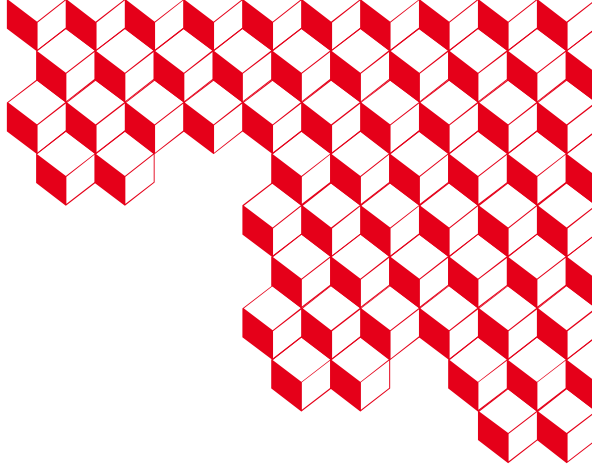


Intrinsically secure applications processors

Olivier SAVRY – Melchior de BEAUDRAP

CEA LETI

PROCESSORS VULNERABILITIES

Spectre & Covert channels

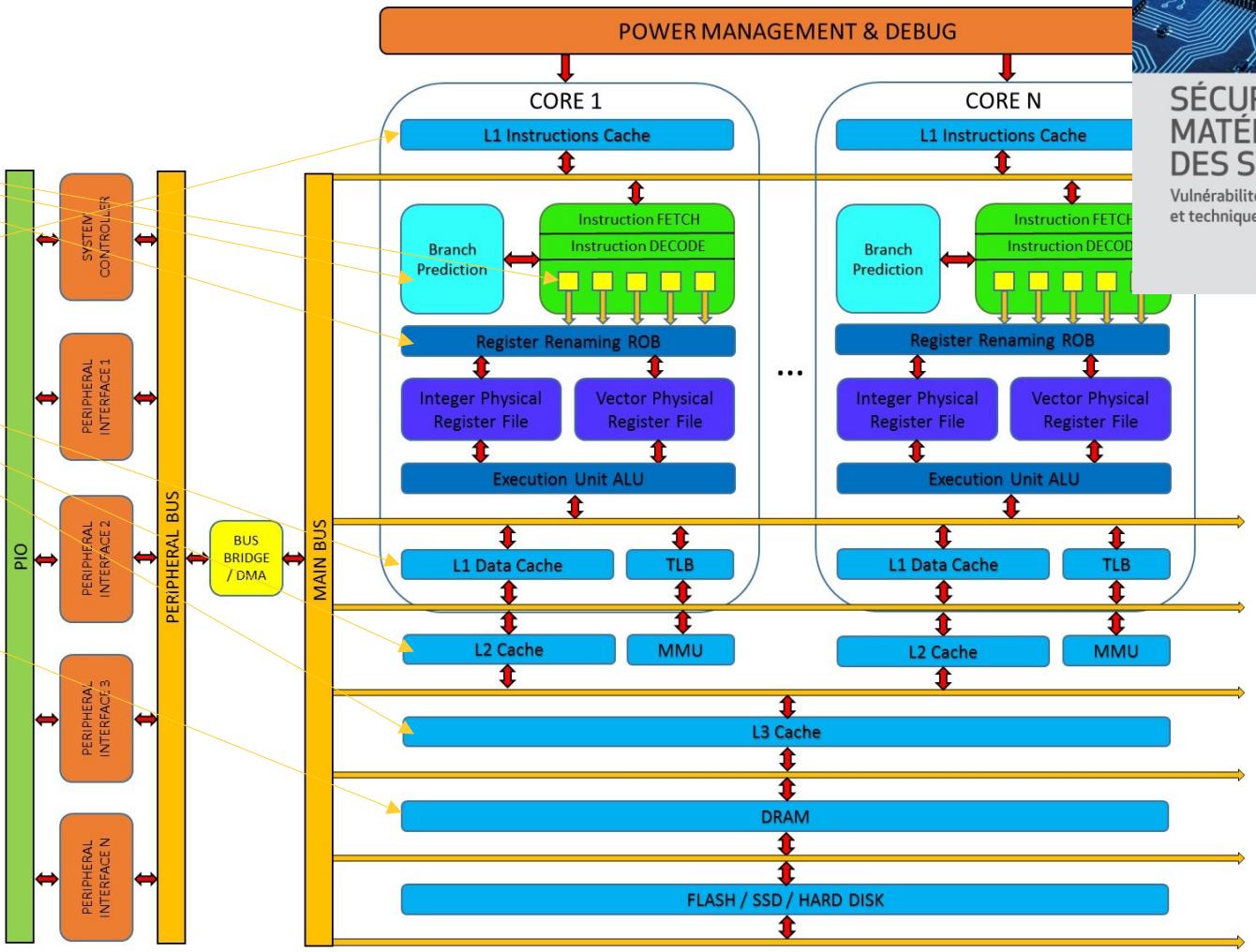
cache timing side-channel
- Flush + Reload
- Prime + Probe
- Flush + flush
- Cachebleed
- Evict + Time
- TLB

DRAM Memory:
- RowHammer

Side Channel Attack

Fault Injection
- Laser
- Glitch,...

