David Cock

October 19, 2022

ETH Zürich
CAB E 77.1
Universitätstrasse 6
8092 Zürich
Switzerland

Telephone, mobile: +41 79 897 11 17
Telephone, work: +41 44 632 66 49
david.cock@inf.ethz.ch
davidcock@fastmail.fm

Education & Employment

2018– *Dozent (Lecturer), Institute for Computing Platforms (Systems Group), D-INFK, ETH Zürich*

2017– *Oberassistent (Senior Researcher), Institute for Computing Platforms (Systems Group), D-INFK, ETH Zürich*

2015–2017 *Postdoctoral researcher, Institute for Computing Platforms (Systems Group), D-INFK, ETH Zürich*

Supervisor Timothy Roscoe.

2014 *PhD, University of New South Wales & National ICT Australia*

"Leakage in Trustworthy Systems", supervisor Gernot Heiser, software systems research group (SSRG).

2005–2009 *PhD-track Research Engineer, National ICT Australia*

L4.verified & seL4 projects, supervisor Gerwin Klein.

2004 *BSc (Hons), University of New South Wales*

Mathematics & Computer Science, major in Systems and Algebra.

Research

The L4.verified project[1] produced the first full correctness proof of a general-purpose operating system (micro-)kernel: seL4[2] (Derrin et al., 2006; Klein et al., 2009, 2010). I was the principal author of the final implementation, which achieved the fastest IPC (inter-process communication) operations ever published on the ARM architecture. The nondeterministic, monadic refinement framework used for the proof was described in Cock et al. (2008).
Roughly 10% of both the C code of seL4, and of the lines of proof script dealt with packed structure manipulation, and were automatically generated (Winwood et al., 2009), using a custom DSL (domain-specific language) compiler (Cock, 2008), which co-generates Isabelle/HOL proof script.
My high-performance, retargetable CPU/system simulator, Lyrebird (Cock, 2010), provided a prototyping platform with an automatically-generated formal model.
My dissertation (Cock, 2014a) dealt with the detection and mitigation of covert and side channels in component systems. We developed techniques for detecting, modelling and mitigating compromising channels in security-critical software, in particular both the empirical evaluation of

[1] http://www.ertos.nicta.com.au/research/l4.verified/
[2] http://ssrg.nicta.com.au/projects/seL4/

hardware-based channel capacity, and the formal verification of probabilistic security properties (Cock, 2011, 2013, 2014b). Our empirical results have been published (Cock et al., 2014; Ge et al., 2018).
My published formalisation of the probabilistic programming logic/refinement framework pGCL(Cock, 2012) was used to machine check these proofs, and has since been accepted to the archive of formal proofs(Cock, 2014c).
My ongoing work includes: Establishing a formally trustworthy, yet precise model of the hardware software interface (Achermann et al., 2017, 2018), both by formal modelling and real-time verification using hardware trace data. Building Enzian, an open-source CPU-FPGA hybrid research computer[3].

Teaching

2018– *Lecturer in Charge, Informal Methods, ETH Zürich*

Formal methods are increasingly a key part of the methodological toolkit of systems programmers - those writing operating systems, databases, and distributed systems. This course is about how to apply concepts, techniques, and principles from formal methods to such software systems, and how to get into the habit of thinking formally about systems design even when writing low-level C code.

2015– *Lecturer, Advanced Operating Systems, ETH Zürich*

This is a Masters-level capstone course in the Systems curriculum at ETH, based on the UNSW model.

2009–2014 *Tutor, Operating Systems and Advanced Operating Systems courses, UNSW*

Small-group instruction of both undergraduate and postgraduate students, and marking duties. Individual guidance and assessment for advanced student projects.

2002–2004 *Consultant & Replacement Tutor, Higher Computing 1A, UNSW*

Individual tuition and guidance for first-year undergraduate students, and supervision of advanced student projects.

Supervised Bachelors/Masters Theses

2019

- Pirmin Schmid. Runtime Verification with TeSSLA on Enzian.

