Security assessment methodology for RISC-V cores

Apurba Karmarkar, Pablo Navarro-Torrero, Eros Camacho-Ruiz*
Macarena C. Martínez-Rodríguez, Piedad Brox

Instituto de Microelectrónica de Sevilla, IMSE-CNM, CSIC, US, Spain

Abstract

The rise in RISC-V processor adaptation in today's electronic landscape is also bringing more security threats to them. In this context, we have developed a methodology to evaluate the security of RISC-V cores against side-channel attacks. The methodology is conceived to be compatible with any RISC-V core and it is focused on the analysis on power traces while the processor is running a cryptographic software algorithm on it. To validate the methodology, we extract power traces of the execution of an AES on a Rocket RISC-V core.

Introduction

Nowadays, the Internet of Things (IoT) technology is widely used in our daily lives, from homes and offices to industries, where a large amount of data is shared. In this context, the security assessment of the processors embedded on IoT devices is a key aspect when facing safety concerns against possible cyberattacks [1]. RISC-V cores are being used as embedded processors for many IoT applications; therefore, their security evaluation is becoming essential, being a critical aspect as well as other key performance indicators for its selection.

Side Channel Attacks exploit the data leakage by measuring physical parameters to extract secret information during the normal operational mode of a system. Some examples of physical characteristics are execution time, power consumption, and electromagnetic emission of the targeted system during its processing. Simple Power Analysis (SPA) attempts to interpret the power consumption of a device and infer information about the performed operations [2]. This type of analysis has been widely performed to evaluate the implementation robustness of cryptographic algorithms in software and hardware. It is usually the first step to identify vulnerabilities and propose different countermeasures to mitigate these kinds of attacks in RISC-V cores [3].

With the motivation to assess existing RISC-V cores in terms of security, we introduce a methodology to capture power traces and enable the observation of possible vulnerabilities of existing RISC-V cores. To that end, we use the variability of power consumption while running different cryptographic algorithms on them. The methodology encompasses the inclusion of a dedicated hardware IP to control the trigger automatically while running the compiled software code on the core without any dependency.

*Corresponding author: camacho@imse-cnm.csic.es.
This research was supported in part by the QUBIP Project with Grant Agreement No. 101119746 under the EU Horizon Europe research and innovation programme.

Methodology

Our work presents a methodology for evaluating side-channel leakages by executing a compiled software cryptographic algorithm on a RISC-V core, implemented on a Field-Programmable Gate Array (FPGA). Therefore, this methodology opens the door to the evaluation of different RISC-V cores while running cryptographic algorithms, independent of any Operating System (OS). For this work, a standalone platform is used to run the cryptographic software executable (ELF) on the core.

The SAKURA-X board is selected since it is particularly designed for performing side channel analysis (SCA) having two different Xilinx FPGAs: Spartan-6 (XC6LX45-2FGG484C) and Kintex-7 (XC7K160T-1FBGC). In addition to this, the SAKURA-X offers an inbuilt circuitry to measure power consumption traces using an oscilloscope, which can be connected to a PC to observe the traces that allow to conduct Power Analysis.

The experimental setup scheme of our methodology to perform the SCA on a RISC-V core implemented on the SAKURA-X FPGA platform is shown in Figure 1 that only requires an oscilloscope in addition to the Sakura-X board itself. The SoC based on the RISC-V core is implemented on the Kintex-7 FPGA of the SAKURA-X board. This RISC-V SoC contains DDR3, UART, SD, and Ethernet interfaces.

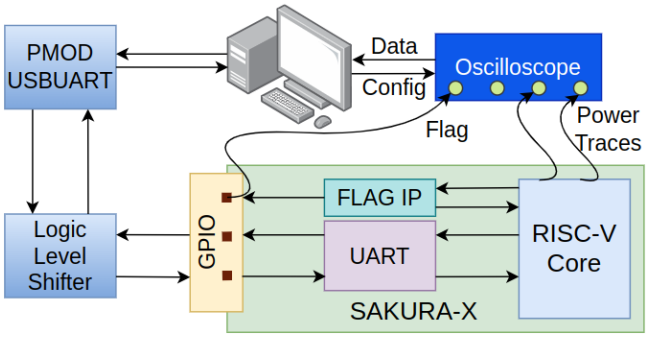

Figure 1: *Experimental Setup*

Firstly, the UART module enables communication between the RISC-V core and the host PC. To that purpose, the SAKURA-X GPIO port is used to assign the UART signals, which then pass through a Pmod LVLSHFT for level shifting (i.e., from 2.7V to 3.3V) before connecting to the Pmod USBUART and finally to the PC's USB port. Secondly, a DDR3 SDRAM is generated using Memory Interface Generator (MIG-7 Serier) with a 16-bit data bus, DDR3 reference clock of 200MHz, and 1GB of memory range. Finally, it is connected to the RISC-V core via an AXI Smartconnect.

