Test and Harware Security

Marion Doulcier, Giorgio Di Natale, Marie-Lise Flottes, Bruno Rouzeyre

LIRMM

Test & Security

G. DiNatale, M. Doulcier, M-L. Flottes, B. Rouzeyre

Pastis 2008

Test & Security : the dilemma

LIRMM

- Circuit testing is mandatory to guarantee a good security level

  A hardware defect may induce some security vulnerability

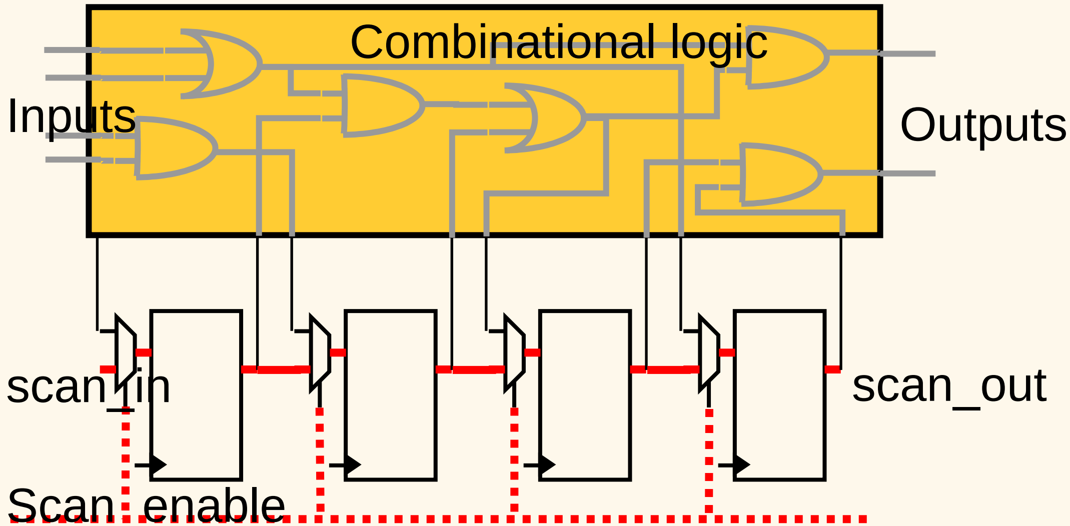
- But

	Test	Security
Observability	↗	↘
Controlability	↗	↘

Testing techniques (1)

LIRMM

- External Test + Scan path



✓ High fault coverage
✓ Automatic generation of scan chains
✓ Easy test sequence generation

