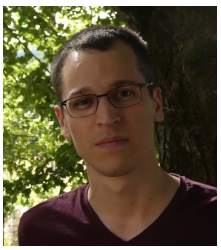
Inria Paris, C225
2 Rue Simone Iff, 75012 Paris
France
+33 6 99 29 05 96
adrien.koutsos@inria.fr
Born on May the 30th, 1992
in Echirolles, France
https://adrienkoutsos.fr



Adrien Koutsos

Curriculum Vitae

I am a researcher at Inria Paris, in the Prosecco team. I am interested in the application of **formal methods** in **security**. I work on proofs of security **protocols** (in particular of authentication and privacy properties), the application of **automated deduction** techniques to help protocol analysis, and the usage of **static analysis** in security.

Employment

Oct. 2020– **Researcher (CR)**, Inria Paris, Prosecco Team, Paris, France

Sept. 2019–Sept. 2020 **Post-doctoral Researcher**, Max Planck Institute for Security and Privacy, Gilles Barthe's group, Bochum, Germany

Sept. 2016–Aug. 2019 **PhD Student and Teaching Assistant**, École Normale Supérieure de Paris-Saclay, Cachan, France

Education

2016–2019 **PhD**, École Normale Supérieure de Paris-Saclay, under the supervision of Hubert Comon
Title: Symbolic Proofs of Computational Indistinguishability
Jury: Catuscia Palamidessi (president), Cas Cremers (reviewer), Bogdan Warinschi (reviewer), Bruno Blanchet, Myrto Arapinis, Hubert Comon

Sept. 2015–June 2016 **Pre-Doctoral Research Internship**, CISPA, Saarland University, Germany, with Matteo Maffei

2013–2015 **Master**, *MPRI (Parisian Master in Computer Science Research)*, École Normale Supérieure de Cachan
Honors: magna cum laude, Ranking : 8/61

2012–2013 **L3 (Bachelor)**, *Computer Science*, École Normale Supérieure de Cachan
Honors: magna cum laude

2010–2012 **Classe préparatoire aux grandes écoles**, *Lycée Champollion*, Grenoble
Major : Mathematics

Prizes

2022 **Distinguished Paper Award**
David Baelde, Stéphanie Delaune, Adrien Koutsos, Solène Moreau:
Cracking the Stateful Nut: Computational Proofs of Stateful Security Protocols using the Squirrel Proof Assistant.
In IEEE CSF 2022.

2020 **PhD thesis — distinction from "GdR Sécurité Informatique"**

2019 **STIC Doctoral School Best Scientific Contribution Award**, *first prize*, for Adrien Koutsos. The 5G-AKA authentication protocol privacy. In IEEE EuroS&P 2019

Publications

International Peer-Reviewed Journals

[1] Manuel Barbosa, Gilles Barthe, Benjamin Grégoire, Adrien Koutsos, and Pierre-Yves Strub. Mechanized proofs of adversarial complexity and application to universal composability. *ACM Trans. Priv. Secur.*, 26(3), jul 2023.

[2] Adrien Koutsos. Decidability of a sound set of inference rules for computational indistinguishability. *ACM Trans. Comput. Log.*, 22(1):3:1–3:44, 2021.

[3] Adrien Koutsos and Victor Vianu. Process-centric views of data-driven business artifacts. *J. Comput. Syst. Sci.*, 86:82–107, 2017.

International Peer-Reviewed Conferences

[4] David Baelde, Adrien Koutsos, and Justine Sauvage. Foundations for cryptographic reductions in CCSA logics. In *CCS*, page to appear. ACM, 2024.

[5] D. Baelde, C. Fontaine, A. Koutsos, G. Scerri, and T. Vignon. A probabilistic logic for concrete security. In *CSF*, pages 324–339. IEEE, 2024.

[6] David Baelde, Adrien Koutsos, and Joseph Lallemand. A higher-order indistinguishability logic for cryptographic reasoning. In *LICS*, pages 1–13, 2023.

[7] David Baelde, Stéphanie Delaune, Adrien Koutsos, and Solène Moreau. Cracking the stateful nut: Computational proofs of stateful security protocols using the squirrel proof assistant. In *CSF*, pages 289–304. IEEE, 2022. **Distinguished Paper Award**.