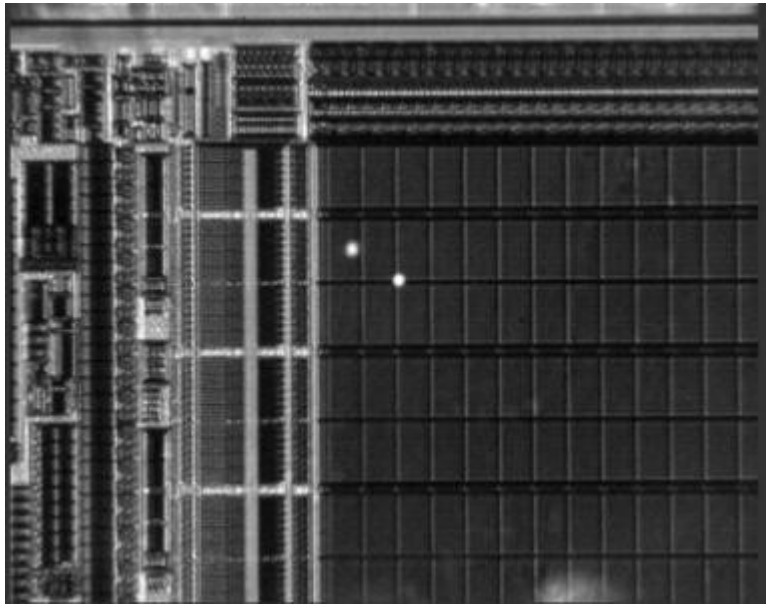
Memory Leakage :
- Spatial and temporal memory safety



Problem : lack of Confidentiality, Integrity and Authenticity of data and instructions

Issues

- **Confidential Computing** solutions force an impossible choice :
  - the extreme slowness of **FHE** (Fully Homomorphic Encryption)
  - or the incomplete protection of **TEEs** (Trusted Execution Environments)
- **Integrity of computation :**
  - How to be sure that the **result of computation is true** ?
  - Generally the solution is **redundancy** (ECC, lock step, …)
    - To be confronted with Multi-beam Laser Injection



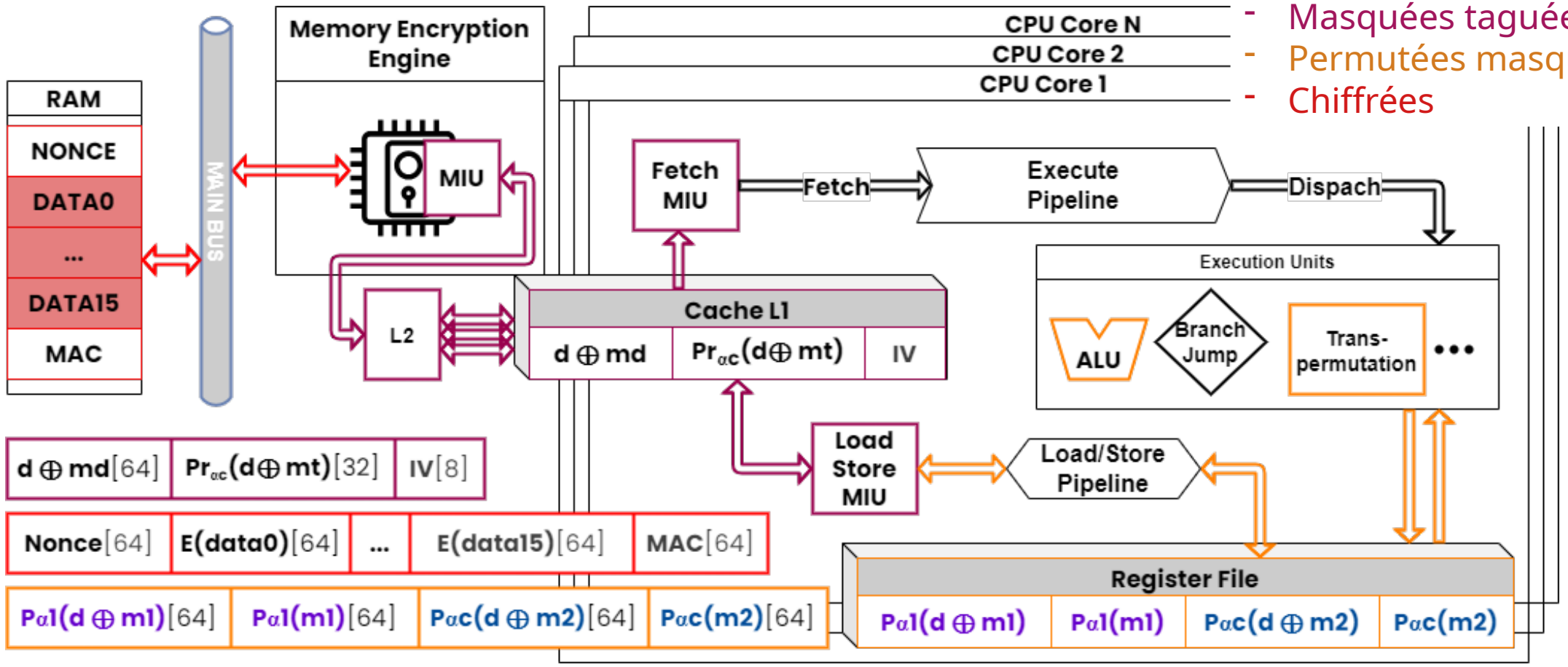
Solution : An Intrinsically Secure Application Processor

- **CONCEPTS :**
  - **Regardless of any vulnerabilities remaining in the software code, this processor guarantees they remain unexploitable**
  - **Security objectives : end to end confidentiality, integrity, authenticity of data and instructions : from FLASH to execution units included**
    - **A disruptive holistic approach that requires extensive integration efforts**
    - **Generally, a vulnerability is fixed, not a set of vulnerabilities**
  - **Trusted implementation**
  - **Performant : superscalar out of order processor : Naxriscv 64 bits**

Intrinsically Secure Application Processor

La stratégie Confidentialité/Intégrité/Authenticité se base sur le chiffrement de la RAM et la permutation et masquage des données dans le cache et le CPU.

