Securing a RISC-V architecture: A dynamic approach

Sébastien Pillement, Maria Mendez Real, Juliette Pottier, Thomas Nieddu, Bertrand Le Gal, Sébastien Faucou, Jean-Luc Béchennec, Mikaël Briday, Sylvain Girbal, Jimmy Le Rhun, et al.

Securing a RISC-V architecture: A dynamic approach

S. Pillement[1], M. Méndez Real[1], J. Pottier[1], T. Nieddu[2], B. Le Gal[2], S. Faucou[3], J.L. Béchennec[3], M. Briday[3], S. Girbal[4], J. Le Rhun[4], O. Gilles[4], D. Gracia Pérez[4], A. Sintzoff[5] and J.R. Coulon[5]

[1]Nantes Université, CNRS, IETR UMR 6164, F-44000 Nantes, France
[2]Univ. Bordeaux, Bordeaux INP, CNRS, IMS, UMR 5218, F-33400 Talence, France
[3]Nantes Université, CNRS, LS2N UMR 6004, F-44000 Nantes, France
[4]Thales Research & Technology, F-91767 Palaiseau, France
[5]Thales DIS, F-13590 Meyreuil, France

***Abstract*—The SecureV (also known as SecV) project offers an innovative, open-source hardware, secure, and high-performance processor core based on the RISC-V ISA. The originality of the approach lies in the integration of a complete solution to increase security based on dynamic code transformation, covering 4 of the 5 NIST[1] functions of cybersecurity via monitoring (identify, detect), obfuscation (protect), and dynamic adaptation (react).**

***Index Terms*—Cybersecurity, RISC-V, open-source hardware**

I. Introduction

With the digitization of the society, security has become one of the most important properties of connected devices. The SecV project proposes an innovative and adaptable approach to increase security of processors. The resulting processor will be a key technology for building secure connected devices and services.

This project is funded by the French ministry of research and has started in 2022 for a 46-month duration (initial stage project). The SecV consortium consists of five partners. Two complementary divisions of the Thales group: Thales Research & Technology (TRT) in charge of upstream R&D, and Thales DIS (formerly INVIA) specialized in the development of secure processors. Three university laboratories (IETR, IMS, LS2N) provide expertise in dynamic system design, runtime verification, and security. The project is led by the IETR lab.

II. Objectives and scientific hypotheses

For decades, processor designers have focused on improving the performance (in the broadest sense) of systems. However, in recent years, it has been shown that attacks directly on the hardware allow stealing information by exploiting vulnerabilities introduced as side effects of performance optimization mechanisms [1], [2]. Thus, proven attacks have highlighted flaws due to the use of cache memories in high performance processors [3], [4], the presence of Hardware Performance Counters (HPCs) [5], or the functions provided by power consumption managers (DVFS) [6], [7]. More recently, Sepúlveda *et al.* [8] showed a combined micro-architectural attack using bus infrastructure information to trigger a cache-based attack. The latest emblematic attacks, Spectre [9] and Meltdown [10], exploit for example the interplay between branch prediction and caches of modern processors. All these attacks show that security must become one of the main concerns in the design of micro-architectures, just like power consumption or performance.

For performance reasons, most of the processor manufacturers integrate into their architectures a hardwired instruction decoder optimized for the most commonly used instructions. However, erratic operating behaviors can potentially remain in any processor. In x86-type processors, a so-called micro-code translation unit translates the CISC instruction set exposed to the user to an internal optimized instruction set (e.g., Intel Micro-Ops [11]). This translation unit, which can be updated, generates pre-coded control words, generally called micro-instructions. The update of this micro-code translator is generally static and is traditionally used for adding features and fixing post-manufacturing silicon bugs. It has been used, for example, to thwart recent vulnerabilities such as Spectre and Meltdown.

