(Project-Funded) Ph.D Proposal (2023)
Secure compiler back-end/hardware codesign

Laure Gonnord – Grenoble-INP/LCIS & LIP

February 16, 2023

1 Context

The increasing complexity of processors and the applications they run means that the software fault models (such as instruction skips, or instruction replacement) usually used to analyze the vulnerability of their code are no longer sufficient to express the diversity of faulty behaviors in modern architectures. Indeed, architecture designers have progressively added many complex hardware blocks (such as pipeline, cache memory, branch prediction, or speculative execution) to processors in order to optimize program execution.

At the same time the development of highly optimising compilers has tried to follow the trend and provide clever analyses and optimisations, but these optimisations have the drawback of making the behavior of compilers barely previsible; and worse, these optimisations might destroy some security properties inserted at source level. The consequence is that compilers optimisations are disabled in most of security toolchains. Recent work [4] has paved the way to using compilers optimisation passes while still guaranteing security properties, but still there remains works to do as far as **specialized backends** are still to be designed.

At the LCIS Laboratory, and more generally in the context of the local (Grenoble-INP) security project (CLAM), a sequence of works has addressed the problematic of designing software and hardware that should be resistant to faulty behaviors and also vulnerability attacks:

- Previous and ongoing PhD thesis had the objective of designing fault models that should be at the same time accurate and usable to formalize realistic attacks on hardware [2].
- Some countermeasures have been proposed, at least manually, to handle these vulnerabilies at the assembly level.
- An ongoing PhD has the objective to obtain realistic and fast code vulnerability analysis on a more higher level code (compiler Intermediate representation), that would be use to *a posteriori* give trust to existing code.
- Another ongoing PhD student, Sebastien Michelland currently works on the design of compiler-based software countermeasures to hardware vulnerabilities, with a particular accent on semantic expressivity.

This PhD project has the objective to capitalize on this existing work and explore more explicitly *back-end* specific optimisations for RISCV architecture modifications targeting *hardware vulnerabilities*. The PhD candidate will design **code analyses and optimisations specially tailored to hardware modifications**, especially targeting security. Finally, it will also explore the less explored feed-back of hardware design on compilers.

As an illustration of this, the litterature has already proposed such codesign:

- A constant time code is more likely to be resistant to timing attacks. This property could be ensured in the source code, but it might be destroyed by further code optimisations in compilers. Another option to ensure such property could be to add constant time operations inside the target machine [1] (like ternary ifs for instance) and ensures at compilation time that potential leaking code uses these new instructions (*Instruction selection*)
- Similarly, low level machine securisation might take the form of more secure registers, the problem is then to modify the backend of the compiler so that to use these secure registers at crucial security places of the code. (*Register allocation*).

2 Objective and Program of the PhD

The context of the PhD is the French national effort in cybersecurity[1], and especially embedded system security, within the ARSENE project, which was accepted Fall 2022.

The goal of this PhD is to investigate the cross fertilization between back-end optimization and the back-end design itself, namely RISCV architectures (which are at the core of the ARSENE project, that assess the choice of this free architecture for european sovereignty), to enhance the capabilities of both software and hardware security mechanisms.

In fact, existing work either use few compilation techniques and try to avoid code modification as much as possible; or are only used by hand. Very few existing works address at the same time the design of the compiler backend in coordination with the securization opportunities proposed by hardware architects [3].

We propose to make a step further in looking specific backend passes, especially instruction selection and register allocation, and design a codesign methodology targetting RISCV security.

Program

- In the first year, the Phd student will extensively study the above cited sources of inspiration under the point of view of low-level security. He/She will propose a first case study of backend optimisation coming from security information, in particular from the RISCV modifications proposed in the team for instance in [3] and the ARSENE Lot 1 outputs. Existing research tools for software vulnerabilies evaluation will be used to evaluate the pertinence of backend securisation.
- The second year will be devoted to designing a first instance of a software and hardware codesign for security purposes, and validate the framework on a case study. This case study might probably come from the use-cases provided by other ARSENE project members.
- For the rest of the Phd, and depending of the current results/ status of the previous activities and the other WP of the ARSENE project, the student might study:
    - A way to be less RISCV-specific (for instance, provide a language based solution to back-end modifications);
    - The link with the secure operating system bricks that are also part of “Lot 2”. In particular, evaluating the end-to-end execution of a secure program on RISCV IoT is one of the challenges of the ARSENE project.