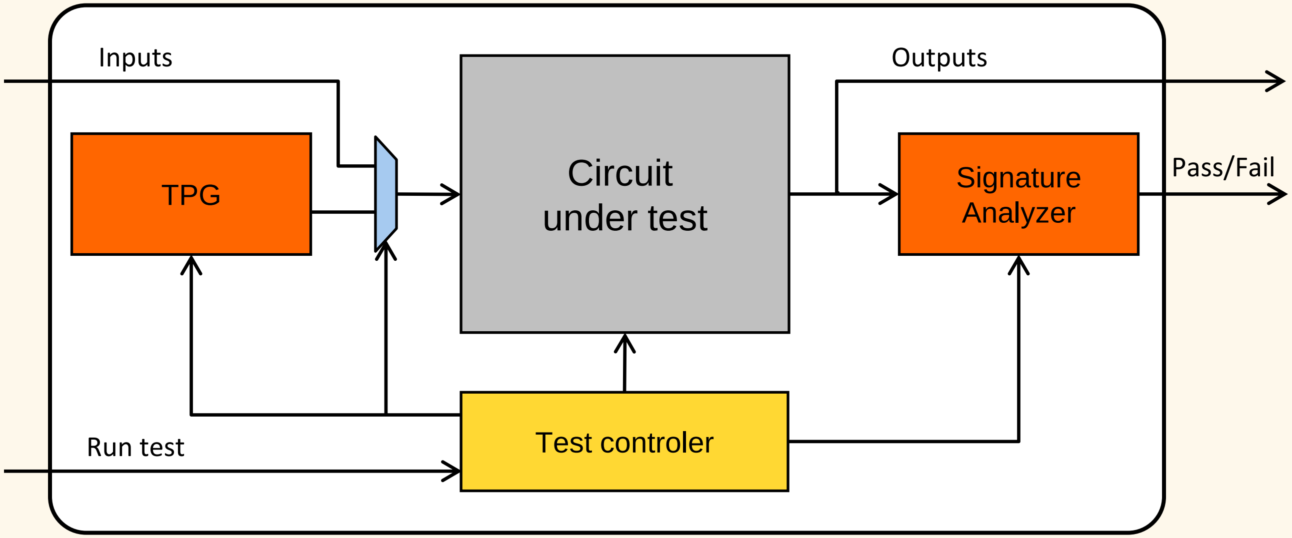
Vulnerability

- Control and observation of internal states of CUT
- => secret data retrieval

Testing techniques (2)

LIRMM

- Built-in Self Test (BIST)



✓ No control/observation from the outside

✓ Area overhead

✓ Fault coverage (pseudo-random testing) ?

Securing the scan chain

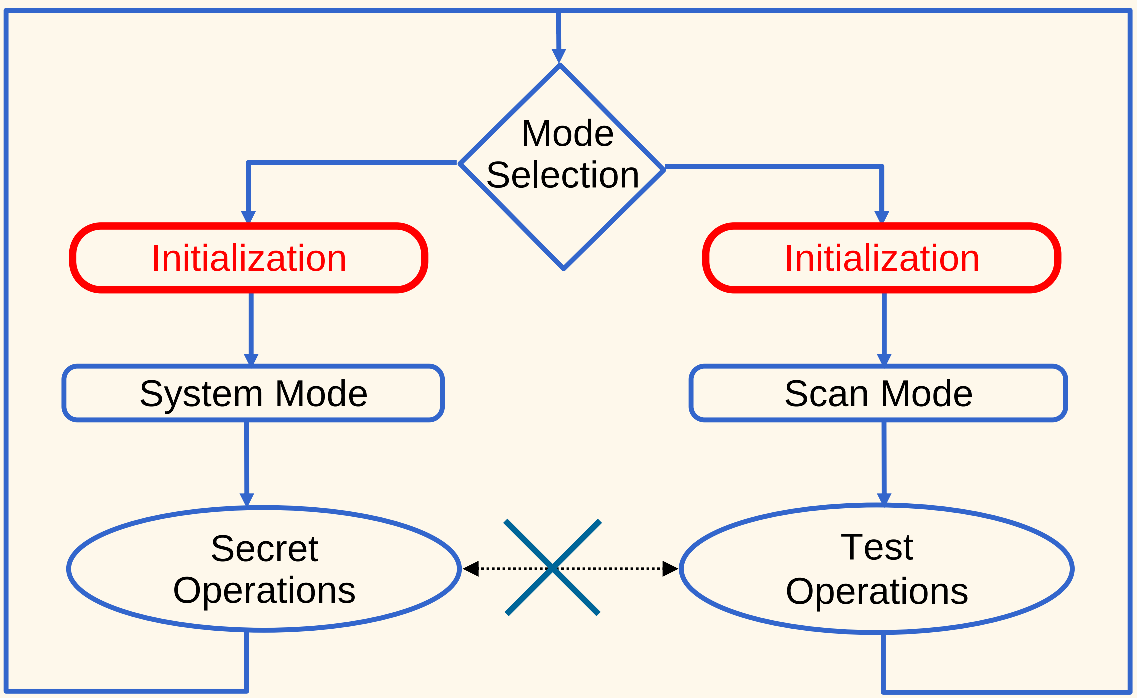
- Goal
  - ✓ No observation or control of the functional data processed by the secure system
- Principle
  - ✓ Prevent illegal scan shift operations
- Solutions
  - ✓ Test mode protection
    - Scan protocol
    - Test Patterns watermarking