The concept of soft-core has grown in recent years, for example on the model of ARM processors, to lead today to the open-source hardware approach marked in particular by the emergence of the commercial-grade RISC-V[2] Instruction Set Architecture (ISA). In this approach, the processor is proposed in the form of a block described in a hardware description language which can be modified according to the requirements of the system. For example, the processor instruction set could be adapted [12] or blocks dedicated to security could integrated into the micro-architecture [13]. A last approach is to add dedicated hardware, as co-processors, to increase the security level of the SoC at the system level [14]. Currently available open-source hardware processor models mainly integrate static instruction decoders, that cannot be modified in normal operation condition (it can be modified by re-synthesis of the whole core). In addition, recent work [15]

contact author: sebastien.pillement@univ-nantes.fr
[1]National Institute of Standards and Technology

[2]https://riscv.org

has demonstrated the interest of dynamic management of microcode translation as a counter-measure to attacks on the cache memory for CISC processors. Conversely, using a microcode decoder can also open the door to specific classes of attacks [16], arguing for a careful design and protection of this unit.

The main objective of the SecV project is to propose an open-source hardware, secure, and high-performance processor core based on the RISC-V ISA. The originality of the approach lies in the integration of a dynamic code transformation unit, adaptive memory management policies, and a dynamic control unit based on runtime verification, opening the way to numerous online adaptations aiming at supporting the concept of cyber resilience [17]. This adaptation capacity will improve the security without having to re-design the chip.

The main outcomes of the project are a prototype of a complete RISC-V ISA based architecture running on top of an FPGA, including the security-oriented blocks designed in the project. An analysis of the security level reached by the propositions is also expected. As we work at the micro-architecture level, the SecV architecture will be at the center of the security kernel of future products.

III. Position of the project as it relates to the state of the art

The security strategy implemented in SecV is strongly based on the obfuscation of the micro-architecture behavior, and aims primarily at countering side channel attacks (at least cache and synchronization attacks). This obfuscation relies on the dynamic adaptation capabilities introduced at the instruction decoder and memory hierarchy levels, driven by the control unit. Obfuscation will be triggered by the execution of *sensitive* code blocks within an application, but also by the control unit when it will detect the violation of a security property of the micro-architecture, denoting for instance a potential attack aimed at taking over the control flow of the programs (e.g. code injection or return-oriented-programming attacks). As the blocks designed in the project are re-programmable, the architecture is intended to be adaptable to other types of threats. For instance, we strongly believe that dynamic power analysis attacks could be handled by the proposed obfuscation approach. In this first step, we consider that physical attacks are out of the scope of the project.

Open-source hardware has attracted a lot of works these past years, due to its numerous advantages. For both industry and academia, the RISC-V initiative brings technological independence from manufacturers, thus offering some form of sovereignty. As the description is provided using a hardware description language, teamwork is enabled and supports the design of dedicated or specialized approaches. Finally the large communities gathered around the flagship projects of open-source hardware have proposed complete frameworks with open and standard APIs, easing the deployment of processors, especially for security (like Securyzr from Secure-IC[3]). An illustration of the benefits of open-source hardware is provided by the PSPC Secure-RISCV[4] project where a dedicated TRNG hardware block is integrated in an IoT oriented RISC-V core from GreenWaves Technologies. The above projects mainly rely on adding new functionalities (in hardware and/or software) to an open-source hardware core. Moreover, they target known attacks by adding efficient static counter-measures. In contrast, the SecV project intervenes at the micro-architecture level, and proposes to rely on dynamic adaptation to increase security through obfuscation. The proposed approach is also extensible, so that it can be configured to deal with current and future threats.

The COFFI project[5], intends to protect the processor against physical attacks and specifically attacks on the control flow. While the approach is not specifically dedicated to RISC-V architecture, this project proposes an holistic approach (from source code, up to the micro-instructions) to implement powerful control flow and execution integrity (CFEI) counter-measures. In the SecV project we do not target specifically physical attacks in a first step, and we do not rely on the knowledge of the source code of the application. However, our architecture could help in implementing the CFEI strategy efficiently.

Lots of projects focus on the design of "IoT-oriented" processor cores (as the projects described above). In SecV we would like to target a high-performance core and take into account the impact of advanced performance oriented features on security.

