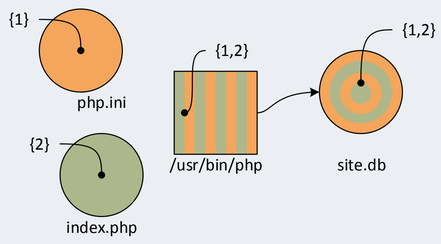
HardBlare: an efficient hardware-assisted DIFC for non-modified embedded processors

Pascal Cotret [α], Guillaume Hiet [β], Guy Gogniat [γ], Vianney Lapôtre [γ]

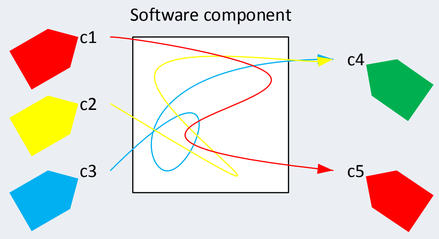
[α] SCEE/IETR, CentraleSupélec, Cesson-Sévigné - FRANCE
[β] CIDRE/INRIA, CentraleSupélec, Cesson-Sévigné - FRANCE
[γ] Lab-STICC, University of South Brittany, Lorient - FRANCE

Introduction

Classic IT security context: main goals are **security policy** definition (in terms of confidentiality/integrity), its enforcement (malicious content filtering, isolation...) and its **monitoring** (intrusion detection, vulnerability analysis).
DIFC (**Dynamic Information Flow Control**) is an approach aiming to ensure that security properties are preserved all along the execution and files storage.



Containers of information: files, variables... Labels (also known as **tags**) are attached to such containers.



Information flow policy ⇒ relation between tags.

Software-Based DIFC

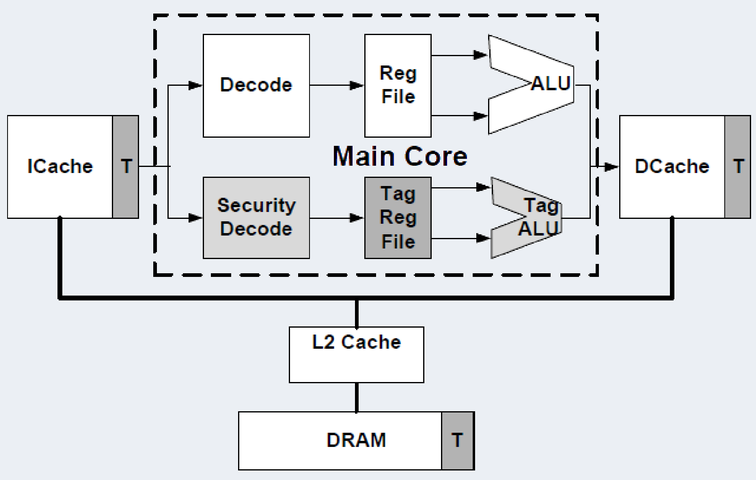
Analysis can be done at the operating system and application levels:

- **OS-level**: dedicated OS or modification of existing OS.
  - ✓ Small overhead ($< 10\%$).
  - ✗ Overapproximation issue.
- **Application-level**: machine code or specific language.
  - ✓ Gain in precision (hybrid analysis).
  - ✗ Huge overhead ($> \times 3$ at least!)

Goals

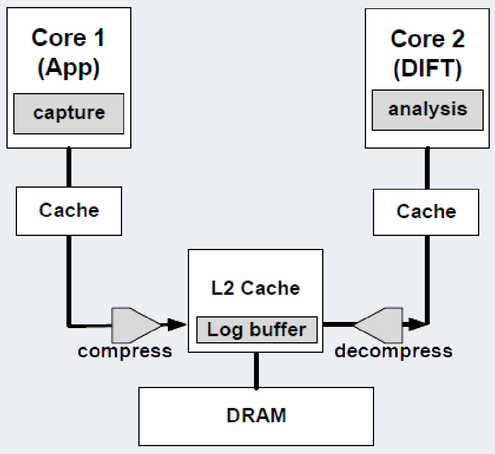
- Modify existing hardware to **accelerate** IFC and **protect** tags.
- **Compatibility** with existing software/hardware.
- Design of a **flexible solution** (no dependency with policy or tag format).

Existing Hardware-Assisted Approaches



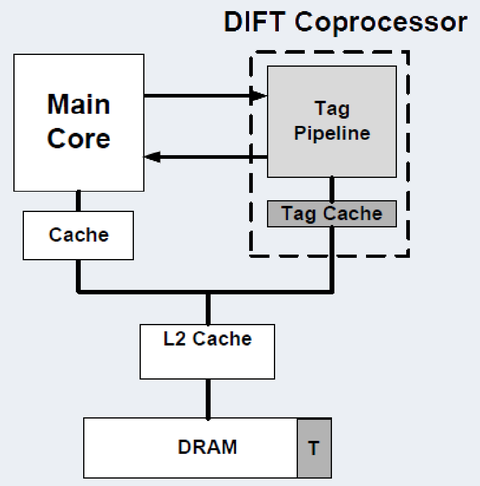
[1] In-core DIFT

✓ Minimal impact on speed
✗ Tag propagation in core pipeline + not portable



[2] Offloading DIFT

Separate processor for DIFC controls
+ tag compression/decompression
Inter-core logic + power consumption $\times 2$