2017

- Daniel Schwyn. Hardware Configuration With Dynamically-Queried Formal Models (Masters).
- Mickey Vänskä. Program Trace Analysis on an FPGA (Bachelors).

[3] `http://enzian.systems/`

2016
- Andrei Pârvu. Program Trace Capture and Analysis for ARM (Masters).
- Marc Tanner. A Debugging Interface for Barrelfish (Bachelors).
- Nicole Thunherr. A Survey of Hardware Assumptions in Contemporary Systems (Bachelors).

2015
- Martynas Pumputis. Message Passing for Programming Languages and Operating Systems (Masters).
- Claudio Foellmi. OS Development in Rust (Masters).

Industrial Experience

2005–2014 *Computer Support Officer, National ICT Australia.*

Technical and performance-optimisation support for the automated regression testing of large mechanised proofs. Resource planning and equipment acquisition.

2003–2005 *Programmer, Brain Resource Pty. Ltd.*

Developed a clinical electroencephalogram acquisition system. Development was primarily in Python and C, on a customised Debian distribution. This system was deployed worldwide.
This system included a customised low-latency audio driver, with experimentally-verified jitter bounds.

Invited, Industrial & Public Talks

2018
- "Modeling the OS/Hardware Interface with Sockeye", ENTROPY 2018, IRCICA, Villeneuve d'Ascq, France (invited).

2017
- "New Projects at ETH Systems", Data61 (CSIRO), Sydney, Australia.
- "Runtime Verification", ARM Research Summit, Robinson College, Cambridge, UK.
- "The Impact of Incomprehensible Hardware on Security", INRIA Rennes - Bretagne Atlantique, Rennes, France (invited).

2016
- "FPGAs as Tools and Architectures at ETH Systems", Xilinx Dublin, Ireland.
- "ARM at ETH Systems", ARM Research Summit, Churchill College, Cambridge, UK.

2014
- "How to navigate the literature", NICTA SSRG PhD student boot camp, UNSW, Sydney, Australia.
- "Measuring and Mitigating Side Channels", NICTA software systems summer school, Sydney, Australia.

2013
- "Lyrebird — A Retrospective", Cambridge Computer Laboratory, Cambridge, UK.

Grants & Awards

2014 *CISRA Best Research Paper Award (UNSW)*

"The Last Mile: An Empirical Study of Timing Channels on seL4" (Cock et al., 2014)

2009–2013 *Australian Postgraduate Award*

A competitive federally-funded full scholarship for research students.

2009–2013 *NICTA Research Project Award*

A competitive scholarship for students undertaking project work at NICTA.

2009–2013 *UNSW Engineering Top-Up Scholarship*

Limited numbers offered annually, awarded for teaching work.

Scientific Engagement

Journal Reviewing

- Elsevier—Science of Computer Programming.
- ACM—Journal of the ACM.
- Springer—Design Automation for Embedded Systems.

Conference/Workshop Reviewing

- ACM—EMSOFT 2019 PC, EMSOFT 2020 PC, EMSOFT 2021 PC.
- Springer LNCS—European Symposium on Programming; International Symposium on Formal Methods, Security Proofs for Embedded Systems; International Symposium on Automated Technology for Verification and Analysis
- USENIX—OSDI, ATC 2019 PC, ATC 2021 PC, ATC 2022 PC, ATC 2023 PC.
- Elsevier—Information and Communication (GandALF 2013).
- IEEE—RTAS 2020 PC.
- Eurosys 2017 Shadow PC.

References

Gernot Heiser Telephone: +61 2 9490 5850
Scientia Professor, John Lions Chair of Operating Systems, UNSW Australia gernot.heiser@data61.csiro.au

Gerwin Klein Telephone: +61 2 9490 5878
Chief Principal Research Scientist, CSIRO Australia gerwin.klein@data61.csiro.au

Timothy Roscoe Telephone: +41 44 632 88 40
Professor, Department of Computer Science, ETH Zürich troscoe@inf.ethz.ch

Publications

Reto Achermann, Lukas Humbel, David Cock, and Timothy Roscoe. Formalizing memory accesses and interrupts. In Holger Hermanns and Peter Höfner, editors, *Proceedings of the 2nd Workshop on Models for Formal Analysis of Real Systems*, volume 244 of *Electronic Proceedings in Theoretical Computer Science*, pages 66–116. Open Publishing Association, April 2017. doi:10.4204/EPTCS.244.4.