This is also the purpose of the ARCHI-SEC[6] project started in 2019. In this project the partners study complex paths to attack modern processors including complex design features (*e.g.*, branch prediction, Out-of-Order execution, cache coherency protocols), and new technologies like non volatile RAM. The main goal of ARCHI-SEC is to model and simulate (using Gem5 simulator) a complex architecture to identify its weaknesses. It is obvious that the SecV project could use the outcomes of the ARCHI-SEC project as inputs for some of its own tasks.

At the European level, we can cite the De-RISC[7] FTI project that aims at designing a European dependable processor based on the RISC-V ISA for space and aeronautics. Even if dependability includes some security issues, the main objective of the project is to design a reliable processor core, having thus different objectives from SecV.

The SECURE-V project aims to make specific key contributions to improve the global security of the final architecture. The proposed innovative solution is built around three main contributions:

1) The dynamic code transformation unit is one of the main ideas of this proposal. Based on the microcoded

[3]https://www.secure-ic.com/solutions/securyzr/

[4]https://laboratoirehubertcurien.univ-st-etienne.fr/en/teams/secure-embedded-systems-hardware-architectures/projects-and-partners-2.html

[5]https://www.mines-stetienne.fr/coffi-en/

[6]https://archi-sec.telecom-paristech.fr/Presentation

[7]https://derisc-project.eu/

approach of CISC processors, this unit will support on-the-fly modifications of the program instructions translation and decoding processes. These changes aim to provide security building blocks by either altering the processor pipeline datapath behavior or hiding the sensitive paths with data obfuscation. These two mechanisms are respectively possible using instructions operations rescheduling or with injections of additional operations into the pipeline. This unit will also allow the dynamic instrumentation of a code without modifying the original binary. This possibility will help to evaluate a code, to monitor suspicious behavior for systems requiring fine-grained monitoring (HUMS - Health and Usage Monitoring System) or to detect malicious intrusions (HIDS - Hybrid Intrusion Detection Systems). Even if this may require a software level intervention (out of the scope of the project), this unit will finally allow the modification of the processor instruction set to support security updates and bugs correction.

2) To support and take full advantage of the dynamic code transformation unit, some elements of the micro-architecture will need to be modified. Built in close conjunction with the dynamic code transformation unit, advanced configurable memory management policies will add an additional level of obfuscation. We can cite for example alternatives to conventional caches such as scratchpads with dynamic management, dedicated memories (or TCM for Tightly Coupled Memory), partially reconfigurable caches in SRAM. This leads to new programming paradigms or associated memory models, making it possible to free oneself from the hardware constraint of managing globally shared and coherent memory space (the underlying programming model is out of the scope of the project). New dynamic security-oriented cache management will also be investigated in order to avoid interference at this level, or to supervise on demand information when identifying suspicious behaviors. Finally, as for the operations in the pipeline, we will also be able to insert (periodically or on a more controlled manner) instructions to different memory areas aiming at different memories accesses activity in order to disrupt side-channel attacks. The other performance elements used for optimization purposes will also be considered to increase the security of the whole core.
3) In order to complete the approach and to control, in particular, the dynamic changes in the behavior of the micro-architecture, a dynamic control block will be developed. By observing the state of the micro-architecture, this unit will be able to determine when specific security properties are violated, denoting an abnormal behavior that may result from an attack, or a sensitive context that needs to be secured. In reaction to the detected situation, it will trigger an adaptation via the dynamic code transformation unit and/or the micro-architecture. The use of runtime verification, a lightweight formal method, to synthesize monitors of security properties, will help to justify the confidence placed in this unit.

IV. The CVA6 core

As previously mentioned, the SecV project will focus on the design of a commercial grade high-performance processor. The project will be built on top of the CVA6 core[8] (formerly known as Ariane), which is supported by the OpenHW Group[9] as an implementation of the RISC-V ISA. Thales is leading the CVA6 project at the OpenHW Group and is an important contributor to CVA6 core family, having provided a 32-bit version and a test bench environment for the 64-bit and 32-bit versions.

The architecture of the CVA6 core is presented in Figure 1 [18]. It is a 6-stage, single issue, in-order CPU which implements the 64-bit RISC-V instruction set. It supports a Unix-like operating system.

