.

12. Sandrine Blazy, Vincent Laporte, and David Pichardie. "An Abstract Memory Functor for Verified C Static Analyzers". In: *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, (*ICFP*). 2016, pp. 325–337.

13. Cédric Fournet, Chantal Keller, and Vincent Laporte. "A Certified Compiler for Verifiable Computing". In: *29th IEEE Computer Security Foundations Symposium*, (*CSF*). 2016, pp. 268–280. DOI: 10.1109/CSF.2016.26.

14. Jacques-Henri Jourdan, Vincent Laporte, Sandrine Blazy, Xavier Leroy, and David Pichardie. "A Formally-Verified C Static Analyzer". In: *Proc. of the 42th Symp. on Princ. of Prog. Languages* (*POPL*). ACM, 2015.

15. Sandrine Blazy, Vincent Laporte, and David Pichardie. "Verified Abstract Interpretation Techniques for Disassembling Low-level Self-modifying Code". In: *Proc. of the 5th conference on Interactive Theorem Proving* (*ITP*). Lecture Notes in Computer Science. Springer-Verlag, 2014.

16. Sandrine Blazy, Vincent Laporte, André Maroneze, and David Pichardie. "Formal Verification of a C Value Analysis Based on Abstract Interpretation". In: *Proc. of the 20th Static Analysis Symposium* (*SAS*). Lecture Notes in Computer Science. Springer-Verlag, 2013.

17. Delphine Demange, Vincent Laporte, Lei Zhao, David Pichardie, Suresh Jagannathan, and Jan Vitek. "Plan B: A Buffered Memory Model for Java". In: *Proc. of the 40th Symp. on Princ. of Prog. Lang.* (*POPL*). ACM, 2013.

18. Gilles Barthe, Marion Daubignard, Bruce Kapron, Yassine Lakhnech, and Vincent Laporte. "On the equality of probabilistic terms". In: *Logic for Programming, Artificial Intelligence, and Reasoning* (*LPAR*). Springer-Verlag. 2010.

Contributions to software developments

Jasmin An infrastructure for high-assurance, high-speed cryptography. https://github.com/jasmin-lang/jasmin
A-3; SO-3; SM-2; EM-3; SDL-4 / DA-4; CD-4; MS-4; TPM-4[1] / 25k lines of Coq; 5k lines of OCaml
Main contributor. Relevant publication: CCS 2017.
This framework features a programming language and its compiler. The language is designed for enhancing portability of programs and for simplifying verification tasks. The compiler is designed to achieve predictability and efficiency of the output code (currently limited to x64 platforms), and is formally verified in the Coq proof assistant. The framework also includes highly automated tools for proving memory safety and constant-time security (for protecting against cache-based timing attacks).

Verasco A static analyzer for the CompCert C#minor language. http://compcert.inria.fr/verasco/
A-3; SO-3; SM-2; EM-1; SDL-2 / DA-4; CD-4; MS-4; TPM-3 / 34k lines of Coq; 6k lines of OCaml
Main contributor (with Jacques-Henri Jourdan). Relevant publications: SAS 2013, POPL 2015, ICFP 2016.
This analyzer establishes the absence of run-time errors in analyzed programs. It is based on abstract interpretation and combines several abstract domains, non-relational (integer intervals, floating-point intervals, integer congruences, points-to properties) and relational (integer linear inequalities, symbolic equalities). Verasco is entirely specified and proved sound using the Coq proof assistant: its proof guarantees, with mathematical certainty, that programs that analyze without alarms are free of run-time errors.

CT-Preservation Coq formalization of constant-time-simulations. https://sites.google.com/view/ctpreservation
A-1; SO-3; SM-1; EM-2; SDL-2 / DA-4; CD-4; MS-4 / 7k lines of Coq
Main contributor (with Benjamin Grégoire). Relevant publication: CSF 2018.

[1] Following Inria EC Criteria for Software Self-Assessment: Software characterization: A: audience; SO: software originality; SM: software maturity; EM: evolution and maintenance; SDL: software distribution and licensing. Own contribution: DA: design and architecture; CD: coding and debugging; MS: maintenance and support; TPM: team/project management.

CircGen A formally verified compiler from CompCert RTL intermediate language to circuit descriptions.
A-1; SO-3; SM-1; EM-1; SDL-1 / DA-3; CD-3
Main contributor (with Bacelar Almeida). Relevant publication: CCS 2017.
Verified back-end for the CompCert compiler, targeting boolean circuits.