Données :
- Masquées taguées
- Permutées masquées
- Chiffrées



Sécurisation de processeurs applicatifs

1. Chiffrement de la ram
2. Masquage
3. Homomorphisme (permutation)
4. Polymorphisme
5. Implémentation

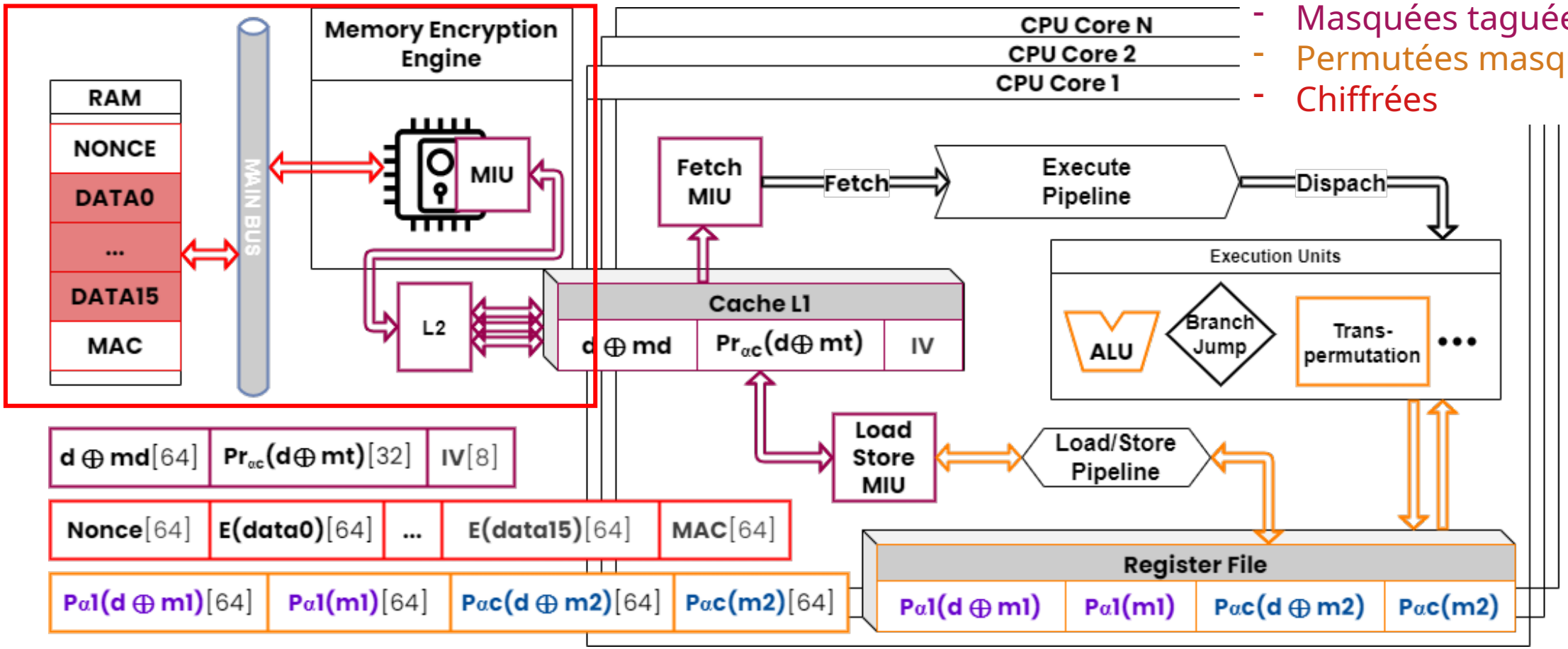
1 ■ Chiffrement de la ram

Confidentialité et Intégrité

Intrinsically Secure Application Processor

La RAM est chiffrée.

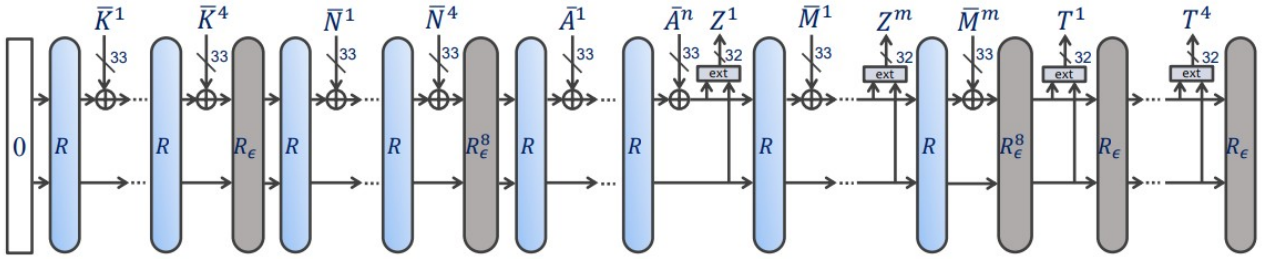
Données :
- Masquées taguées
- Permutées masquées
- Chiffrées



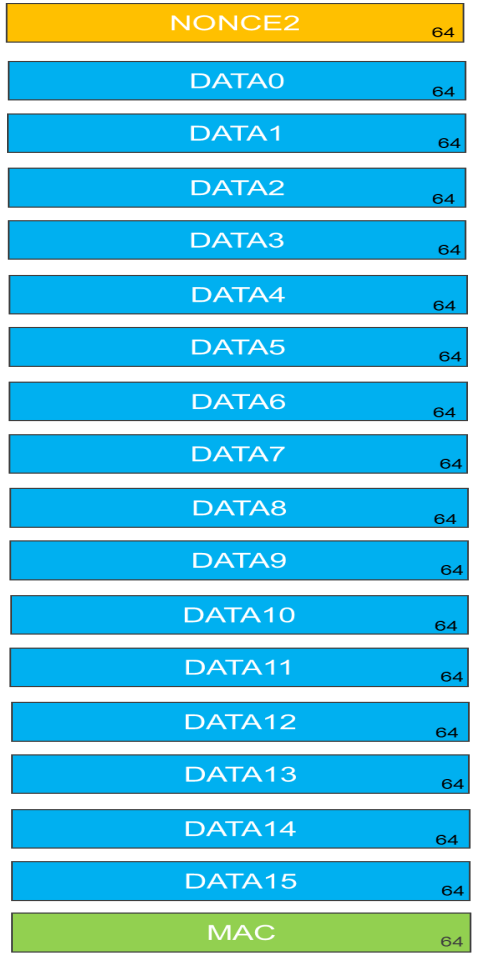
Macroscopic structure of DRAM encryption

