Master level Internship: Formalizing Hardware Security Mechanisms: interrupts, MMU and SMT proofs

Pierre Wilke
pierre.wilke@centralesupelec.fr
CIDRE-SUSHI team

Context The security of applications eventually depends on the security of all the abstractions layers they are built upon. The application itself may contain some hardening techniques, it may benefit from isolation provided by the operating system (OS), and it may also be secured through the use of hardware security mechanisms. For critical applications, formal methods (e.g. static analysis, model checking, formal proof) have been successfully applied at the source code level (e.g. Astrée, VST), at the compiler level (e.g. CompCert [6]), at the OS level (seL4 [5], CertiKOS [4]), and more recently at the hardware level (e.g. Kami [3], Kōika [2]).

Several of these approaches are based on the Coq proof assistant, which allows for formal definition of semantics and proofs. The Coq proof assistant is a program that helps building proofs and rigorously checks that the proofs are correct. The Kōika language is a high-level hardware design language (HDL) embedded in Coq, together with a verified compiler to Verilog. Thanks to this approach, one can design a circuit in the high-level HDL (Kōika) and generate from that a description in a low-level HDL (Verilog) that standard hardware tools can use to synthesise the circuit on FPGA. The authors proved that the compiler preserves the semantics of Kōika.

In the SUSHI group, we aim at proving security properties guaranteed by hardware/software mechanisms. We proposed an approach to prove security properties about Kōika designs. This requires to compile Kōika designs to an intermediate representation more suitable than the Kōika representation to formal verification. We proved that this compilation is correct and successfully apply this methodology to prove the security of a shadow stack\(^2\) in a RISC-V processor [1]. This forms a foundation for proving more complex security mechanisms.

Our current approach consists in automatically compiling Kōika designs to a more explicit representation, and then manually proving the properties of interest on this representation. The manual proof effort for this second step is still very high, and specific to each property and security mechanism. In order to automate this last step, we have started to leverage SMT (Satisfiability Modulo Theory) solvers. This approach looks very promising: we have managed to prove automatically the security properties about our shadow stack described in [1].

Internship The goal of the internship is to design and prove other security mechanisms in the RISC-V processor written in the Kōika language. This will also be the opportunity to contribute to better integrate the SMT solver within the proof workflow in Coq.

Here are the main steps of the internship:

- get familiar with Coq and the Kōika language;
- get familiar with the RISC-V processor written in that language\(^3\) and the simulation and synthesis toolchains;
- extend the processor with new mechanisms (e.g. forward-edge Control-Flow Integrity or memory isolation);
- state the security properties that these mechanisms should enforce;
- prove these properties;
- integrate with the SMTCoq \(^4\) plugin.

Required skills or interests The candidate should have familiarity with at least one of the following:

- hardware design languages (e.g. Verilog/VHDL) and computer architecture;
- or formal methods (e.g. Coq, SMT solvers).

---

\(^1\)https://coq.inria.fr
\(^2\)https://en.wikipedia.org/wiki/Shadow_stack
\(^3\)https://gitlab.inria.fr/cidre-public/koika-llr/-/blob/main/examples/rv/RVCore.v
\(^4\)https://smtcoq.github.io/
Institute  The internship will take place at CentraleSupélec in Rennes, France, in the soon-to-be-created SUSHI Inria team. This team is part of the IRISA laboratory. The internship will be advised by Pierre Wilke and Guillaume Hiet.

Practical aspects  The internship will last 5 months, starting from February, 2024. The intern will receive a "gratification" of 609€ per month. Housing options may be available on campus, or close to the campus.

This internship could be followed by a Ph.D. thesis on similar subjects.

References


---

5The team is currently CIDRE: https://team.inria.fr/cidre/
6https://www.irisa.fr/en