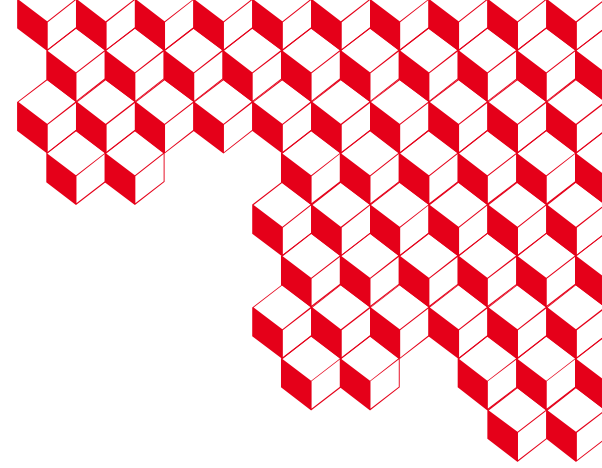




Beyond Redundancy: Hardware Polymorphism Against Multi-Beam Fault Injection

Olivier Savry — CEA-Leti, Université
Grenoble Alpes

Multi-beam injection breaks spatial redundancy

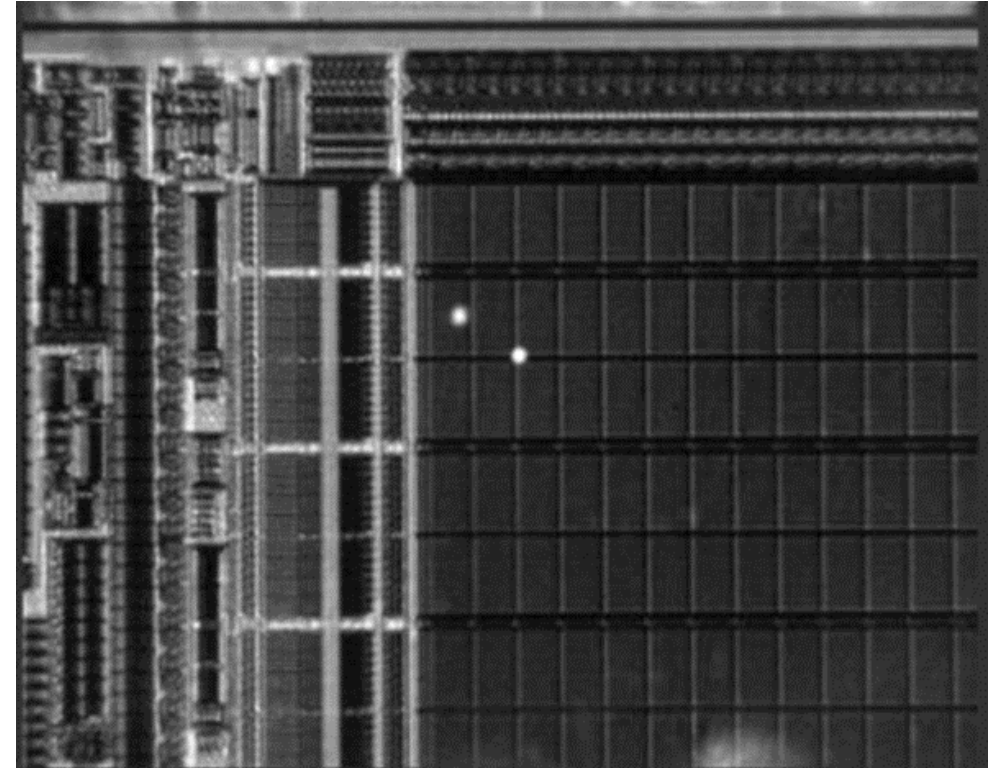
No longer a single, well-localised shot

- 2 spots already defeat a DFA-protected AES (FDTC 2016)
- 4 synchronised spots today (CARDIS 2021)
- Many more if the shots are spread in time

Redundancy answers with replicas

- Detect k faults: $m \geq k + 1$
- Correct by voting: $m \geq 2k + 1$
- RISC-V TMR cores: 100 to 200 % area