[8] Gilles Barthe, Adrien Koutsos, Solène Mirliaz, David Pichardie, and Peter Schwabe. Semantic foundations for cost analysis of pipeline-optimized programs. In *SAS*, volume 13790 of *Lecture Notes in Computer Science*, pages 372–396. Springer, 2022.

[9] Manuel Barbosa, Gilles Barthe, Benjamin Grégoire, Adrien Koutsos, and Pierre-Yves Strub. Mechanized proofs of adversarial complexity and application to universal composability. In *CCS*, pages 2541–2563. ACM, 2021.

[10] Gilles Barthe, Sunjay Cauligi, Benjamin Grégoire, Adrien Koutsos, Kevin Liao, Tiago Oliveira, Swarn Priya, Tamara Rezk, and Peter Schwabe. High-assurance cryptography in the spectre era. In *IEEE Symposium on Security and Privacy*, pages 1884–1901, San Fransisco / Virtual, United States, May 2021. IEEE.

[11] David Baelde, Stéphanie Delaune, Charlie Jacomme, Adrien Koutsos, and S. Moreau. An interactive prover for protocol verification in the computational model. In *IEEE*

Symposium on Security and Privacy, pages 537–554, San Fransisco / Virtual, United States, May 2021. IEEE.

[12] 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 *IEEE Symposium on Security and Privacy*, pages 965–982. IEEE, 2020.

[13] Adrien Koutsos. The 5G-AKA authentication protocol privacy. In *IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019*, pages 464–479. IEEE, 2019.

[14] Adrien Koutsos. Decidability of a sound set of inference rules for computational indistinguishability. In *32nd IEEE Computer Security Foundations Symposium, CSF 2019, Hoboken, NJ, USA, June 25-28, 2019*, pages 48–61. IEEE, 2019.

[15] Hubert Comon and Adrien Koutsos. Formal computational unlinkability proofs of RFID protocols. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 100–114, 2017.

[16] Stefano Calzavara, Ilya Grishchenko, Adrien Koutsos, and Matteo Maffei. A sound flow-sensitive heap abstraction for the static analysis of android applications. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 22–36, 2017.

[17] Adrien Koutsos and Victor Vianu. Process-centric views of data-driven business artifacts. In Marcelo Arenas and Martín Ugarte, editors, *18th International Conference on Database Theory, ICDT 2015, March 23-27, 2015, Brussels, Belgium*, volume 31 of *LIPIcs*, pages 247–264. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

Invited Articles

[18] David Baelde, Stéphanie Delaune, Charlie Jacomme, Adrien Koutsos, and Joseph Lallemand. The squirrel prover and its logic. *ACM SIGLOG News*, 11(2):62–83, 2024.

Community Service

Editorial board, ACM TOPS, since 2024

PC member, CSF'25, CCS'24, CCS'23

External reviewer, ACM TOPS (journal), JCS (journal), EuroCrypt'21, ESORICS'20, IEEE TDSC'20, IEEE CSF'20, C2SI'19, IEEE POST'18

Examiner, oral examiner for "épreuve pratique d'algoritmique et de programmation" of the ENS competitive examen
2 years (2024, 2023)

Event organisation, Co-organised the 2024 Annual Meeting of the Working Group "Formal Method and Security" (GT-MFS, associated to the GdR Security and Privacy of the CNRS)

Participation in Scientific Events and Projects

Current project: PEPR SVP

Past projects (as external member): ANR Tecap and ANR Sequoia

Speaker in Summer Schools

July 2022 **Cyber in Nancy**, Nancy, program
Formal Proofs of Cryptographic Protocols in Squirrel (with David Baelde)