The front end is composed of a classical control stage using an instruction cache. The CVA6 supports speculative execution and includes a complete branch prediction unit. The decode stage supports the RISC-V compressed instructions, and manages the instructions alignment. The CPU issues the instructions in-order, but the execute stage can handle them out-of-order. The execute stage supports the classical arithmetic and logic operations, a dedicated multiplier/divider (supporting the RISC-V M extension), a load/store unit and a branch dedicated unit, linked with the branch prediction of the front-end stage. All the operations are carried-out through a data cache. The last commit stage re-orders the instructions, and updates the architecture registers while managing the interruptions.

The core is configurable and can be adapted in-size and in operation to support various requirements.

V. Methodology and organization

a) Project Organization: The SecV project is composed of 4 scientific Work Packages (WP), and one dedicated to the general management of the project. One key result of this last WP will be the establishment of a close collaboration between all partners and a good dissemination of the project results. The scientific WPs coverage of the SecV project is detailed in Figure 1 on top of the CVA6 core which is the intended RISC-V compliant architecture.

A. WP 1 – Dynamic code transformation unit:

This work package aims at designing the unit capable of performing transformation steps on a code by dynamically translating and decoding standard or custom RISC-V instructions at run-time. The proposed architecture will rely on a set of context-sensitive decoders to handle high-performance or critical instructions, and macro-instructions. This unit will implement various security primitives to tackle most common security flaws. Its dynamic nature will be used for enabling

[8]https://github.com/openhwgroup/cva6

[9]https://openhwgroup.org consortium of organizations of which Thales is an active member.

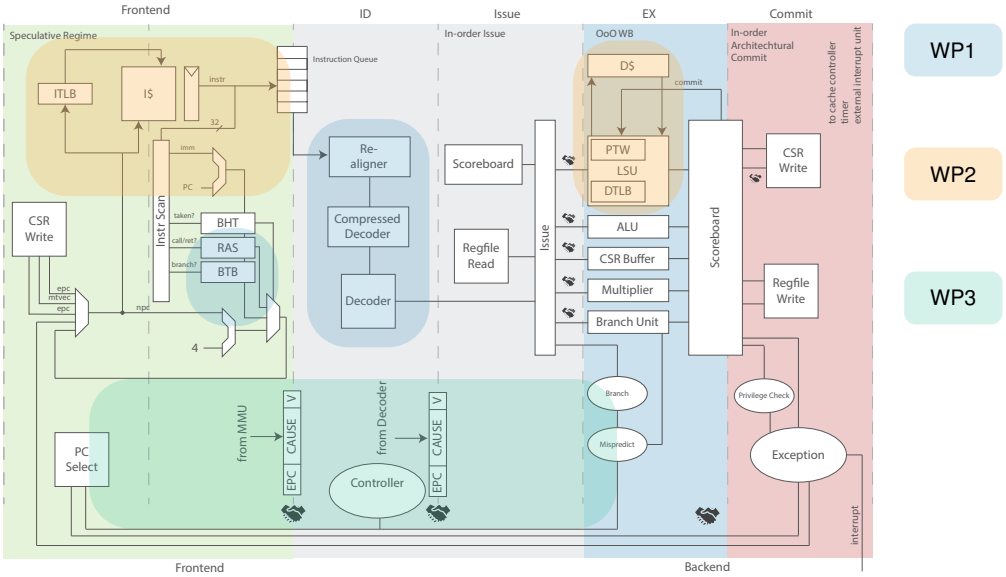

Fig. 1. Structure of the CVA6 processor core and focus of the different SecV WP's. The WP 4 concerns the hardware implementation of the other WPs' propositions in the CVA6 core project. Original picture taken from [18].

context-sensitive application monitoring at hardware level. All various micro-architectural design decisions taken will be evaluated from security requirements perspective. To build a complete and secure architecture and promote this work, this fundamental building block will be implemented and fully integrated in the CVA6 processor pipeline on a reprogrammable hardware target as part of the WP 4.

Mainly involving the IMS lab, this WP builds on the skills of Thales DIS in terms of the design of secured processors and the IETR on the design of dynamic architectures. The dynamic code instrumentation part will be supported by TRT.