**$k = 4$ spots demand five-way replication just to detect.
Pure replication is hitting an economic wall.**



Two laser spots fired simultaneously on the back side of a die

Beyond redundancy: polymorphism



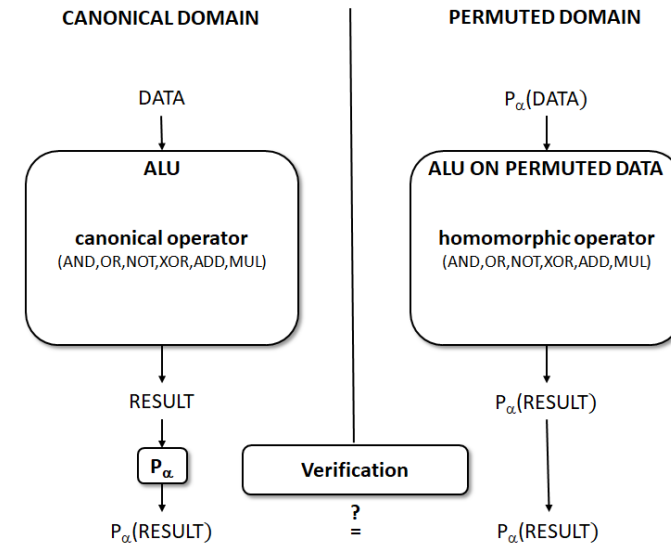
Beyond redundancy: polymorphism

Randomise the wires, do not add copies

- Secret key drawn on-chip, renewed at runtime
- The attacker can no longer pre-compute where to hit

Contribution

- A keyed bit permutation P_α as the redundant share
- Every ALU operator rebuilt homomorphically
- Five pipeline checkpoints, trappable exception
- Re-keying on the fly, no cleartext exposed
- Silicon: CV32E40P, 22 nm FDSOI, laser bench



+74 % area
at iso-frequency, 100 MHz

1 31 → 22
exploitable faults, single-beam
campaign

31-bit key
renewed in 31 cycles, no cleartext
exposure

Adversary model

What the adversary can do

- Full physical access to the die
- Knows the design and the layout entirely
- k simultaneous spots, any number over time
- Faults: set, clear, stuck-at, whole cell
- Target: register file, ALU operands, LSU

What stays out of reach

- α drawn on-chip, never leaves the core
- $P_\alpha(x)$ internal: no clean pair is exposed
- A failed probe is detected, and detection re-keys

Assumed protected elsewhere

- Instruction path: control-flow integrity
- Memory: authenticated encryption, ECC
- Key registers: replicated, zero key forbidden

A keyed homomorphic permutation

We need $f(x \text{ op } y) = f(x) \text{ op}' f(y)$

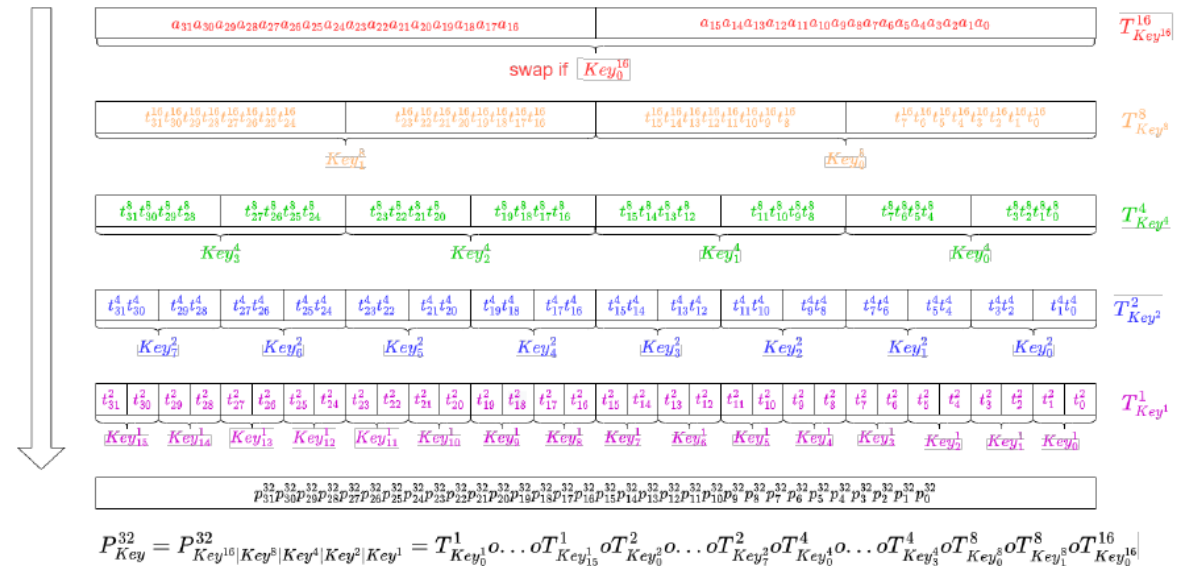
- A cheap consistency check, not encryption