For a 64-bit processor:

- Authenticated encryption by block of 16 data of 64 bits
- Width of a cache line: 128 bytes
- Overhead memory space : +12,5 %, boot Linux 30% slower
- AE algorithm: Subterranean 2.0
- Key: 128 bits, Nonce: 128 bits, MAC: 64 bits
- 2 times faster, 2 times smaller than ASCON :
  - throughput 2.4 cycles/word of 64 bits
- Block shape is transparent to the CPU
- Detect Rowhammer attacks



Nonce=f(@)



K. Ait Lahssaine, O. Savry. CIAMH :Confidentiality, authenticity and integrity across the memory hierarchy, RISC-V Europe Summit 2025.

3. Masquage

Confidentialité

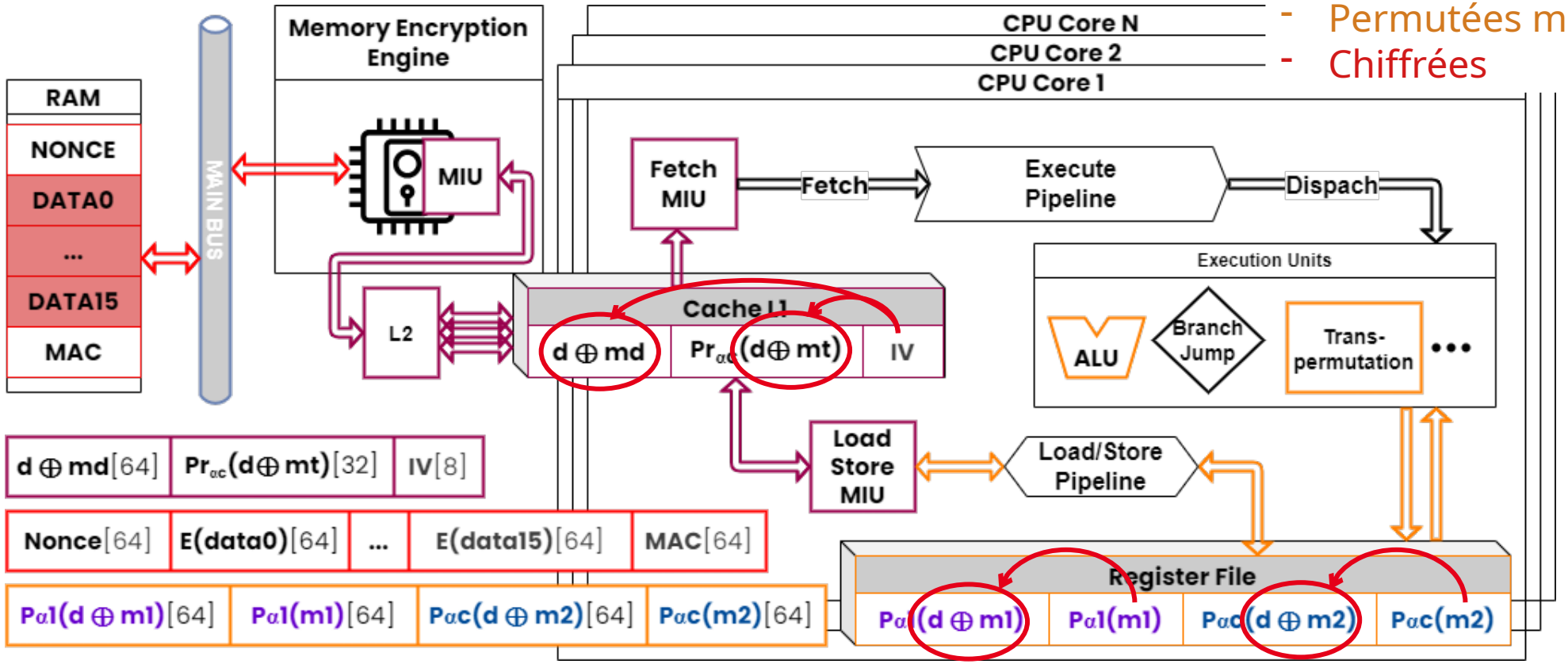
Masquage

Toute les données dans le CPU et le Cache sont Masquées :

- Xor entre la donnée et une valeur aléatoire (masque)

Dans le cache, une seed(IV) combinée à l'adresse, sert à générer les 128 bits de masques pour éviter de les stocker