B. WP 2 – μ-architectural modifications

The objective of this WP is to propose micro-architectural modifications that will benefit from the code transformation unit developed in WP 1. The context-sensitive approach of the code translation will trigger specific hardware behaviors enhancing the security aspects of the micro-architecture while it is executing specific security-related functions; or to facilitate the integration of alternative domain specific hardware components such as scratchpads or tightly coupled memories. Also supported by the dynamic code transformation block, this WP will explore alternative cache memories as well as adaptable security-oriented cache management strategies according to the current execution context and suitable security level.

This WP mainly involves TRT and IETR on the memory hierarchy and cache management part.

C. WP 3 – Dynamic control of the architecture adaptation

The objective of this WP is to provide the target micro-architecture with a programmable unit in charge of both runtime verification, to identify suspicious behavior of a applications, and implement the dynamic control of the complete architecture ensuring that the dynamic changes on the dynamic code translation unit do not jeopardize the normal functioning of the architecture.

The runtime verification part aims to detect the violation of security properties expressed on the behavior of the tasks and/or the states of the micro-architecture. The dynamic control part will trigger security management policies of the dynamic code transformation unit and/or of the micro-architecture either when an attack is suspected or at the request of the program.

Built on the skills of LS2N and IETR, this WP will require strong collaboration with all of the project partners, since it affects the overall control of the system.

D. WP 4 – Prototype and evaluation

The objective of this work package is to implement a running prototype of the contributions developed in the other work packages and to evaluate the level of security achieved thanks to the SecV propositions.

To achieve this objective, tasks will include:

- the global integration of the different hardware blocks in the commercial-grade CVA6 core,
- and an assessment of the proposed approaches to evaluate the achieved security level.

All the developments will follow the open-source hardware philosophy and are intended to be included in the OpenHW Group CVA6 project.

Led by Thales DIS and taking advantage of its expertise on the CVA6 processor core, this WP will bring together all the validation elements of the various contributions in the CVA6 project.

VI. Expected scientific & technological benefits of the project

The project is built on top of the CVA6 core as an implementation of the RISC-V ISA, and will study the improvement of the security of this processor core for industrial use. All the contributions will be validated either by simulation or in a physical prototype (FPGA). The different proposals will be evaluated through models and cycle-accurate simulation, and according to safety and security indicators and metrics that will be defined in the project. Whenever possible the solutions will be implemented and validated against reference attacks (such as [19]), on a real FPGA target, in order to validate their effectiveness and to evaluate their real cost.

Improving security of embedded systems is currently mainstream, but providing a cross-domain one-fits-all solution is extremely challenging. The SecV project proposes a dynamic approach reusable across several domains by changing the translation strategy and therefore able to propose a common cross-domain customizable hardware, adapting to each domain specific security constraints.

To improve the security of processors, the strategy of the SecV project is to adapt the behavior of the micro-architecture when a sensitive context is detected, by exploiting a dynamic code transformation unit and dynamic memory management policies. This approach is particularly innovative. The results of the project will allow us to better understand how inline adaptation of the micro-architecture can improve security, but also the cost in terms of performance of integrating such capabilities into systems. The targeted security improvement is aimed at documented software attacks at the micro-architecture level, but the ability to re-program the adaptation blocks is also intended to allow future attacks to be addressed without the need for deeper architecture changes. The long-term goal is therefore to design an architecture with security sustainable over time.

The SecV project partners will adopt an open-source strategy to distribute the project results. As previously explained in the project description, some additional blocks to improve security of the open-source CVA6 RISC-V core will be developed by all partners. If compliant with the OpenHW Group requirements, the prototype will be made available on their open-source repository[10], offering the possibility for internal and external entities to reuse this platform or its components.

The adoption of a strategy of dissemination of the main project results under open licenses and the willingness to work on their integration into an existing ecosystem will promote their reuse within research and development activities conducted by third parties (either academic or industrial). In order to facilitate the development of these activities, a careful and detailed documentation will be provided along with the progress of the project. The documentation will include tutorials, examples, and application use cases.