Bitwise homomorphisms = coordinate re-indexings

- Information-preserving iff bijective
- So the share must be a bit permutation
- Free: every Boolean operator, and invertibility

Five-level tree of Fredkin gates, 31 key bits

- Block locality kept: shifts stay tractable
- A group of order 2^{31} , closed under composition
- Static codes would expose a fixed target



The 32-bit permutation as a five-level tree of conditional transpositions

Rebuilding the ALU: Boolean operators and addition

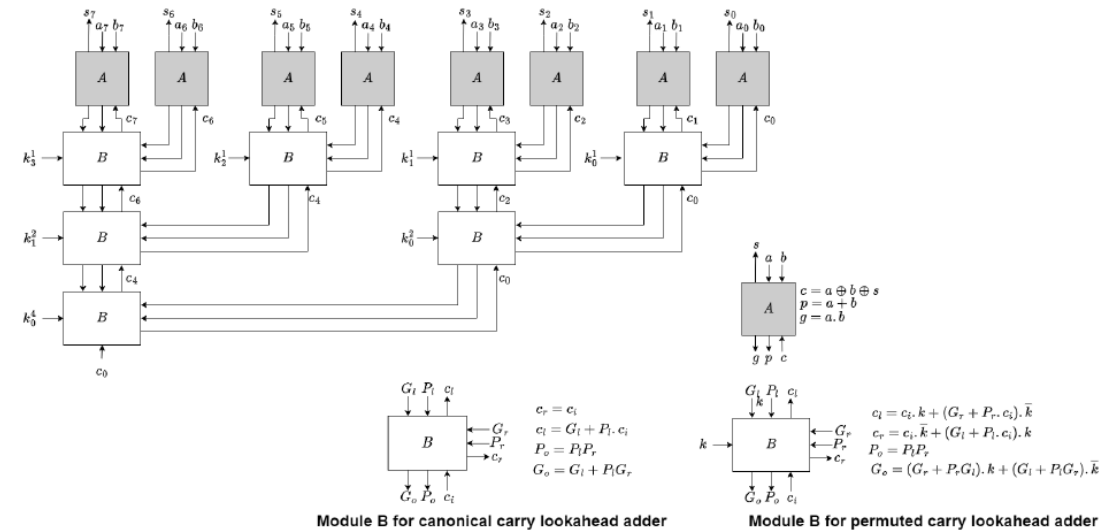
Bitwise Boolean: nothing to do

- AND, OR, XOR, NOT all commute with $P\alpha$

Addition: the carry breaks block locality

- Kogge-Stone tree matches the permutation tree
- Generate and propagate are swap-invariant
- Only the carry direction flips, one XOR per level
- Same adder for SUB, SLT, SLTU, BEQ, BNE

+67 % area, one extra gate of depth



Canonical (top) vs permuted (bottom) carry-look-ahead adder

Rebuilding the ALU: shifts and multiplication

One shift stage, three trees in cascade

- Block tree: displaces the blocks
- Key tree: computes the resulting key K_D
- Black tree: repairs, swap iff $\text{Key} \oplus K_D = 1$
- Total depth $\log_2 n$, as a canonical shifter