    } protection against illegal usage of the test mode
  - ✓ System mode protection
    - Scan chain scrambling
    - Scan enable tree protection
    - Spy FFs

    } protection against scan chain probing attacks

Test mode protection

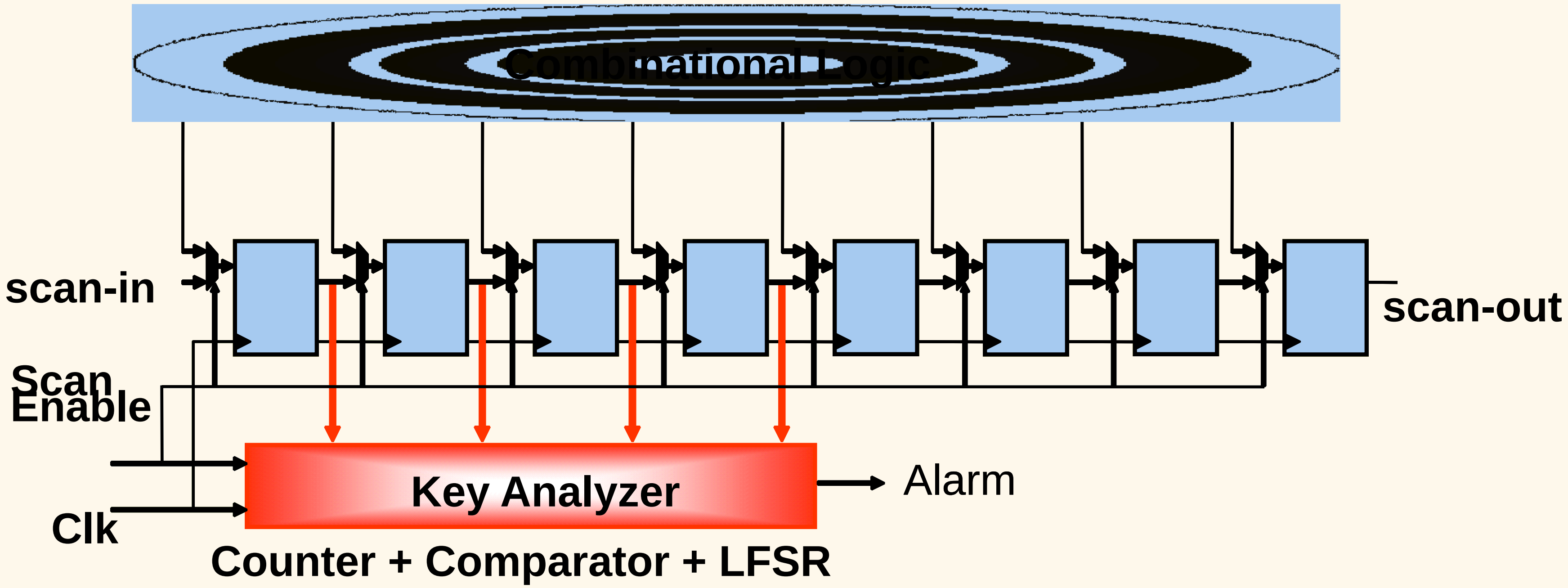
- Scan protocol
  - ✓ The circuit is initialized before and after test mode
  - ✓ Initialization is checked before switching to another mode
  - ✓ Switch between the 2 modes, bypassing the initialization, is detected



Test mode protection

LIRMM

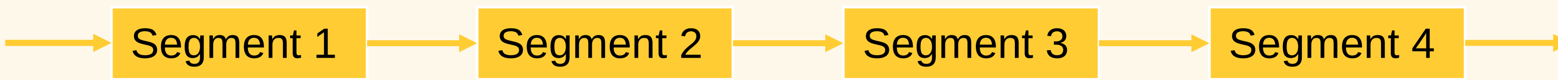
- Test pattern watermarking
  - ✓ Test patterns embed authentication keys
  - ✓ Keys are dynamically changed (e.g. LFSR-based)