[1] https://www.gouvernement.fr/recherche-programmes-et-equipements-prioritaires-de-recherche-3-milliards-d-euros-mobilises-pour-la

3 Location and Supervising

The funding of the Phd as well as its environment (travels, conferences, equipment) is guaranteed by the ARSENE Project. The salary before taxes is around 2k€ per month. The candidate will also have the possibility to teach up to an amount of 64 hours per year (additional salary of $\sim 200e$ per month).

The PhD will take place in the LCIS lab, Valence, France in the CTSYS team[2]. The lab and the team take a strong place in the Cybersecurity european and french communities: they are involved the national PEPR project ARSENE, and many european and/or industrial projects around security.

The laboratory and the Ésisar engineer school also share a cybersecurity platform for graduate teaching, that contains hardware to physically realize hardware attacks, and also a private network for admin security teaching. The school and UGA/Grenoble INP also recently initiated an ambitious project called Cyberskills (french national "Metiers d'avenir" project; 4.3M€ for the Grenoble/Valence consortium) that will capitalize on the research outputs of ARSENE in order to provide updated course offers in embedded systems security. As an example, a new apprenticeship engineer curriculum opens in september 2023[3].

The thesis will be supervised by:

- **Directrice** Laure Gonnord, **habilitated** Professor at Grenoble INP, Esisar (50% on this PhD). Currently co-advising two other PhDs students: one in 3rd year in sept 2023, with Gabriel Radanne, at LIP; and antoher one on semantic approaches for compilation, 2nd year in sept 2023, coadvised with Christophe Deleuze at LCIS. Laure Gonnord has many experiences in static analysis applied to compilation in general, and its application to HPC and embedded systems. She has designed analyses that especially target the LLVM compiler's intermediate representation and low-level memory information.
- **Coencadrant** D. Hely or C. Deleuze, Associate Professors at Grenoble INP, Esisar. They both have experience in in the development of hardware and software fault models.

References

[1] José Bacelar Almeida, Manuel Barbosa, Gilles Barthe, Vincent Laporte, and Tiago Oliveira. Certified compilation for cryptography: Extended x86 instructions and constant-time verification. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology – INDOCRYPT 2020*, pages 107–127, Cham, 2020. Springer International Publishing.

[2] Louis Dureuil, Guillaume Petiot, Marie-Laure Potet, Thanh-Ha Le, Aude Crohen, and Philippe de Choudens. Fissc: A fault injection and simulation secure collection. In Amund Skavhaug, Jérémie Guiochet, and Friedemann Bitsch, editors, *Computer Safety, Reliability, and Security*, pages 3–11, Cham, 2016. Springer International Publishing.

[3] Johan Laurent, Vincent Beroulle, Christophe Deleuze, Florian Pebay-Peyroula, and Athanasios Papadimitriou. Cross-layer analysis of software fault models and countermeasures against hardware fault attacks in a RISC-V processor. *Microprocessors and Microsystems: Embedded Hardware Design (MICPRO)*, 71:102862, November 2019. Conference: ASHA Convention. Boston, MA. 2018.

[4] Son Tuan Vu, Karine Heydemann, Arnaud de Grandmaison, and Albert Cohen. Secure delivery of program properties through optimizing compilation. In *Proceedings of the 29th International Conference on Compiler Construction*, CC 2020, pages 14–26, New York, NY, USA, 2020. Association for Computing Machinery.

[2] https://lcis.grenoble-inp.fr/themes/securite-des-systemes-embarques-et-distribues-critiques

[3] https://esisar.grenoble-inp.fr/fr/formation/apprentissage