Données :
- Masquées taguées
- Permutées masquées
- Chiffrées



2. Homomorphisme

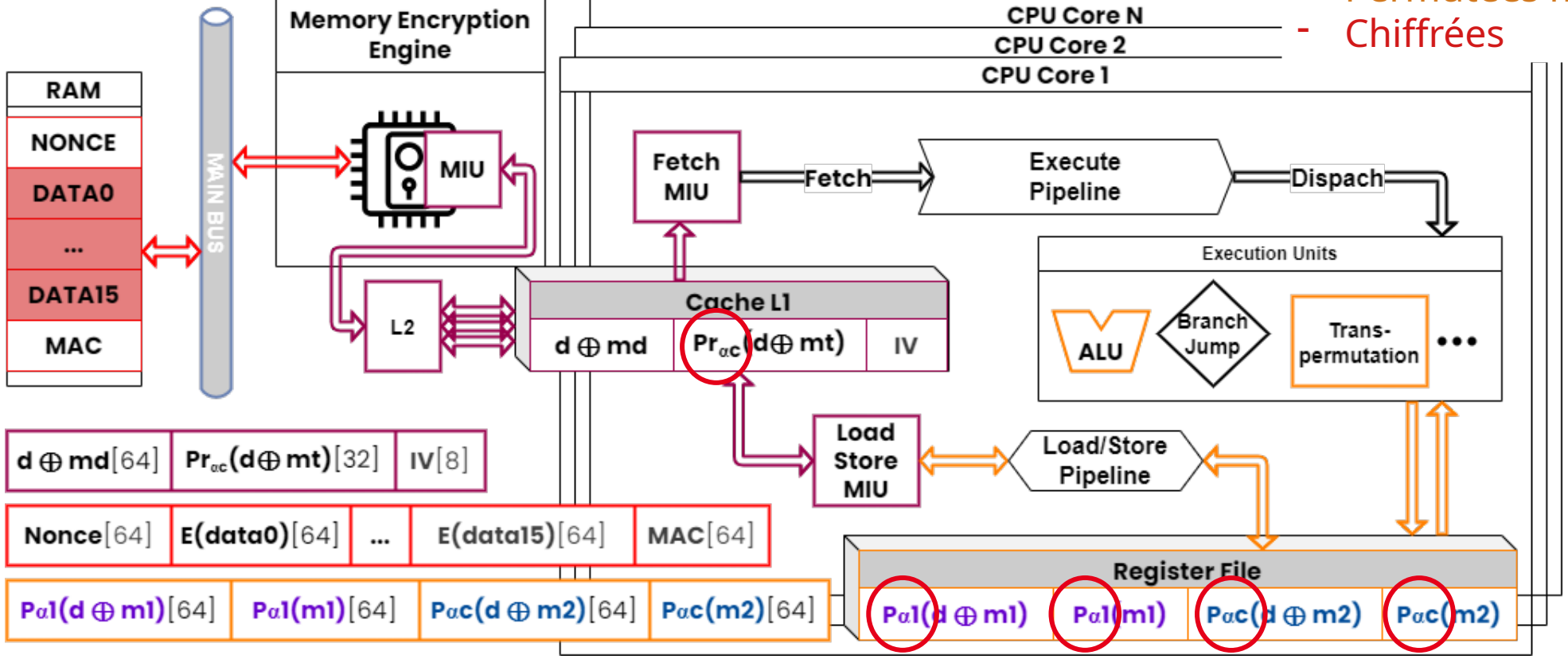
Permutation : Intégrité

Homomorphisme

Les données dans le CPU sont permutées et redondées

Les données Cache sont redondées

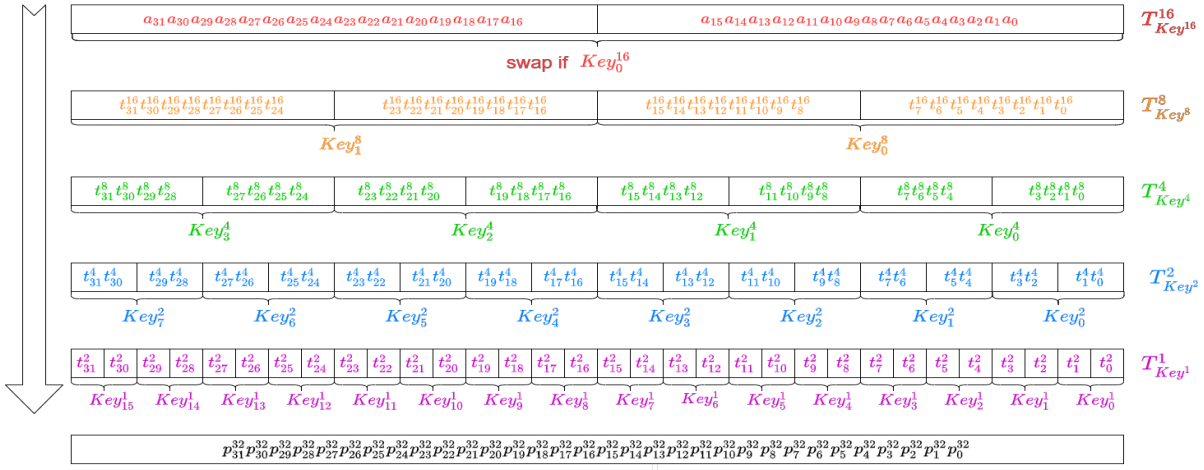
Données :
- Masquées taguées
- Permutées masquées
- Chiffrées



AKHACIA : Homomorphic integrity tags

A bitwise permutation

- **Homomorphic for logic and arithmetic** computation
  - Easy verification and proof of results
- **Dependant of a secret key**
  - To blur fault injection setup
  - Hardware polymorphism when the key is changed
- Compatible with **masking** for confidentiality
- AND, OR, XOR, SHIFT and addition and multiplication are implemented
  - Karatsuba permuted multiplication in 1 cycle
- But also **SIMD instructions** and multiple floating point formats
  - Suitable for **IA and NN** computation
  - FP32, FP16, Q7, Q15 additions and multiplications
- Bit manipulation routines implemented
- Implementation in **VASCO2 and VASCO3**
- **Security evaluation with Laser Faults Injection**

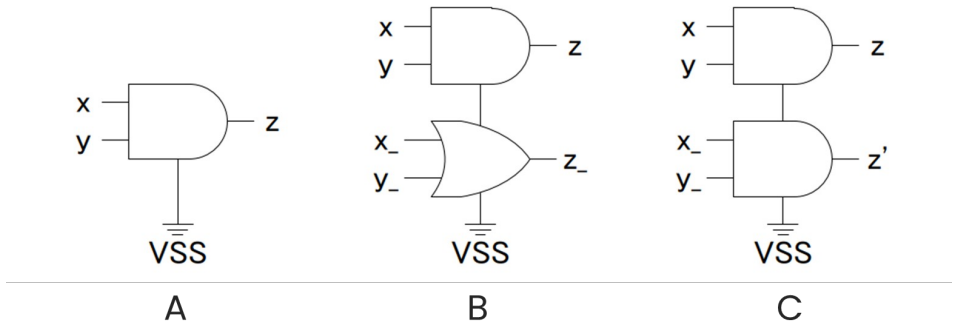


$$P_{Key}^{32} = P_{Key^{16}|Key^8|Key^4|Key^2|Key^1}^{32} = T_{Key^{16}}^{16}(T_{Key^8}^{8}(T_{Key^4}^{4}(T_{Key^2}^{2}(T_{Key^1}^{1}))))$$