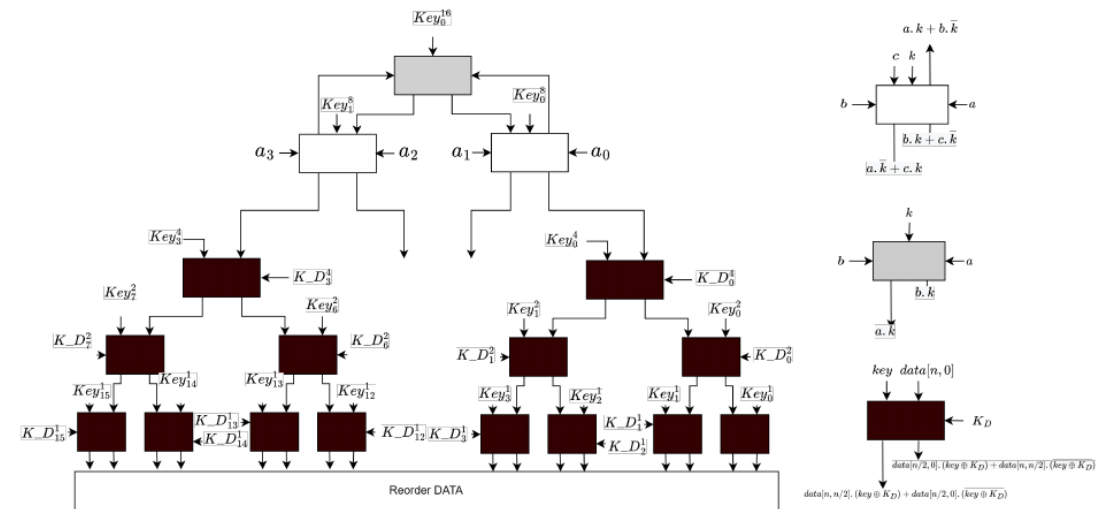
One circuit, four shift modes

- SRL: complementary key. SRA: MSBFind tree
- ROL/ROR: propagate looped back into carry

Multiplier: shift-and-add, mask instead of branch

- Constant time, n cycles, no timing leak

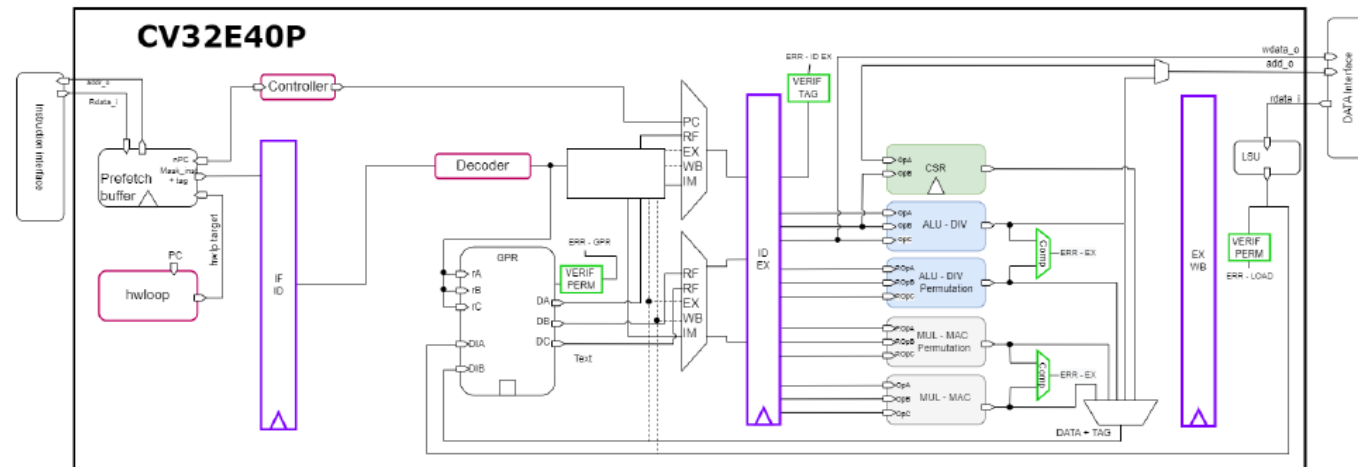
Shifter: +233 %, the main optimisation target



8-bit shift on 32-bit permuted data. block tree and black tree

Integration in a RISC-V CV32E40P

- In-order four-stage core, 22 nm FDSOI, 100 MHz
- Register file doubled, ALU duplicated, lock-step
- Five checkpoints: register-file ports, EXECUTE, LSU
- Mismatch raises a precise exception, recovery re-keys
- PoC: one canonical ALU, one permuted ALU
- Design target: two permuted ALUs, independent keys



CV32E40P pipeline with the duplicated permuted register file and ALU. The five checkpoints are the green boxes.