Reto Achermann, Lukas Humbel, David Cock, and Timothy Roscoe. Physical addressing on real hardware in Isabelle/HOL. In Jeremy Avigad and Assia Mahboubi, editors, *Proceedings of the 9th International Conference on Interactive Theorem Proving*, volume 10895 of *Lecture Notes in Computer Science*, pages 1–19. Springer, July 2018. doi:10.1007/978-3-319-94821-8_1.

Reto Achermann, Nora Hossle, Lukas Humbel, Daniel Schwyn, David Cock, and Timothy Roscoe. A least-privilege memory protection model for modern hardware. 2019. doi:10.48550/ARXIV.1908.08707. eprint.

Reto Achermann, David Cock, Roni Haecki, Nora Hossle, Lukas Humbel, Timothy Roscoe, and Daniel Schwyn. Generating correct initial page tables from formal hardware descriptions. In *Proceedings of the 11th Workshop on Programming Languages and Operating Systems*, page 69–75. ACM, October 2021a. doi:10.1145/3477113.3487270.

Reto Achermann, David Cock, Roni Haecki, Nora Hossle, Lukas Humbel, Timothy Roscoe, and Daniel Schwyn. mmapx: Uniform memory protection in a heterogeneous world. In *Proceedings of the 18th Workshop on Hot Topics in Operating Systems*, page 159–166. ACM, June 2021b. doi:10.1145/3458336.3465273.

Gustavo Alonso, Timothy Roscoe, David Cock, Mohsen Ewaida, Kaan Kara, Dario Korolija, David Sidler, and Zeke Wang. Tackling hardware/software co-design from a database perspective. In *Proceedings of the 10th Conference on Innovative Data Systems Research*, January 2020. doi:10.3929/ethz-b-000456368.

David Cock. Bitfields and tagged unions in C: Verification through automatic generation. In Bernhard Beckert and Gerwin Klein, editors, *Proceedings of the 5th International Verification Workshop*, volume 372 of *CEUR Workshop Proceedings*, pages 44–55, August 2008. URL `http://ceur-ws.org/Vol-372/paper06.pdf`.

David Cock. Lyrebird – assigning meanings to machines. In Gerwin Klein, Ralf Huuck, and Bastian Schlich, editors, *Proceedings of the 5th International Conference on Systems Software Verification*, pages 1–9. USENIX, October 2010. URL `https://www.usenix.org/legacy/events/ssv10/tech/full_papers/Cock.pdf`.

David Cock. Exploitation as an inference problem. In *Proceedings of the 4th ACM Workshop on Artificial Intelligence and Security*, pages 105–106. ACM, October 2011. doi:10.1145/2046684.2046702.

David Cock. Verifying probabilistic correctness in Isabelle with pGCL. In *Proceedings of the 7th International Conference on Systems Software Verification*, volume 102 of *Electronic Proceedings*

in Theoretical Computer Science, pages 167–176. Open Publishing Association, November 2012. doi:10.4204/EPTCS.102.15.

David Cock. Practical probability: Applying pGCL to lattice scheduling. In *Proceedings of the 4th International Conference on Interactive Theorem Proving*, volume 7998 of *Lecture Notes in Computer Science*, pages 311–327, Rennes, France, July 2013. Springer. doi:10.1007/978-3-642-39634-2_23.

David Cock. *Leakage in Trustworthy Systems*. PhD thesis, UNSW Computer Science and Engineering, Sydney, Australia, August 2014a. `https://doi.org/10.26190/unsworks/16942`.

