Report on the outcomes of a Short-Term Scientific Mission[1]

Action number: CA20111

Grantee name: Volker STOLZ

Details of the STSM

Title: **Coupling SMT and theorem proving to prove hardware/software security mechanisms**

Start and end date: 01/09/2024 to 11/09/2024

Description of the work carried out during the STSM

Description of the activities carried out during the STSM. Any deviations from the initial working plan shall also be described in this section.

(max. 500 words)

At CentraleSupélec, we have established a lab setup with the necessary components to be able to run example programs and obtain program traces on ZCU4 boards. The infrastructure is accessible to lab-researchers and remotely. Our work is based on the CSAL-library, and the code that we have been using to obtain traces is available in our public fork at https://github.com/AuroraRennes/CSAL/tree/vs-aurora. As it turns out, traces obtained in this way are wrapped into another protocol (D4.2 “Frame descriptions” in [1]), for which a separate decoder pass is required (“deformatting”). As an alternative, we have investigated the use of the `perf`-tool, which produces directly readable traces through the OpenCSD-library. This can only be an interim solution for prototyping, as we are not interested in traces recorded into files, but eventually passed from the processor into a monitor on the FPGA that we plan to specify in the Kôika Hardware Description Language [2,3].

Stolz then had sessions with Baty, Hiet and Wilke gaining insights into the use of Kôika for hardware-specification, so that future work can be split between the Norwegian and the French partner. Furthermore, we developed an outline on specifying the trace source as well as part of the processor-design in the same spirit as [3].

[1] “ARM® CoreSightTM Architecture Specification v3.0”, ARM, 2017.

[1] This report is submitted by the grantee to the Action MC for approval and for claiming payment of the awarded grant. The Grant Awarding Coordinator coordinates the evaluation of this report on behalf of the Action MC and instructs the GH for payment of the Grant.

[2] Bourgeat, T., Pit-Claudel, C., Chlipala, A., Arvind: The Essence of Bluespec: a Core Language for Rule-Based Hardware Design. PLDI 2020. ACM (2020).

[3] “A generic framework to develop and verify security mechanisms at the microarchitectural level: application to control-flow integrity”, Matthieu Baty, Pierre Wilke, Guillaume Hiet, Arnaud Fontaine, Alix Trieu, 36th IEEE Computer Security Foundations Symposium, https://inria.hal.science/hal-04118645/. Also see https://gitlab.inria.fr/SUSHI-public/FMH/herve

Description of the STSM main achievements and planned follow-up activities

Description and assessment of whether the STSM achieved its planned goals and expected outcomes, including specific contribution to Action objective and deliverables, or publications resulting from the STSM. Agreed plans for future follow-up collaborations shall also be described in this section.

(max. 500 words)

The meeting has allowed us to transfer knowledge between the partners on the practical and theoretical level (O-C1: Bring together members of the different communities working on proofs in Europe).

On the practical side, we have developed a deployment (networking, build-host, software installation) for the ZCU4-boards that both partners received through AMD’s University/Donation Program that the Norwegian partner can now replicate at their site. This will allow us to conduct practical experiments on either side of the collaboration. On the theoretical side, we have now a common understanding of the necessary steps to model behaviour on the lowest level of processors in a high-level language that will eventually also support verification.

We have planned the further collaborative tasks: student-projects will be published at both partners for the upcoming semester to develop an ARM Trace decoder in a language amenable to formal verification. Depending on the level and qualification of the prospective students, this would either be formal specification via Coq with extraction to OCaml, or only a direct implementation in OCaml that could be subjected to verification afterwards. The resulting specification, decoder and proofs will be made publicly available and possibly advancing development of libraries and proof techniques, contributing to O-R3: Make techniques for program verification more effective and more accessible to all stakeholders.

For the specification of the generating-side of hardware, we have planned the development of the necessary processor-component in Kôika based on Baty’s work for a new PhD student that will start in Rennes last quarter 2024. He will be supported by a new post-doc in the research group starting 1.10. We plan implementing one of the several possible trace buffer modes (e.g. software FIFO, circular buffer) for the RV32I, and specifying a consumer to test our work. The consumer will be designed as *another* hardware component (again, most likely through Kôika). The consumer can be parametrized either (for prototyping) by synthesizing an over-abstraction of the control-flow graph as a security monitor directly onto the hardware, or (later) by reading the graph from memory. These outputs will contribute to O-C2: Act as a stakeholder platform in the field of formal proofs from its theoretical grounds to its industrial applications, O-C8: Disseminate the results of the Action activities to the scientific community, the industry, the certification bodies, the European institutions and to the general public.

Future Outcomes:

- Repository with specification and examples of trace decoder (2025)
- Publication on high-level specification of trace mechanism incl. artefact (pre-print 2025, possible targets SILM Workshop, IEEE Computer Security Foundations Symposium, DATE: Design, Automation and Test in Europe Conference)
- Publication on formally verified trace consumer (of either software- or hardware-traces) incl. artefact (ditto)

Rennes, 23.9.2024

Volker STOLZ **Guillaume HIET**