Refreshing the key: transmutation

A static key would eventually leak

Naive re-keying exposes the cleartext for one cycle

Look instead for α'' with $P_{\alpha''}(P_{\alpha}(x)) = P_{\alpha'}(x)$

- Exists and is unique: the family is a group
- Not $\alpha \oplus \alpha'$, the levels do not commute
- A generator walks the tree: XOR, then swap below
- The data path is traversed once

$$P_{\alpha''}(P_{\alpha}(x)) = P_{\alpha'}(x)$$

one combinational pass, no cleartext on any wire

Combinational, depth 5, on the key only: 3 101 GE

31 registers re-keyed in 31 cycles

About 0.05 % of cycles at a 2^{16} period

A KDF from the boot key preserves authenticity

Silicon cost: area and timing

189 600 GE, +74 % over the plain core

- 47 points: the duplicated register file
- 27 points: permuted ALU and checkpoints

Iso-frequency at 100 MHz, no retiming

- Critical path stays in the canonical multiplier
- Added depth bounded analytically, not measured

Lock-step: 100 % of the back-end, static layout

Module	Area (GE)	Overhead
Permutation P_α	2 013	—
Canonical CLA adder	1 806	—
Permuted CLA adder	3 015	+67 %
Canonical barrel shifter	4 976	—
Permuted barrel shifter	16 596	+233 %
Iterative permuted multiplier	10 440	32 cycles
Transpermutation key generator	3 101	—

Gate equivalents with NAND2 = 4 GE. The multiplier and the key generator are synthesised but not integrated in the test chip.

Multi-beam coverage: the analytical argument

k spots split as k_0 on α_0 and k_1 on α_1

- Undetected only if both bit sets coincide
- $k_0 \neq k_1$ is always detected, so every odd k