Masked and permuted ALU with HDRL

(**H**omogeneous **D**ual-**R**ail **L**ogic)

- Numerous first-order masking tests: DOM, SESYM, with delay management -> **Too slow, too big!**
- Return to a solution with an “acceptable” compromise: drastic reduction in SNR
- Use of HDRL



A : classical AND **B**: AND WDDL **C**: AND HDRL

x	y	AND(x,y)	OR(x,y)	NOT(x)	x′	y′	AND(x′,y′)	OR(x′,y′)	NOT(x′)
0→0	0→0	0 →0	0 →0	1 →1	1→1	1→1	1 →1	1 →1	0 →0
0→0	0→1	0 →0	0 →1	1 →1	1→1	1→0	1 →0	1 →1	0 →0
0→0	1→0	0 →0	1 →0	1 →1	1→1	0→1	0 →1	1 →1	0 →0
0→0	1→1	0 →0	1 →1	1 →1	1→1	0→0	0 →0	1 →1	0 →0
0→1	0→0	0 →0	0 →1	1 →0	1→0	1→1	1 →0	1 →1	0 →1
0→1	0→1	0 →1	0 →1	1 →0	1→0	1→0	1 →0	1 →0	0 →1
0→1	1→0	0 →0	1 →1	1 →0	1→0	0→1	0 →0	1 →1	0 →1
0→1	1→1	0 →1	1 →1	1 →0	1→0	0→0	0 →0	1 →0	0 →1
1→0	0→0	0 →0	1 →0	0 →1	0→1	1→1	0 →1	1 →1	1 →0
1→0	0→1	0 →0	1 →1	0 →1	0→1	1→0	0 →0	1 →1	1 →0
1→0	1→0	1 →0	1 →0	0 →1	0→1	0→1	0 →1	0 →1	1 →0
1→0	1→1	1 →0	1 →1	0 →1	0→1	0→0	0 →0	0 →1	1 →0
1→1	0→0	0 →0	1 →1	0 →0	0→0	1→1	0 →0	1 →1	1 →1
1→1	0→1	0 →1	1 →1	0 →0	0→0	1→0	0 →0	1 →0	1 →1
1→1	1→0	1 →0	1 →1	0 →0	0→0	0→1	0 →0	0 →1	1 →1
1→1	1→1	1 →1	1 →1	0 →0	0→0	0→0	0 →0	0 →0	1 →1

- Duplication of the circuit but with inverted inputs
  - Same probability of 1, same number of transitions
  - But requires AND(x',y') to be indistinguishable from AND(x,y): OK for power measurement but not in EM
- Implementation of the HDRL AND gate on a single LUT6_2 (6 inputs, 2 outputs) then parallel and adjacent routing
- Complete reverse engineering of Xilinx Artix7 and Kintex7 FPGAs
  - Made possible by Rapidwright from the X-ray project
  - Routing of HDRL pairs with Vivado, rerouting of x' (and y') and adjacent rerouting to x (and y) with Rapidwright
  - Many different types of wires, not necessarily compatible

