Countermeasures to (transient) Side-Channel Attacks in a Formally Verified Compiler

VERIMAG Laboratory (Grenoble – France)
Teams: PACS (Proofs and Code Analysis for Security) and Formal Proofs
Advisors: David Monniaux & Bruno Ferres
david.monniaux@univ-grenoble-alpes.fr
bruno.ferres@univ-grenoble-alpes.fr

Keywords secured compilation; `CompCert`; hardware security; micro-architecture

Context

Timing side-channel attacks have long been known: the actions of a program have measurable consequences on the timing of another program running on the same machine, through caches, branch predictors, and other shared resources. Hardware and software countermeasures have been proposed against such attacks. Software countermeasures include, for instance, never accessing memory through an address computed from secret data, and thus implementing look-up tables in cryptographic primitives without using memory arrays.

High-performance processors exploit speculative execution: the processor begins executing instructions without knowing whether they will be accessible according to branches, and retracts their architectural effects (their effects on registers, memory etc. as visible from the program) if they are not. The processor however does not retract the microarchitectural effects of these *transient executions*, such as loading values into caches. The first published transient side-channel attacks were *Spectre* and *Meltdown*, in January 2018, and since then a number of other attacks exploiting side channels and speculation have been published. For instance, it became known in 2022 that the Apple M1 and M2 processors speculatively prefetch data at addresses that look like pointers, and though this was initially considered difficult to exploit, an attack against cryptographic primitives was published in March 2024.[1]

Countermeasures against side-channel and transient attacks may be implemented in hardware, software, or a mixture thereof. For instance, a processor may have "barrier" instructions that block speculative executions. For processors that do not have such instructions, it may still be possible to block speculation using specific programming patterns. Special instructions or programming patterns may be used manually by the programmer, or may be inserted at compile-time.

Objectives

In the context of the national (ANR) project ARSENE[2], we aim at providing a security-enhanced, verified compiler. In particular, we are working with the CompCert compiler[3], the first formally verified C-compiler. In this project, we are investigating the possibility of adding low-level security features (against the attacks presented above) directly in the compilation flow.

[1] *GoFetch*, `https://gofetch.fail/files/gofetch.pdf`

[2] `https://www.pepr-cyber-arsene.fr/`

[3] `https://compcert.org/`

It is then expected that the researcher will

1. investigate software or hardware/software countermeasures currently implemented (e.g. in gcc or clang).
2. get in touch with proposals of hardware support for countermeasures in the RISC-V platform.
3. design software countermeasures (possibly based on specific hardware support) and implement them in the CompCert formally verified compiler.[4]
4. prove that these countermeasures do not thwart normal executions of programs (*soudness*).
5. hopefully, prove that these countermeasures have an effect against a certain model of attacks (*adequacy*).

Applicant Profile

The ANR funding can be used either for a PhD or for a postdoc, depending on the profile of the chosen applicant.

In either way, the applicant should have knowledge and/or interest for the following topics:

- software/hardware security
- formal methods, especially in the context of compilation and verification of programs
- hardware architectures of processors

To apply for a PhD funding, the applicant should have a M2/Engineering degree in Computer Sciences, or a related topic.

To apply for a postdoc funding, the applicant should have a PhD degree in those same domains.

Contact

If you are interested in this job proposal, do not hesitate to contact the advisors for further information. To apply, please join to your e-mail a curriculum (and a grade transcript for a PhD funding), as well as a motivation letter and any document that may support your application.

[4] More specifically, in the *Chamois* branch
`https://gricad-gitlab.univ-grenoble-alpes.fr/certicompil/Chamois-CompCert`