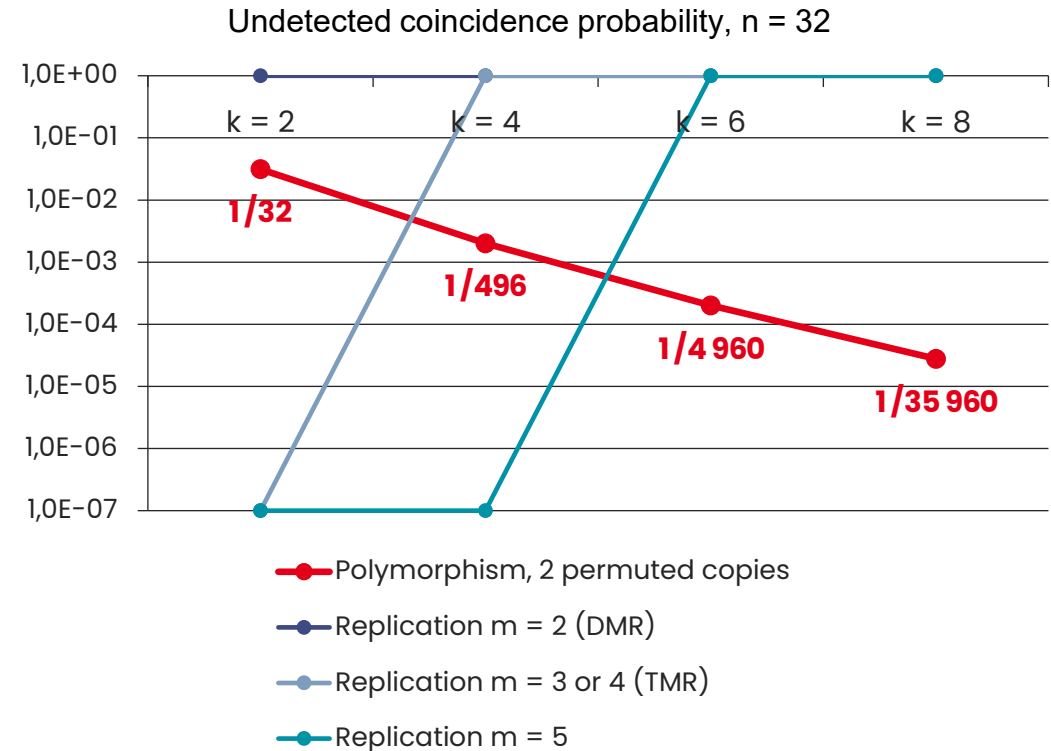
Re-keying keeps the attacker key-blind

Only escapes: all-zeros and all-ones

$$\Pr[\text{undetected}] \leq \frac{1}{\binom{n}{k/2}}$$

Worst case $k_0 = k_1 = k/2$, with $n = 32$ bits

**Replication is a cliff: certain escape once $k \geq m$.
Polymorphism is a slope that keeps falling.**



Logarithmic scale, worst case $k_0 = k_1 = k/2$. With m replicas the layout is static and known, so the same fault is reproduced on every copy. the escape is certain as soon as $k \geq m$. Certain detection is an exact zero and is drawn on the axis floor.

Single-beam laser campaign on the die

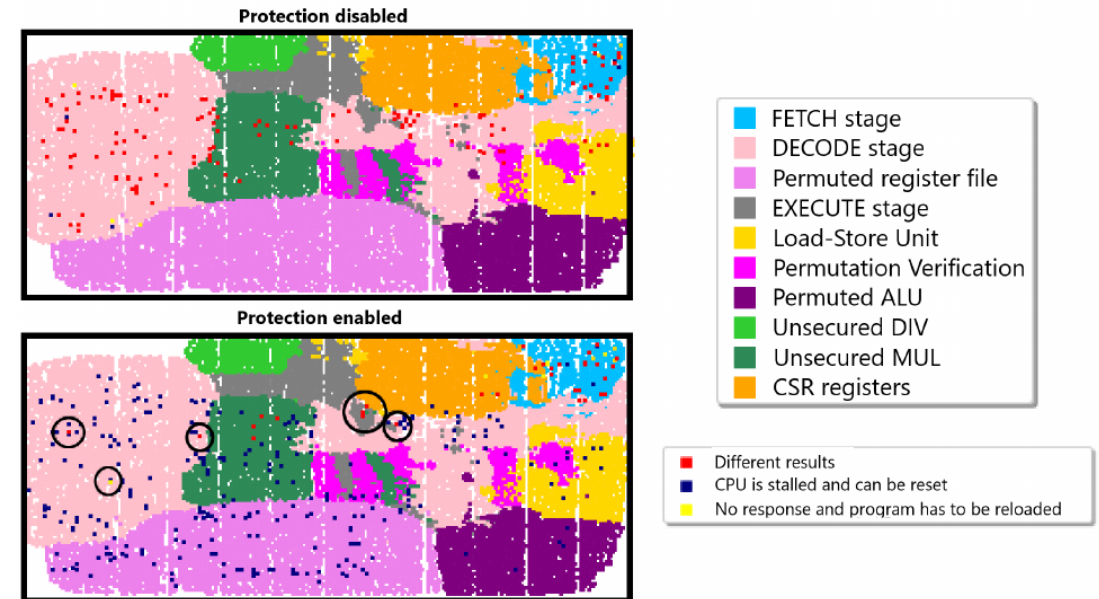
Back side, 975 nm, 40 ns, die thinned to 50 μm

- 10 μm grid, one pulse every 100 instructions
- Narrow window in FDSOI: 290 to 340 mW

Exploitable faults: 131 $\rightarrow$ 22

- Detected and stopped: 5 $\rightarrow$ 195
- 21 of the 22 residuals outside the protected path
- Detections cluster where the protection lives

A campaign shows faults, it cannot prove their absence



Injection map, countermeasure off (top) and on (bottom). Red: exploitable. Blue: detected and stopped.