4. Polymorphisme

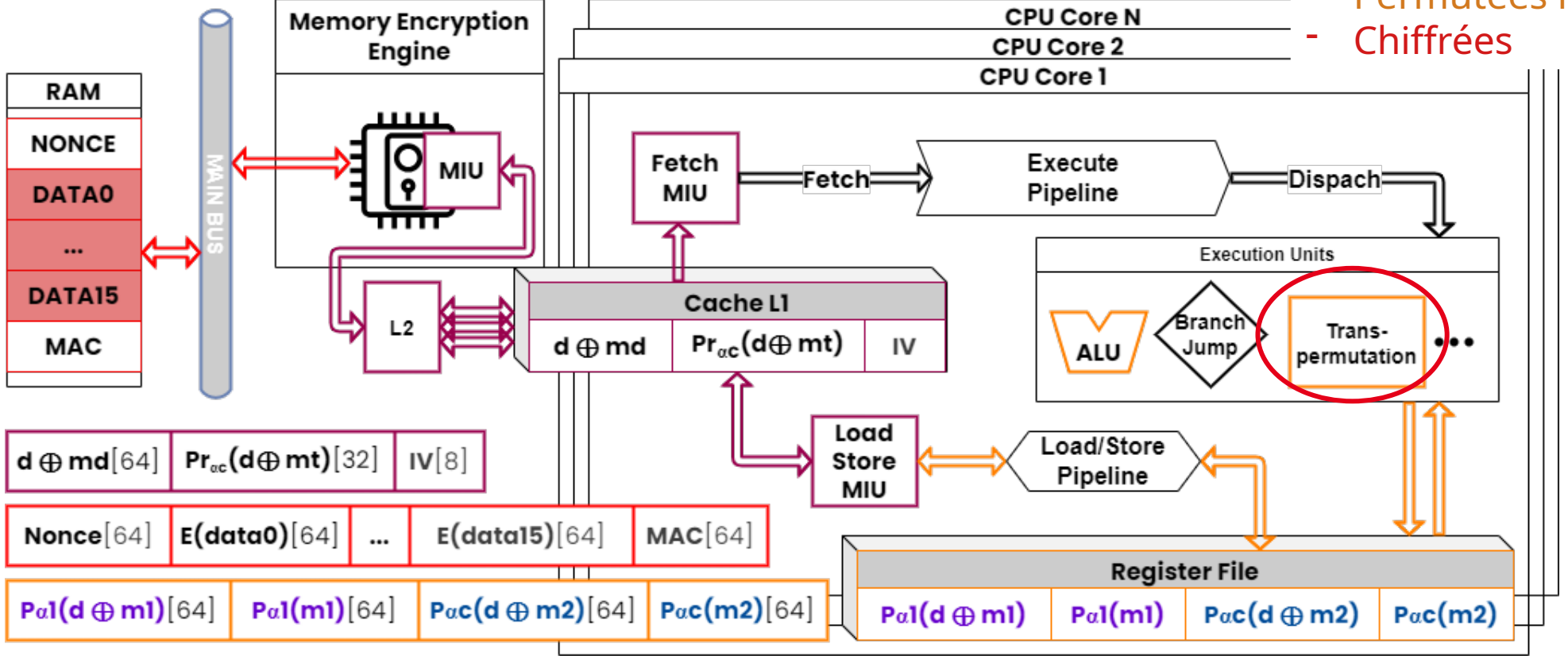
Intégrité

Polymorphisme

La clef de permutation dans le cœur change régulièrement :

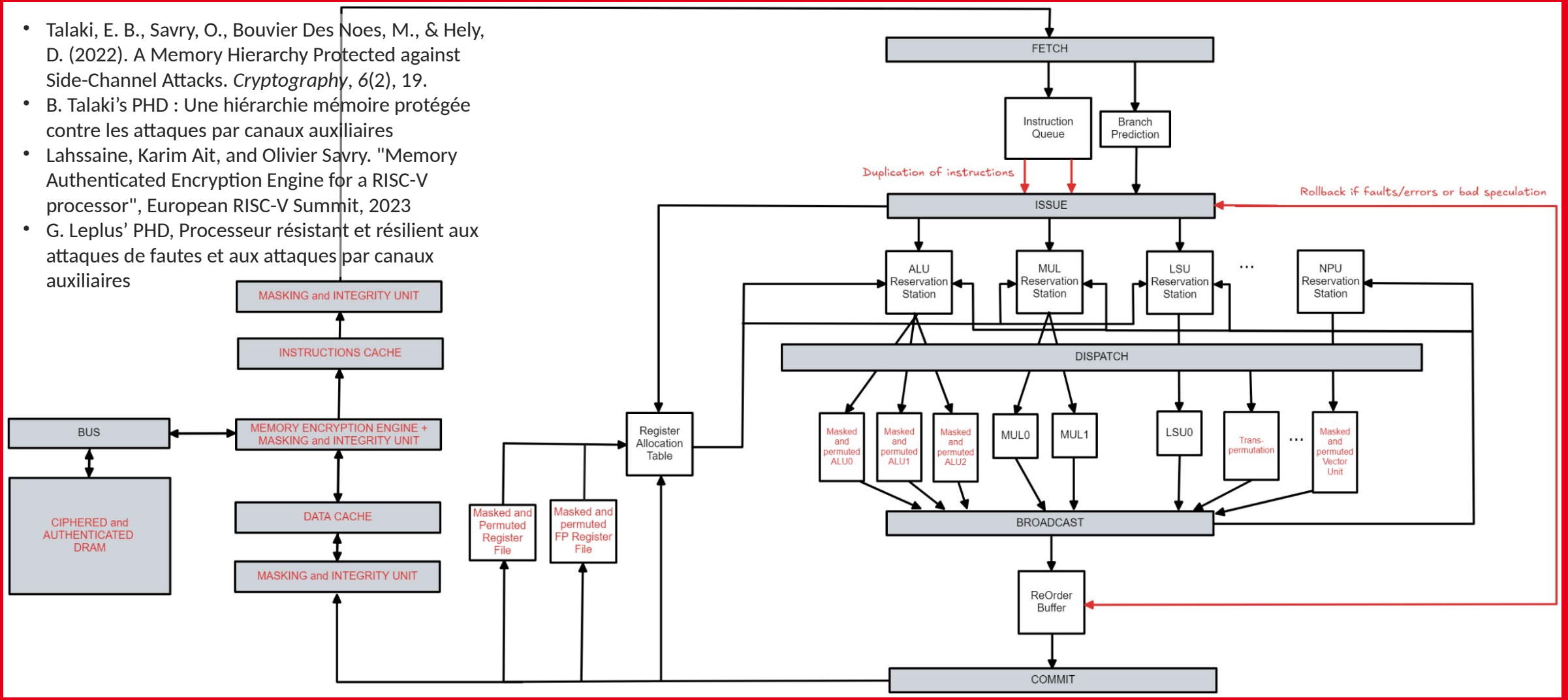
Les valeurs de chaque registres sont trans-permutées