PinocchioQ A Certified Compiler for Verifiable Computing.
A-1; SO-3; SM-1; EM-1; SDL-1 / DA-4; CD-4
Main contributor (with Chantal Keller). Relevant publication: CSF 2016.
Verified back-end for the CompCert compiler, targeting quadratic arithmetic programs.

Coq A formal proof management system. https://coq.inria.fr/
A-4; SO-4; SM-4; EM-4; SDL-5 / DA-2; CD-2; MS-2; TPM-1
Full-time developper from September 2018 to December 2019.

Nixpkgs A collection of packages for the Nix package manager. https://github.com/NixOS/nixpkgs/
A-4; SO-4; SM-4; EM-4; SDL-5 / DA-2; CD-3; MS-2; TPM-1
Regular contributor (one of the about two hundreds trusted people with write access to the main repository).
The *Nix Packages collection* holds tens of thousands of descriptions of programs and libraries for various Unix platforms. They can be used to manage (build, install…) these packages and their dependencies.

Teaching Experience

Student supervision

- Roméo La Spina (master's student, ÉNS Rennes, France), Proving that compilation preserves the "Speculative Constant-Time" security property, Spring 2021.
- Sai Vigna Surapaneni (undergraduate, IIT Bombay, India), Experimental validation of the x86 formal model used in the Jasmin compiler, Spring 2021.

Student co-supervision

- Prajeeth Sankaranarayanan (undergraduate, IIT Bombay, India), Design of a malicious ballot-box for Belenios, Spring 2021.
- Thibaut Pérami (undergraduate), Automated analyses of multi-party computations, Summer 2017.
- Arthur Blot (master's student), Verification of constant-time implementations, Spring 2017.

Teaching Assistant

Introduction to Logic Bachelor, Telecom Nancy, Fall 2021 (20 hours). Instructor: Sébastien Da Silva.

Introduction to Theoretical Computer Science (Logic, Languages, Automata) Bachelor, Telecom Nancy, Spring 2021 (42 hours). Instructor: Sébastien Da Silva.

From Fall 2012 to Spring 2015, I performed 64h per year of teaching assistant duties.

Introduction to UNIX Bachelor, ĒNS Rennes, Fall 2013, Fall 2014. Person in charge: Prof. Benoît Cadre

Object Oriented Programming Bachelor, ĒNS Rennes, Spring 2013, Spring 2014, Spring 2015.
Instructor: Alexandru Costan, INSA of Rennes.

Introduction to Computer Science Bachelor, ĒNS Rennes (dept. of Mathematics), Spring 2015.
Instructor: Prof. David Pichardie.

Formal Methods Master, University of Rennes, Fall 2012, Fall 2014. Instructor: Prof. Sandrine Blazy.

Introduction to Algorithmics Bachelor, University of Rennes, Fall 2013. Instructor: Gilles Lesventes.

Programming Language Semantics Master, University of Rennes, Fall 2012, Fall 2013. Instructor: David Cachera.

Selected Talks

Fine-Grained Constant-Time Policies with Jasmin & EasyCrypt. Invited talk. Le génie logiciel au service de la sécurité des systèmes, logiciels et réseaux, CNAM, Paris, Nov. 24, 2022. URL: https://glsec22.sciencesconf.org/.

High-Assurance Cryptography in Jasmin & Spectre Security. Invited talk. Team Cambium, Inria, Paris, Sept. 26, 2022. URL: https://cambium.inria.fr/seminaires/annonces/20220926.Vincent.Laporte.txt.

Jasmin: a Certified Workbench for High-Assurance and High-Speed Cryptography. Invited talk. The Coq Workshop 2021 (online), July 2, 2021. URL: https://coq-workshop.gitlab.io/2021/.

Modèle de fuite structurée & applications à la compilation sécurisée. Invited talk. Journées AFADL 2021 (online), June 16, 2021. URL: https://www.lirmm.fr/afadl2021/?page=programme.

Certified Compilation for High-Assurance Cryptography. Invited talk. 68NQRT seminar, Irisa, Rennes (online), Dec. 3, 2020. URL: http://68nqrt.inria.fr.

The Last Mile: High-Assurance and High-Speed Cryptographic Implementations. Accepted talk. 41st IEEE Symposium on Security and Privacy, (S&P), *online*, May 19, 2020. URL: https://www.ieee-security.org/TC/SP2020/program.html.