[10]https://github.com/openhwgroup/cva6

VII. Acknowledgment

This research is supported by the French research agency (ANR) under the contract ANR-21-CE-39-0017. The authors would like to thank Guillaume Bouffard and Philippe Trebuchet from ANSSI for their valuable help in this project.

References

[1] Qian Ge, Yuval Yarom, Frank Li, and Gernot Heiser. Your processor leaks information - and there's nothing you can do about it. *arXiv preprint arXiv:1612.04474*, 2016.

[2] Chester Rebeiro and Debdeep Mukhopadhyay. Micro-architectural analysis of time-driven cache attacks: Quest for the ideal implementation. *IEEE Transactions on Computers*, 64(3):778–790, 2013.

[3] Daniel Gruss, Clémentine Maurice, Klaus Wagner, and Stefan Mangard. Flush+flush: A fast and stealthy cache attack. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 279–299. Springer, 2016.

[4] Gorka Irazoqui, Thomas Eisenbarth, and Berk Sunar. S$a: A shared cache attack that works across cores and defies vm sandboxing – and its application to aes. In *2015 IEEE Symposium on Security and Privacy*, pages 591–604. IEEE, 2015.

[5] Sarani Bhattacharya and Debdeep Mukhopadhyay. Utilizing performance counters for compromising public key ciphers. *ACM Trans. Priv. Secur.*, 21(1), 2018.

[6] S. Noubir, M. Mendez Real, and S. Pillement. Towards malicious exploitation of energy management mechanisms. In *Design, Automation & Test in Europe Conference & Exhibition*, pages 1043–1048, 2020.

[7] Adrian Tang, Simha Sethumadhavan, and Salvatore Stolfo. Clkscrew: Exposing the perils of security-oblivious energy management. In *26th USENIX Security Symposium*, pages 1057–1074, 2017.

[8] Johanna Sepúlveda, Mathieu Gross, Andreas Zankl, and Georg Sigl. Beyond cache attacks: Exploiting the bus-based communication structure for powerful on-chip microarchitectural attacks. *ACM Transactions on Embedded Computing Systems (TECS)*, 20(2):1–23, 2021.

[9] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, et al. Spectre attacks: Exploiting speculative execution. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1–19. IEEE, 2019.

[10] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, et al. Meltdown: Reading kernel memory from user space. In *27th USENIX Security Symposium*, pages 973–990, 2018.

[11] Robert P. Colwell. The origin of intel's micro-ops. *IEEE Micro*, 41(6):37–41, 2021.

[12] H. Handschuh. Risc v and security: How, when and why. In *International Conference on Cryptographic Hardware and Embedded Systems – CHES*. Springer, 2019.

[13] Gregory T Sullivan, André DeHon, Steven Milburn, Eli Boling, Marco Ciaffi, Jothy Rosenberg, and Andrew Sutherland. The dover inherently secure processor. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–5. IEEE, 2017.

[14] Zhenya Zang, Yao Liu, and Ray CC Cheung. Reconfigurable risc-v secure processor and soc integration. In *International Conference on Industrial Technology (ICIT)*, pages 827–832. IEEE, 2019.

[15] Mohammadkazem Taram, Ashish Venkat, and Dean M Tullsen. Context-sensitive decoding: On-demand microcode customization for security and energy management. *IEEE Micro*, 39(3):75–83, 2019.

[16] Philipp Koppe, Benjamin Kollenda, Marc Fyrbiak, Christian Kison, Robert Gawlik, Christof Paar, and Thorsten Holz. Reverse engineering x86 processor microcode. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1163–1180, 2017.

[17] Alexander Kott and Igor Linkov. To improve cyber resilience, measure it. *Computer*, 54(2):80–85, 2021.

[18] F. Zaruba and L. Benini. The cost of application-class processing: Energy and performance analysis of a linux-ready 1.7-ghz 64-bit risc-v core in 22-nm fdsoi technology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(11):2629–2640, Nov 2019.

[19] John Wilander, Nick Nikiforakis, Yves Younan, Mariam Kamkar, and Wouter Joosen. Ripe: Runtime intrusion prevention evaluator. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 41–50, 2011.