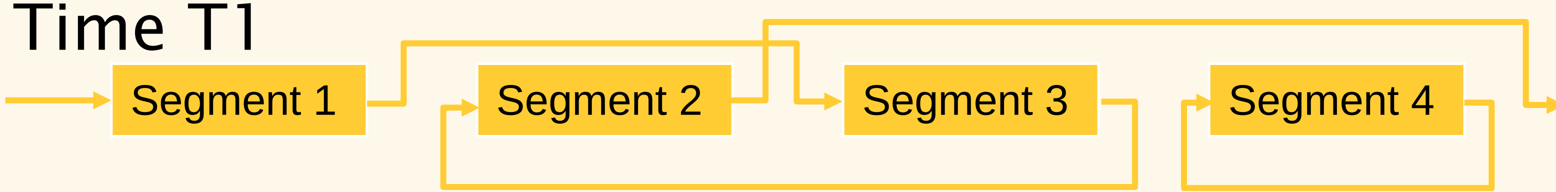
System mode protection

LIRMM

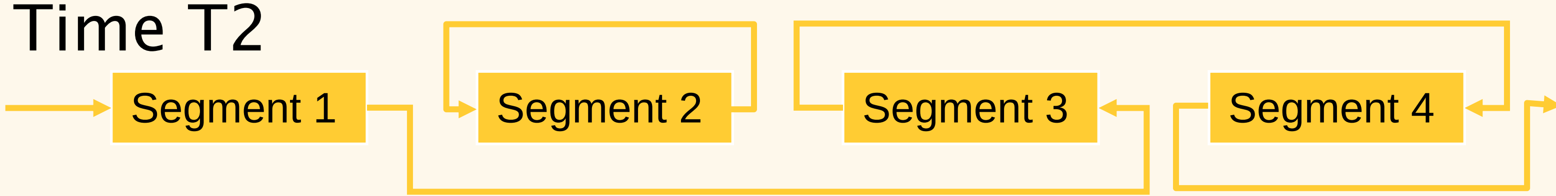
- Scrambling method
  - ✓ Scan path with a prefixed segment organization during test mode



  - ✓ Scan path with random segment organization if shift during system mode
    - Time T1

