Unveiling the Invisible Threads: Dynamic Information Flow Tracking and the Intriguing World of Fault Injection Attacks

William PENSEC[1], Vianney LAPÔTRE[1] and Guy GOGNIAT[1]
[1]UMR 6285, Lab-STICC, Université Bretagne Sud, Lorient, France
firstname.lastname@univ-ubs.fr

1 Introduction

Different software attacks, such as buffer overflow, SQL injections, and malware, can be detected using Dynamic Information Flow Tracking (DIFT) techniques. These techniques involve attaching and propagating tags to information containers at runtime, allowing for the detection of malicious behavior. The literature has explored various implementations of DIFT, including hardware, software, and hybrid approaches [1]. Depending on the type of DIFT used, information containers can range from files to registers.

Hardware DIFT solutions can be classified into two categories: off-core and in-core. Off-core DIFT relies on a dedicated co-processor for tag-related operations, eliminating the need for internal modifications to the processor. In-core DIFT involves modifying the internal structure of the processor. Tag-related operations are distributed across the pipeline stages and executed in parallel with data treatments.

This work focuses on the D-RI5CY processor, which implements the in-core DIFT proposed in [2]. Our objective is to investigate the impact of Fault Injection Attacks (FIA) on the efficiency of the D-RI5CY DIFT mechanism. We conduct fault injection simulations to evaluate the sensitivity of the D-RI5CY DIFT and identify the hardware components related to DIFT that require protection.

2 Motivation

Fault Injection Attacks (FIAs) can be conducted through power supply or clock perturbations, EM pulses, or laser shots. The impact of each injection method varies. Laser-based injections offer precise spatial and temporal control, while power supply or clock perturbations affect the entire circuit, resulting in limited spatial precision.

Numerous studies have demonstrated the vulnerabilities of critical systems to FIAs, glitch injections on the power supply have been used to manipulate the program counter (PC) as shown in [3]. These physical attacks effectively bypass protection mechanisms, allowing attackers to hijack the targeted system.

Figure 1 provides an overview of the D-RI5CY processor, highlighting DIFT-related components in red. These components store, propagate, and verify tags during sensitive application execution. The security policy is configured through two Control and Status Registers (CSRs) named TPR and TCR.

In this study, we propose combining software and physical attacks to overcome the implemented DIFT mechanism in the D-RI5CY processor. The analysis results will aid in building a robust DIFT mechanism that considers both software and physical attacks in future work.

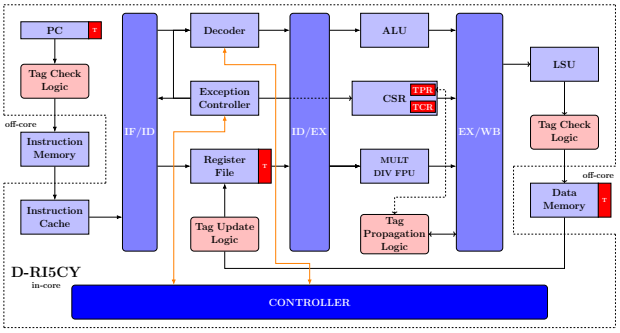

Figure 1: *Overview of the D-RI5CY processor*

3 Main results

We assume an attacker capable of injecting faults into the registers associated with DIFT-related components, as certain physical faults can violate setup/hold time constraints in flip-flops. We consider 3 types of injections: set to 0, set to 1, or a bit-flip at a random position of the targeted register.

To study the behaviour of DIFT against FIAs, we carried out a fault injection campaign in simulation, considering four different case studies. For the four cases, 72 of the 3726 simulations lead to a successfull attack (1.93%). This campaign revealed 12 critical DIFT-related registers out of 54. We observe that the vulnerabilities are associated with data paths made up of AND gates.

References

[1] Kejun Chen et al. "Dynamic Information Flow Tracking: Taxonomy, Challenges, and Opportunities". In: *Micromachines* (2021). DOI: 10.3390/mi12080898.

[2] Christian Palmiero et al. "Design and Implementation of a Dynamic Information Flow Tracking Architecture to Secure a RISC-V Core for IoT Applications". In: *High Performance Extreme Computing*. 2018. DOI: 10.1109/HPEC.2018.8547578.

[3] Niek Timmers et al. "Controlling PC on ARM Using Fault Injection". In: *Fault Diagnosis and Tolerance in Cryptography (FDTC)*. 2016. DOI: 10.1109/FDTC.2016.18.