Données :
- Masquées taguées
- Permutées masquées
- Chiffrées



Secure Superscalar Out of Order Processor

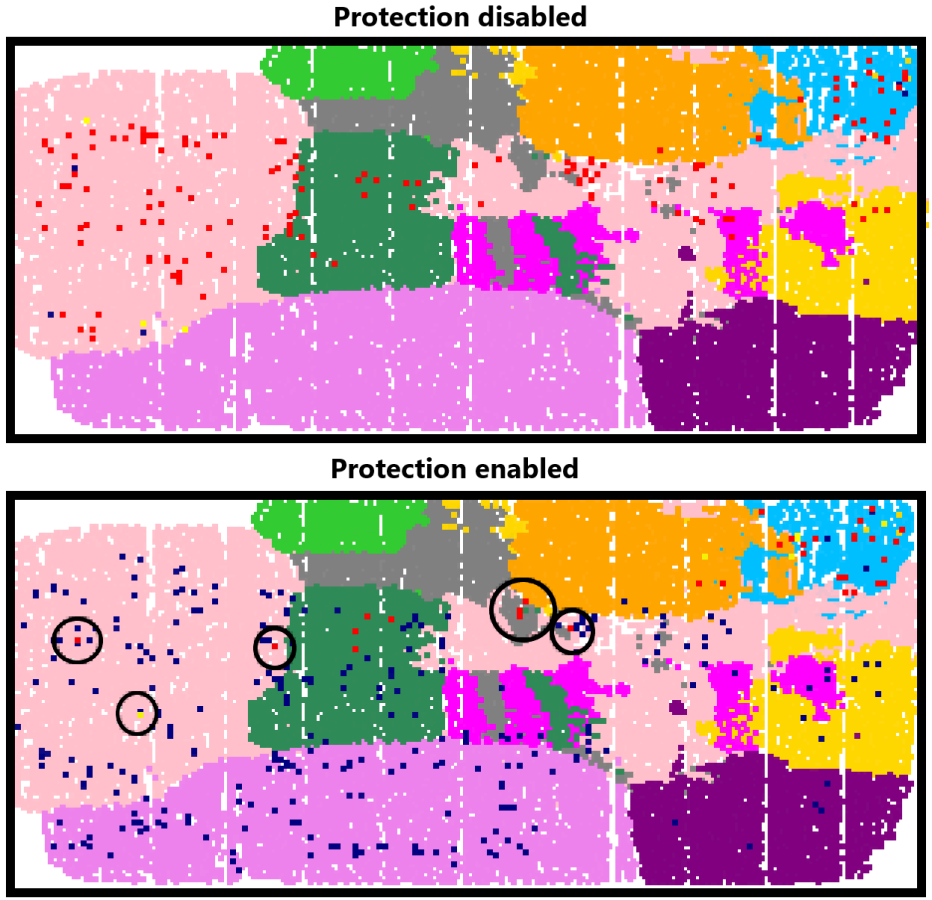
- Talaki, E. B., Savry, O., Bouvier Des Noes, M., & Hely, D. (2022). A Memory Hierarchy Protected against Side-Channel Attacks. *Cryptography*, *6*(2), 19.
- B. Talaki's PHD : Une hiérarchie mémoire protégée contre les attaques par canaux auxiliaires
- Lahssaine, Karim Ait, and Olivier Savry. "Memory Authenticated Encryption Engine for a RISC-V processor", European RISC-V Summit, 2023
- G. Leplus' PHD, Processeur résistant et résilient aux attaques de fautes et aux attaques par canaux auxiliaires

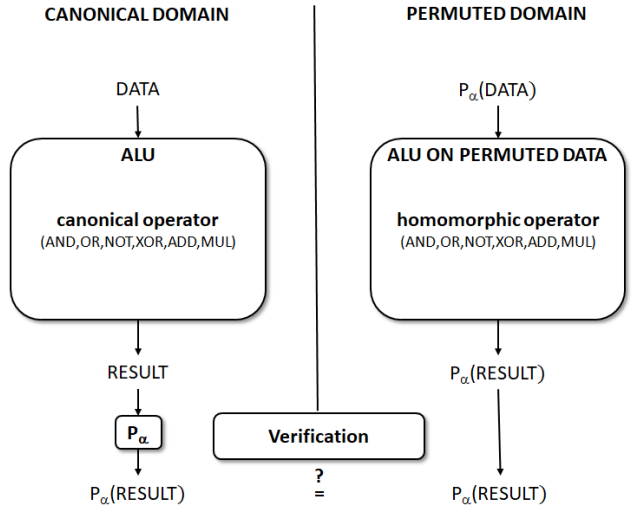


5. Implémentation

Cva6 - NaxRiscV – FPGA -ASIC

Evaluation sécuritaire sur ASIC
Tag d'intégrité homomorphes





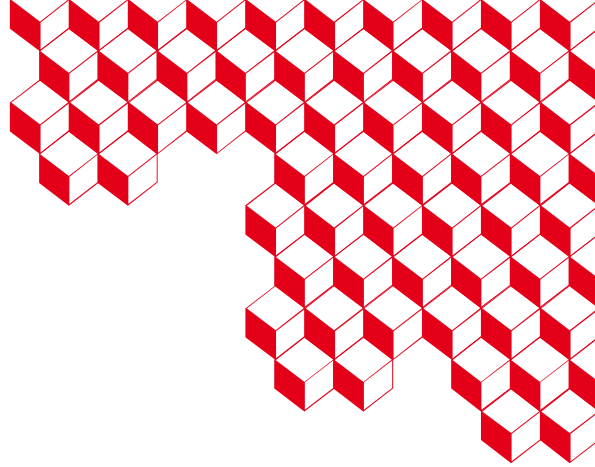
- Implémenation des tags d'intégrité homomorphe sur ASIC en technologie FDSOI 22nm
- ALU (ADD/SUB, AND, OR, XOR) et shifter canonique et permuté
- MUL et DIV seulement canonique
- Attaque sur AES 128 logiciel
- Les fautes qui permettaient un résultat différent (et ainsi une attaque DFA) sont fortement diminuées
- Elles restent dans les parties non protégées comme le FETCH (les instructions) et le MUL

Performance - NaxRiscV - FPGA

- Naxriscv Superscalaire Out of Order 64 bits avec
  - Moteur de chiffrement authentifié
  - Caches masqués
  - Tags d'intégrité homomorphes dans les caches
  - Banc de registres masqués et permutés (tags homomorphes)
  - ALU masquée et permutée (calculs effectués sur les tags d'intégrité homomorphes)
- Performances

	Non Secure Nax	Secure Nax	comparaison	CVA6	OpenTitan
Fréquence MAX	125	70	**-44%**	50	15
Coremark	4,52/MHz	3,31 /MHz	**-27 %**	2,5 /MHz	?
LUT	32623	101922	**+212 %**	45000	50000
FF	24020	37862	**+57 %**	30000	35000
Boot Linux	223 s	291 s	**-30 %**	?	?

Démonstration



Merci