Perspectives

Confidentiality: masking composes, combined gadgets still needed

Authenticity: session keys derived from a boot key

Leakage rate: measure it, then fix the re-keying period

Out-of-order: two permuted ALUs per reservation station

Vectors and floats: SIMD lanes, mantissa and exponent

Conclusion

- Beyond a few spots, randomise the layout instead of adding replicas
- A 31-bit keyed permutation, built as a tree of transpositions
- Homomorphic ALU: no extra cycle, no longer critical path
- 74 % area at iso-frequency, silicon in 22 nm FDSOI
- Exploitable faults 131 → 22 on the laser bench

Next: a multi-beam bench, and two permuted ALUs

Polymorphism

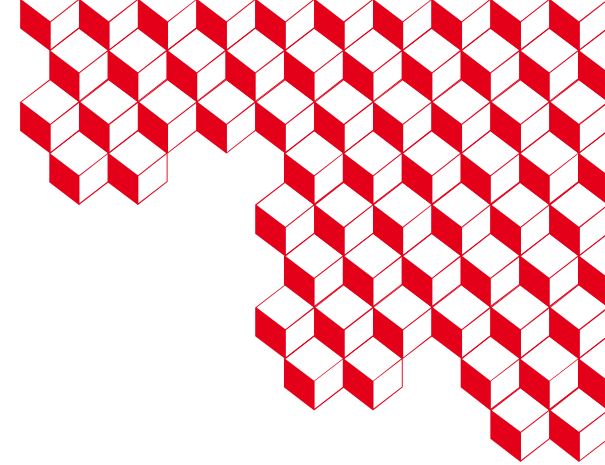
instead of a fifth replica

Homomorphic ALU

add, shift, multiply, all keyed

Silicon-proven

22 nm FDSOI, laser bench



Thank you

Questions?

CEA-Leti

17 avenue des Martyrs

38054 Grenoble Cedex 9, France

olivier.savry@cea.fr

Backup: inside the permutation tree

Each node is a Fredkin gate

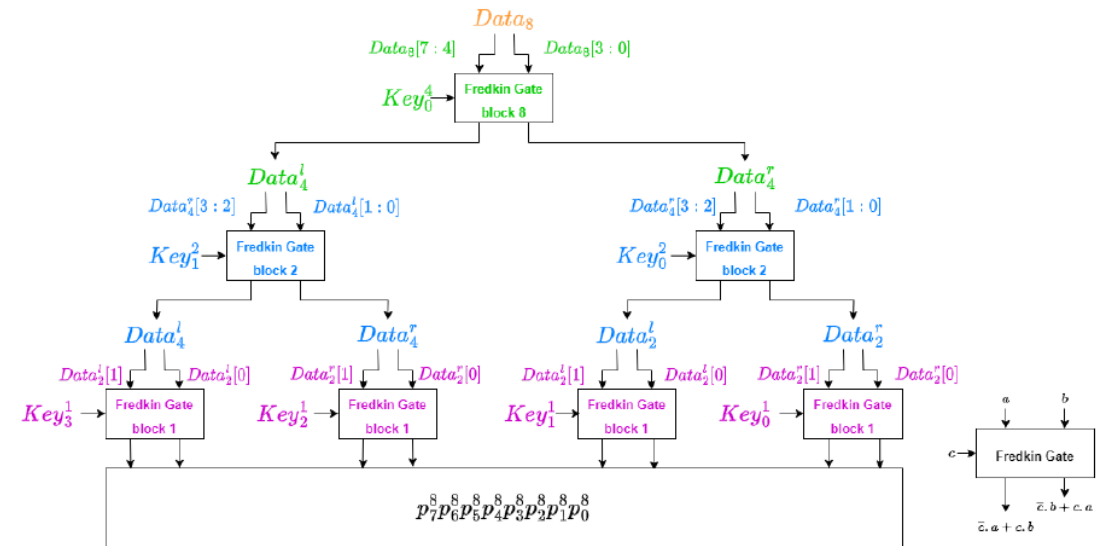
- $c = 0$ keeps (a, b) , $c = 1$ swaps to (b, a)
- Level i exchanges pairs of 2^i -bit blocks