In our methodology, the first step is to generate the bare-metal program. To that end, the software code of the implemented cryptographic algorithm is cross-compiled to generate the executable ELF. Then, it is loaded and executed using the bootrom without needing an SD slot, making it more versatile. Then, it is stored on the FPGA flash memory using the generated memory configuration file, followed by booting from the Configuration Memory Device in the Vivado Hardware Manager.

A hardware flag IP module is developed to provide the flag/trigger to the oscilloscope to capture the power traces during the cryptographic algorithm software execution. This flag IP module is designed with an AXI4-Lite interface to communicate with the core, where the flag is controlled from the software by writing to the specific register address corresponding to the hardware IP module. The output flag is connected to a GPIO port of the SAKURA-X board to control the trigger of the oscilloscope. Thereafter, the power traces of the cryptographic operation are captured from the SAKURA-X board, only when the expected flag value is obtained on the corresponding GPIO port.

Methodology verification example

To show that the methodology is working properly, a software implementation of AES-128 is running on the Rocket 64-bit single core [4] at 100 MHz. The implemented core gives the software full access to hardware such as I/O, memory, and interrupt. The power traces of AES operation were captured from the SAKURA-X board, only when the expected flag value (activated when the AES operation is running) is obtained on the corresponding GPIO port using the Flag IP.

Figure 2 illustrates the captured power trace of the AES-128 encryption operation (in blue) and the trigger signal generated by the flag IP (in red) that identifies the operation period.

The traces obtained are very promising, as they suggest that by simply capturing a sufficient number of traces and performing a visual inspection, we can first achieve results through SPA, and later apply more powerful techniques such as Differential power analysis (DPA) and Correlation Power Analysis (CPA).

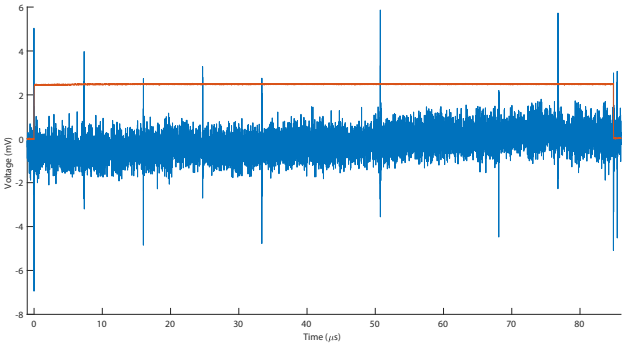

Figure 2: *Power trace of AES-128 Encryption*

Conclusions

A methodology has been proposed to enable the security assessment of RISC-V cores. The methodology presented herein is valid for different types of cores and it minimizes the experimental setup (only requires a Sakura board and an oscilloscope). The methodology is based on the use of the Sakura board where a RISC-V core and a hardware module that allows to trigger the execution of the algorithm on it, are implemented to capture the traces. This trigger is controlled from the software without depending on any external inputs. To show its correct functionality, power traces are captured while running an AES over a Rocket core, which enables SCA techniques, such as SPA, DPA, and CPA, but the methodology is extensible to other cryptographic algorithms, and other RISC-V cores.

References

[1] Hittu Garg and Mayank Dave. "Securing IoT Devices and SecurelyConnecting the Dots Using REST API and Middleware". In: *International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)* (2019). DOI: 10.1109/IoT-SIU.2019.8777334.

[2] Paul Kocher, Joshua Jaffe, and Benjamin Jun. "Differential power analysis". In: *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*. Springer. 1999, pp. 388–397.

[3] Abolfazl Sajadi et al. "A Systematic Comparison of Side-channel Countermeasures for RISC-V-based SoCs". In: *Nordic Circuits and Systems Conference (NorCAS)* (2024). DOI: 10.1109/NorCAS64408.2024.10752477.

[4] *Vivado-RISC-V*. URL: https://github.com/eugene-tarassov/vivado-risc-v.