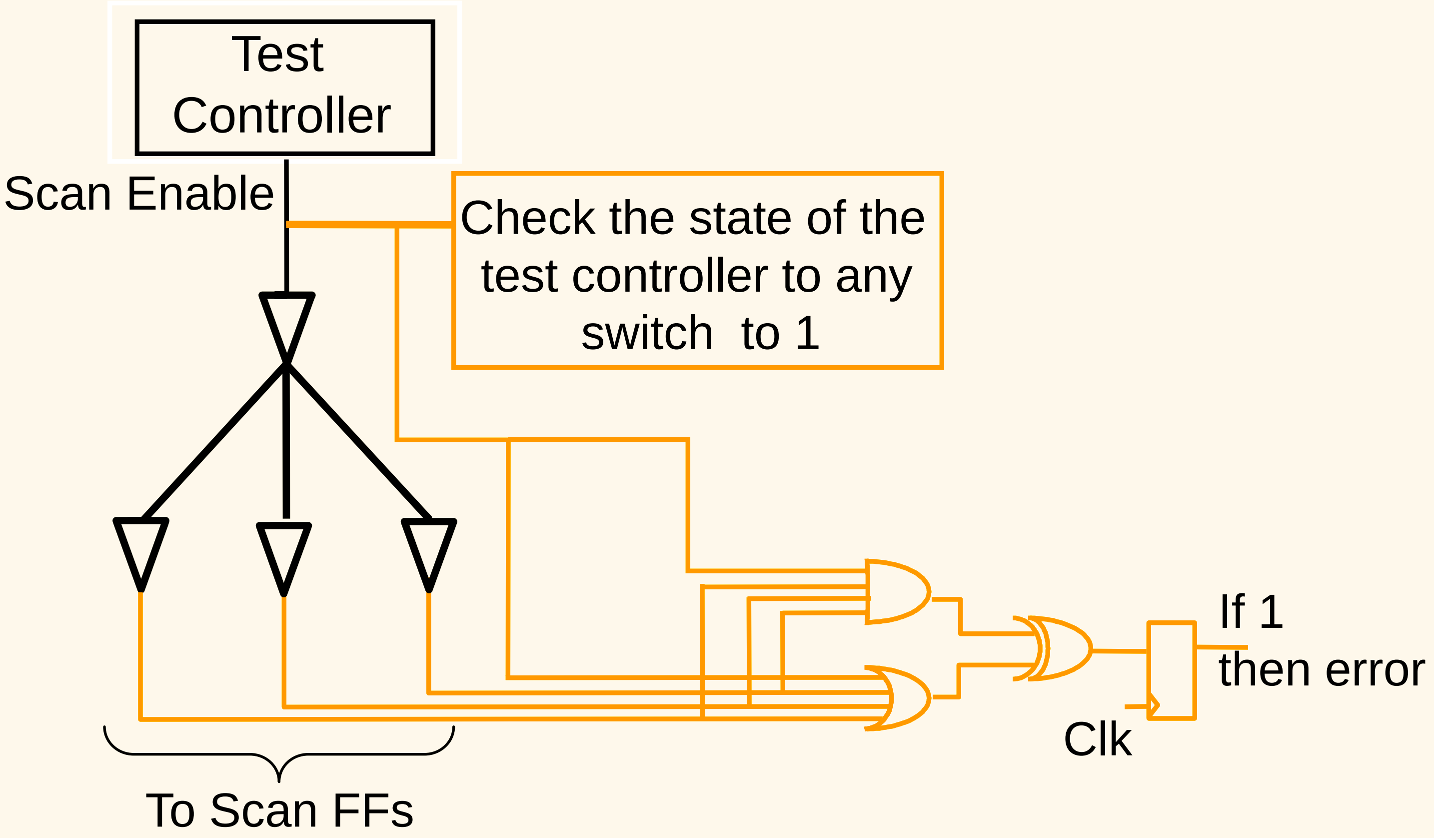


    - Time T2



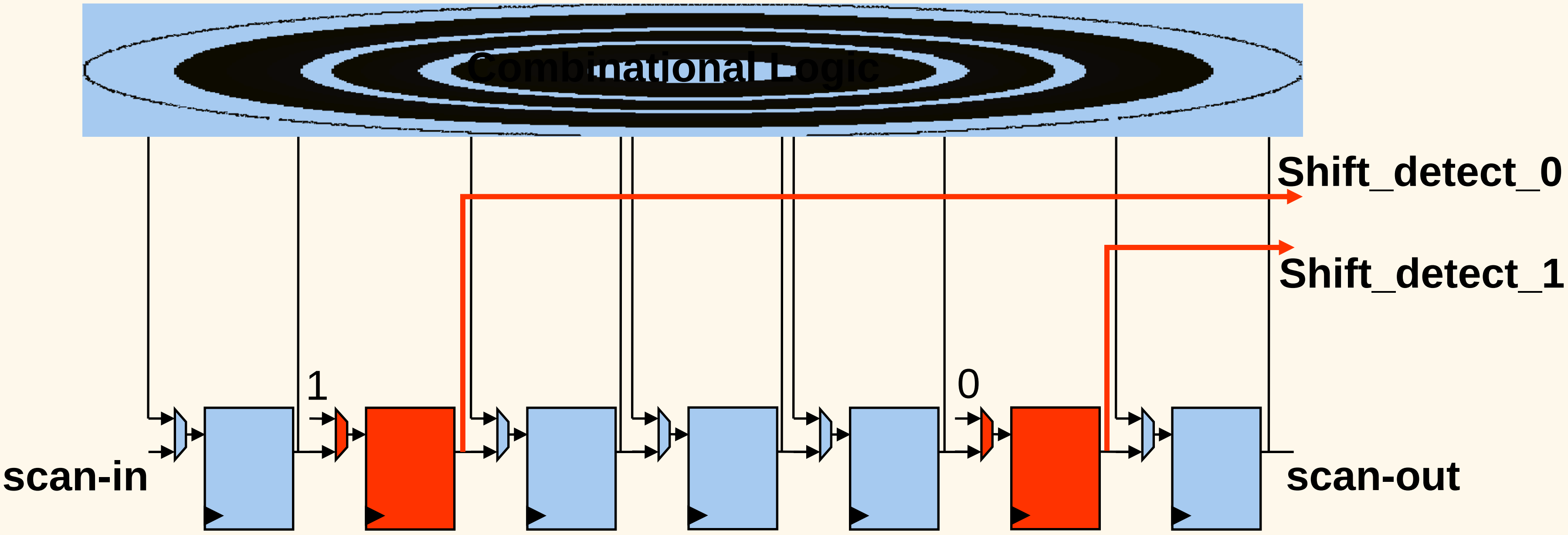
System mode protection

- Scan-Enable Tree Protection
  - ✓ Compare the scan enable signals at different locations



System mode protection

- Spy Flip-Flops
  - ✓ Include Spy cells in the scan chain
  - ✓ Control the spy cells to a constant value
  - ✓ Observe the spy cells states



Experimental results

		Scrambling	Scan enable	Spy cell	Pattern watermarking