Involutive gates, non-involutive composition

- Levels do not commute
- Depermutation: same topology, reverse order

Automorphisms of the binary tree on 32 leaves

- A Sylow-2 subgroup of S_{32} , order 2^{31}



The same tree on an 8-bit word, three levels of Fredkin gates

Backup: campaign breakdown by unit



Unit	Exploit. OFF	Detect. OFF	Exploit. ON	Detect. ON
FETCH (instructions, unprotected)	21	2	12	—
DECODE (control, register-file read)	90	3	3	82
Permuted register file	—	—	—	63
EXECUTE / permuted ALU	2	—	1	13
Load-store unit	4	—	—	10
Multiplier (unprotected in the PoC)	13	—	4	25
CSR (unprotected)	1	—	2	2
Total	131	5	22	195

Exploitable = faulty AES output, the precondition for a differential fault analysis. The two configurations are independent single-pass scans of the same die with the same number of injections, so counts are aggregate, not point-paired.

Backup: the cost of pure replication

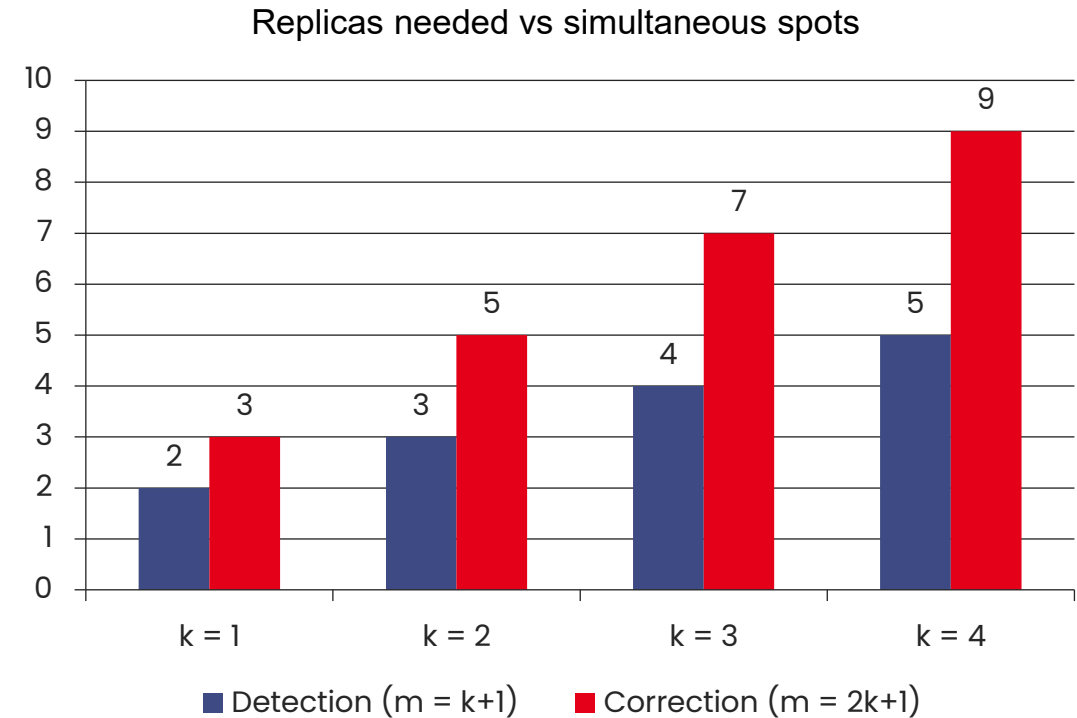
Under k localised spots, no coherent propagation

- Detection: $m \geq k + 1$, one copy must stay clean
- Correction by voting: $m \geq 2k + 1$

At $k = 4$, detection alone needs five replicas

- Two beyond the dynamic TMR of recent cores
- Klessydra-dfT03, SentryCore: 100 to 200 % area
- Area, synchronisation and voting grow super-linearly

The permuted data path costs 74 %, threshold fixed



Number of replicas required, by attack order