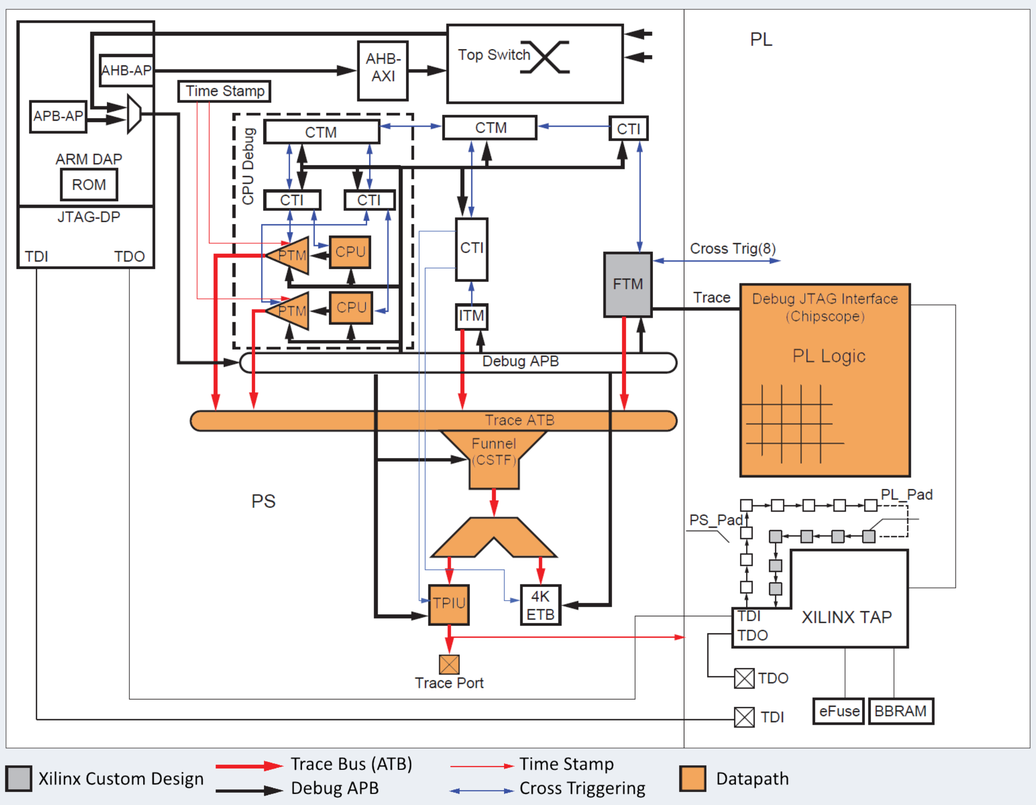


[3] Off-core DIFT

Dedicated co-processor
+ nearly no modification of the main processor
Main core may stall

Source: Hari Kannan, *Building Hardware Systems for Information Flow Tracking*, PhD defense (2010). `http://csl.stanford.edu/~christos/publications/2010.hari_kannan.phd_thesis.slides.pdf`

HardBlare Approach



HardBlare targets **heterogeneous SoCs** combining a hardcore processor based on the **ARMv7** architecture with a **FPGA** (Zynq from Xilinx, SoC from Altera).

Main processor must send a **tuple** composed of the following elements:

- Instruction pointer (i.e. program counter).
- Instruction encoding (fine-grained level).
- Memory addresses and processed registers (in case of load/store instructions).

As we cannot modify the ARM chip, how can we get the needed information from the main processor ?

- A **trace mechanism** should help to perform DIFC controls.
- Processor/coprocessor **synchronization**:
  - Performed on system calls (a few hundred cycles).
  - Synchronization time overhead: around ten cycles.

⇒ **Evaluation** in terms of **performances** and **security** on a full-fledged Linux system with standard binaries.

Source: Xilinx Inc., *Zynq-7000 All Programmable SoC*, Technical Reference Manual (2015). `http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf`

Some References

[1] M. Dalton, H. Kannan, and C. Kozyrakis, "Raksha: A flexible information flow architecture for software security," in *In International Symposium on Computer Architecture (ISCA*, 2007.

[2] V. Nagarajan, H.-S. Kim, Y. Wu, and R. Gupta, "Dynamic information tracking on multicores," in *12th Workshop on the Interaction between Compilers and Computer Architecture (INTERACT)*, Feb 2008.

[3] H. Kannan, M. Dalton, and C. Kozyrakis, "Decoupling dynamic information flow tracking with a dedicated coprocessor," in *Proceedings of the 2009 IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2009, Estoril, Lisbon, Portugal, June 29 - July 2, 2009*, pp. 105–114, 2009.

[4] G. Venkataramani, I. Doudalis, Y. Solihin, and M. Prvulovic, "Flexitaint: A programmable accelerator for dynamic taint propagation," in *In 14th International Symposium on HighPerformance Computer Architecture (HPCA-14*, 2008.

Main Contributions at a Glance

- Hardware-assisted DIFC system with limited time overheads.
- Approach based on a non-modified CPU with a standard Linux and generic binaries ⇒ Could be implemented by industrial partners in medium-term.
- Hardened with hardware security mechanisms: trusted coprocessor storage and bus protection in terms of confidentiality/integrity.
- Contributions on software-related issues as well (static/dynamic IFC analysis, aka hybrid analysis).
- Perspectives on runtime reconfiguration and multicore/manycore systems.