Constant-Time Programming: Formal Verification & Secure Compilation. Invited talk. Campus de Jussieu, Paris, Sept. 25, 2019. URL: https://www.cryptoexperts.com/verisicc/seminaire.html.

Jasmin: a workbench for high-assurance low-level programming. Invited talk. Team Pesto, Loria, Nancy, Jan. 21, 2019.

Secure compilation of side-channel countermeasures: the case of cryptographic 'constant-time'. **Distinguished paper**. Accepted talk. 31st IEEE Computer Security Foundations Symposium, (CSF), Oxford, July 12, 2018.

Secure compilation of side-channel countermeasures: the case of cryptographic "constant-time". Invited talk. Dagstuhl seminar 18201, Secure Compilation, May 18, 2018. URL: https://www.dagstuhl.de/de/programm/kalender/semhp/?semnr=18201.

Compilation sûre de contre-mesures aux attaques par canaux auxiliaires. Invited talk. GT Méthodes Formelles pour la Sécurité, Cachan, Feb. 7, 2018. URL: http://www.lsv.fr/~baelde/gtmfs/

Provably secure compilation of side-channel countermeasures. Invited talk. Prosecco Seminars, Inria, Paris, Dec. 14, 2017. URL: http://prosecco.gforge.inria.fr/events.php

Jasmin: High-Assurance and High-Speed Cryptography. Accepted talk. 24th ACM Conference on Computer and Communications Security, (CCS), Dallas, Nov. 2, 2017. URL: https://acmccs.github.io/session-H4/

Verified Translation Validation of Static Analyses. Software Seminar Series, Imdea Software Institute, Madrid, June 27, 2017.

A Certified Compiler for Verifiable Computing. Accepted talk. 29th IEEE Computer Security Foundations Symposium, (CSF), Lisbon, June 29, 2016.

Formal Verification of a C Value Analysis Based on Abstract Interpretation. Invited talk. Journées nationales du GDR-GPL, Paris, June 12, 2014. URL: http://gdr-gpl.cnrs.fr/node/129

Automatic Refinement for Verified Compilation. 6e rencontres de la communauté française de compilation, Annecy, Apr. 4, 2013. URL: http://compilfr.ens-lyon.fr/sixiemes-rencontres-de-la-communaute-francaise-de-compilation/

Participation in Funded Research Projects

Formally verified Key Management Services, funded by Amazon Web Services. April 2018-September 2018. Partners: IMDEA, Inria, HASLab. Funding: 0.4 M€.

SynCrypt: Synthesis in Cryptography, September 2015-August 2018, funded by the U.S. Office of Naval Research (USA). Partners: IMDEA, Stanford U., U. Pennsylvania. Goal: develop automated tools for cryptographic implementations. Funding: IMDEA 1 M$.

Verified and Efficient Elliptic Curve Cryptography, funded by Google. December 2016–November 2017. Funding: 0.1 M€.

Verasco: formal verification of static analyzers and of compilers, 2012–2015, funded by *Agence nationale de la recherche* (grant ANR-11-INSE-003). Partners: Inria, Airbus, Université Rennes 1, Verimag.

External Reviews

35th IEEE Computer Security Foundations Symposium (CSF) August 7–10, 2022, Haifa, Israel.

Journal of Functional Programming Special Issue on Secure Compilation, Cambridge University Press, 2021.

34th IEEE Computer Security Foundations Symposium (CSF) June 21–25, 2021, Dubrovnik, Croatia.

35th ACM/IEEE Symposium on Logic in Computer Science (LICS), July 8–11, 2020.

25th ACM Conference on Computer and Communications Security (CCS), October 15–19, 2018.

31st IEEE Computer Security Foundations Symposium (CSF), July 9–12, 2018.

27th International Conference on Compiler Construction (CC), February 24–25, 2018.

22nd European Symposium on Research in Computer Security (ESORICS), September 11–15, 2017.

26th European Symposium on Programming (ESOP), April 22–29, 2017.

23rd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), April 22–29, 2017.

26th International Conference on Compiler Construction (CC), February 5–6, 2017.

14th Asian Symposium on Programming Languages and Systems (APLAS), November 21–23, 2016.

21st International Symposium on Formal Methods (FM), November 9–11, 2016.