David Cock. From probabilistic operational semantics to information theory - side channels with pGCL in Isabelle. In *Proceedings of the 5th International Conference on Interactive Theorem Proving*, volume 8558 of *Lecture Notes in Computer Science*, pages 177–192, Vienna, Austria, July 2014b. Springer. doi:10.1007/978-3-319-08970-6_12.

David Cock. pGCL for Isabelle. *Archive of Formal Proofs*, July 2014c. `http://isa-afp.org/entries/pGCL.shtml`, Formal proof development.

David Cock, Gerwin Klein, and Thomas Sewell. Secure microkernels, state monads and scalable refinement. In Otmane Ait Mohamed, César Muñoz, and Sofiène Tahar, editors, *Proceedings of the 21st International Conference on Theorem Proving in Higher Order Logics*, volume 5170 of *Lecture Notes in Computer Science*, pages 167–182. Springer, August 2008. doi:10.1007/978-3-540-71067-7_16.

David Cock, Qian Ge, Toby Murray, and Gernot Heiser. The last mile: An empirical study of timing channels on seL4. In *Proceedings of the 21st ACM SIGSAC Conference on Computer and Communications Security*, pages 570–581. ACM, November 2014. doi:10.1145/2660267.2660294.

David Cock, Abishek Ramdas, Daniel Schwyn, Michael Giardino, Adam Turowski, Zhenhao He, Nora Hossle, Dario Korolija, Melissa Licciardello, Kristina Martsenko, Reto Achermann, Gustavo Alonso, and Timothy Roscoe. Enzian: An open, general, CPU/FPGA platform for systems software research. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, page 434–451. ACM, 2022. doi:10.1145/3503222.3507742.

Philip Derrin, Kevin Elphinstone, Gerwin Klein, David Cock, and Manuel M. T. Chakravarty. Running the manual: An approach to high-assurance microkernel development. In *Proceedings of the ACM SIGPLAN Haskell Workshop*, September 2006. doi:10.1145/1159842.1159850.

Qian Ge, Yuval Yarom, David Cock, and Gernot Heiser. A survey of microarchitectural timing attacks and countermeasures on contemporary hardware. *Journal of Cryptographic Engineering*, 8(1):1–27, 2018. doi:10.1007/s13389-016-0141-6.

Roni Haecki, Lukas Humbel, Reto Achermann, David Cock, Daniel Schwyn, and Timothy Roscoe. CleanQ: a lightweight, uniform, formally specified interface for intra-machine data transfer. 2019. doi:10.48550/ARXIV.1911.08773. eprint.

Lukas Humbel, Daniel Schwyn, Nora Hossle, Roni Haecki, Melissa Licciardello, Jan Schaer, David Cock, Michael Giardino, and Timothy Roscoe. A model-checked I2C specification. In Alfons Laarman and Ana Sokolova, editors, *Proceedings of the 27th International SPIN Symposium on Model Checking of Software*, pages 177–193. Springer, 2021. doi:10.1007/978-3-030-84629-9_10.

Gerwin Klein, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, and Simon Winwood. seL4: Formal verification of an OS kernel. In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles*, pages 207–220. ACM, October 2009. doi:10.1145/1629575.1629596.

Gerwin Klein, June Andronick, Kevin Elphinstone, Gernot Heiser, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, and Simon Winwood. seL4: Formal verification of an operating system kernel. *Communications of the ACM*, 53(6):107–115, June 2010. doi:10.1145/1743546.1743574.

Jasmin Schult, Daniel Schwyn, Michael Giardino, David Cock, Reto Achermann, and Timothy Roscoe. Declarative power sequencing. *ACM Transactions on Embedded Computing Systems*, 20(5s), September 2021. doi:10.1145/3477039.

Simon Winwood, Gerwin Klein, Thomas Sewell, June Andronick, David Cock, and Michael Norrish. Mind the gap: A verification framework for low-level C. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics*, volume 5674 of *Lecture Notes in Computer Science*, pages 500–515, Munich, Germany, August 2009. Springer. doi:10.1007/978-3-642-03359-9_34.