Software Implementations

- **Squirrel**, *Main Developer*, Squirrel is an interactive proof assistant dedicated to the formal verification of cryptographic protocols in the computational model. It is based on a higher-order probabilistic logic which supports generic mathematical reasoning as well as cryptographic-specific reasoning.
  Concretely, Squirrel allows to specify security protocol in a variant of the applied pi-calculus, and properties of those protocols using its probabilistic logic. Then, the properties are to be proved by the users through tactics. Squirrel supports protocols with unbounded replication and persistent state, and supports both correspondence (e.g. authentication) and indistinguishability properties (e.g. strong secrecy, unlinkability).
  Licence: MIT
  Code: https://github.com/squirrel-prover/squirrel-prover/
  Website: https://squirrel-prover.github.io/
- **Jasmin**, *Contributor*, Jasmin is a language and a compiler designed to write high-assurance and high-speed implementations of cryptographic primitives. The Jasmin programming language smoothly combines high- and low-level constructs. It is highly predictible, and allow programmers to control many low-level details that are performance-critical (e.g. instruction selection and scheduling, registers spilling, etc). At the same time, they can also rely on high-level abstractions (variables, functions, arrays, loops, etc.) to structure their code and make it more amenable to formal verification. Its semantics is formally defined, and the compiler is proved semantics-preserving in Coq. Jasmin programs can be automatically checked for safety and termination using a dedicated static analyzer.
  I added a static analyser in the Jasmin tool-chain to prove automatically the safety of Jasmin programs. More precisely, the analyser proves the absence of runtime errors (e.g. division by zero or out-of-bound array access), and automatically infers a relational memory calling contract of the program, i.e. symbolic ranges where memory access can safely take place. It relies on abstract interpretation techniques. The analyser has been used to prove the safety of some highly optimized cryptographic primitives (e.g. ChaCha20 and Poly1305).
  Code: https://github.com/jasmin-lang/jasmin/tree/array_cast
  Instructions to run the static analyser: https://github.com/akoutsos/jasmin-safety

- **EasyCrypt**, *Contributor*, EasyCrypt is a proof-assistant for reasoning about relational properties of probabilistic programs with adversarial code. Its main application is the construction and verification of cryptographic designs. EasyCrypt implements several program logics (probabilistic relational Hoare logic, cost logic, ...), built on top of an ambient higher-order logic.
  I extended the tool to allow to reason about EasyCrypt's programs complexity, using an extended Hoare logic. EasyCrypt programs can contain adversarial components (i.e. unknown code), which are modeled using abstract functors (morally, these can be instantiated by any concrete functor with the correct type). Consequently, to bound the complexity of such programs, we must be able to restrict an abstract functor's instantiations. To do this, I modified EasyCrypt type system and type-checking, by adding *restrictions* to functors types.
  Licence: MIT
  The webpage: https://www.easycrypt.info/trac/
  Code: https://github.com/EasyCrypt/easycrypt/
- **HornDroid**, *Contributor*, HornDroid is an information-flow analyser for Android applications. It is a static analyser, which translates a program to a set of Horn clauses over-approximating the program semantics. It relies on the SMT solver Z3.
  I improved the precision of the HornDroid static analyser by implementing a flow-sensitive abstraction of the memory heap. My improvement allows HornDroid to perform strong updates on heap-allocated data structures, increasing its precision, without losing soundness (even for concurrent applications).
  https://github.com/ylya/horndroid

Conference Talks

2022 **IEEE Computer Security Foundations Symposium**, *CSF 2022*
Cracking the Stateful Nut: Computational Proofs of Stateful Security Protocols using the Squirrel Proof Assistant

2021 **ACM Conference on Computer and Communications Security**, *CCS 2021*
Mechanized Proofs of Adversarial Complexity and Application to Universal Composability

2019 **IEEE European Symposium on Security and Privacy**, *EuroS&P 2019*
The 5G-AKA Authentication Protocol Privacy

2019 **32nd IEEE Computer Security Foundations Symposium**, *CSF 2019*
Decidability of a Sound Set of Inference Rules for Computational Indistinguishability

2017 **30th IEEE Computer Security Foundations Symposium**, *CSF 2017*
Formal Computational Unlinkability Proofs of RFID Protocols

2015 **18th International Conference on Database Theory**, *ICDT 2015*
Process-Centric Views of Data-Driven Business Artifacts

Other Talks

2024 **Mechanizing and Automating Cryptographic Arguments**
- **invited talk:** ProTeCS workshop (affiliated to Eurocrypt'24), ETH Zurich, 24/05/2024

2023 **Verifying Cryptographic Protocols**
- Inria Paris Seminar: "demi-heure de science", Inria, Paris, 09/11/2023

2022 – 2023 **Mechanized Proofs of Adversarial Complexity and Application to UC**
- **invited talk:** SCOT seminar (webpage), virtual, 15/12/2023
- Annual Meeting of the French Working Group on Formal Method, Fréjus, 23/03/2022

2019 **The 5G-AKA Authentication Protocol Privacy**

- STIC Doctoral School Best Scientific Contribution Award, CentraleSupelec Paris-Saclay, 28/11/2019
- Pesto Team Seminar, LORIA, Nancy, 21/11/2019
- Prosecco Team Seminar, Inria, Paris, 05/11/2019
- Grace Team Seminar, LIX, École Polytechnique, Palaiseau, 21/05/2019
- SoSySec Seminar, IRISA, Rennes, 18/01/2019

2019 **High-Assurance and High-Speed Cryptographic Implementations Using the Jasmin Language**

- Celtic Team Seminar, IRISA, Rennes, 09/10/2019

2018 **Deciding Indistinguishability: A Decision Result for a Set of Cryptographic Game Transformations**

- TECAP Seminar, ANR Project, LSV, Cachan, 14/03/2018

2017 **Formal Computational Unlinkability Proofs of RFID Protocols**

- SEQUOIA Seminar, ANR Project, LSV, Cachan, 06/03/2017

Teaching Activities

2022-2023 **Vacations**, *UPMC (P6) and Université Paris-Cité (P7)*

- Functional Programming, tutorial sessions and lectures (CM + TD), L2
  Using the Ocaml language
- MPRI 2.30: Proofs of security protocols, lectures (CM), M2

2021-2022 **Vacations**, *UPMC (P6) and Université Paris-Diderot (P7)*

- Introduction to C Programming, tutorial sessions (TD), L1
- Functional Programming, tutorial sessions (TD), L2
  Using the Ocaml language
- MPRI 2.30: Proofs of security protocols, lectures (CM), M2

2020-2021 **Vacations**, *UPMC (P6)*

- Introduction to C Programming, tutorial sessions (TD), L1
- Logic, tutorial sessions (TD), L2
  Syntax and semantics of FO logic, natural deduction.

2018-2019 **Teaching Assistant**, *Université Paris-Diderot (P7)*

- Initiation to Programming, practical sessions, L1
  Basics of programming, using the Python language.
- Languages and Automatas, tutorial sessions, L2
  Regular languages and automatas, algorithms (Thompson, Glushkov, Moore, Brozozwksi), residual, basics of context-free languages and pushdown automatas.

2016-2018 **Teaching Assistant**, *École Normale Supérieure de Paris-Saclay*

- Computability theory, tutorial sessions, L3
  Turing machines, reductions, undecidability (halting problem, paving, PCP).
- Complexity theory, tutorial sessions, L3
  Complexity classes (NL, P, NP, PSPACE), complete problems, reductions.
- Probabilistic Aspects of Computer Science, tutorial sessions, M1
  Markov chains, Markovian Decision Processes, probabilistic automatas, stochastic games.

Research Internships

2015-2016 **Pre-Doctoral Research Internship (ARPE)**, *CISPA*, Saarland University, Germany, 9 months, with Matteo Maffei

I worked on a flow-sensitive static analyser for Android applications, using abstract interpretation techniques. This allows to prove the absence of security leaks.

2015 **Internship**, *LSV*, Cachan, 4.5 months, with Hubert Comon

I worked on an automated deduction algorithm to prove equivalence properties in the Bana-Comon model.

2014 **Internship**, *UCSD*, San Diego, California, 4.5 months, Victor Vianu

I worked in database theory and verification, on the regularity of views of data-centric workflows.

2013 **Short Internship**, *Verimag*, Grenoble, 2 months, with Pascal Lafourcade

I worked on automatic proofs of IND-CPA security property for cryptographic schemes, using a Hoare logic.

Scientific Vulgarization

26/01/2018 **Speaker at the Workshop on the protection of young people online**, *École Normale Supérieure de Paris-Saclay*

Title: "Authentification: application aux mineurs"