Insertion flow		RTL	RTL + place&route	RTL	RTL
Test	Test time	0%	1%	5%	0.4%
Design	Area	0.2%	0.3%	1.8%	~0%
	power c.	7%	0%	0%	0%
Security		+++	++	++	+

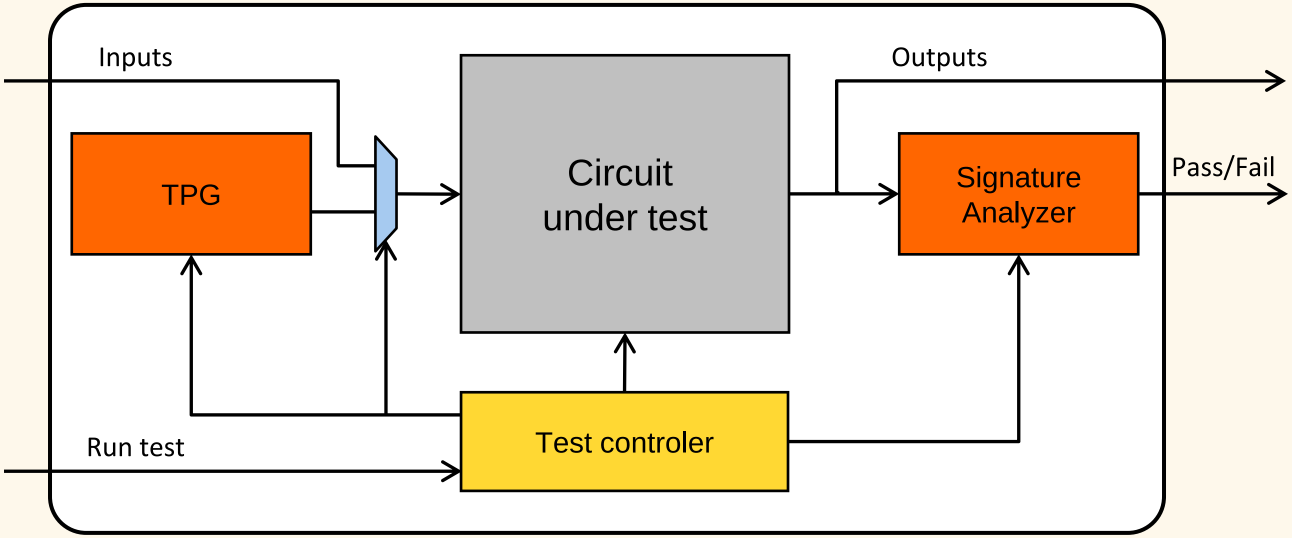
To resume

LIRMM

- Countermeasures address two kinds of attack
  - ✓ Legal activation of the test circuitry
    - corruption of the authentication scheme
    - malfunction of the security
    - insider attack
  - ✓ Physical access to the chip
    - high knowledge of the circuit
    - very expensive equipment

BIST

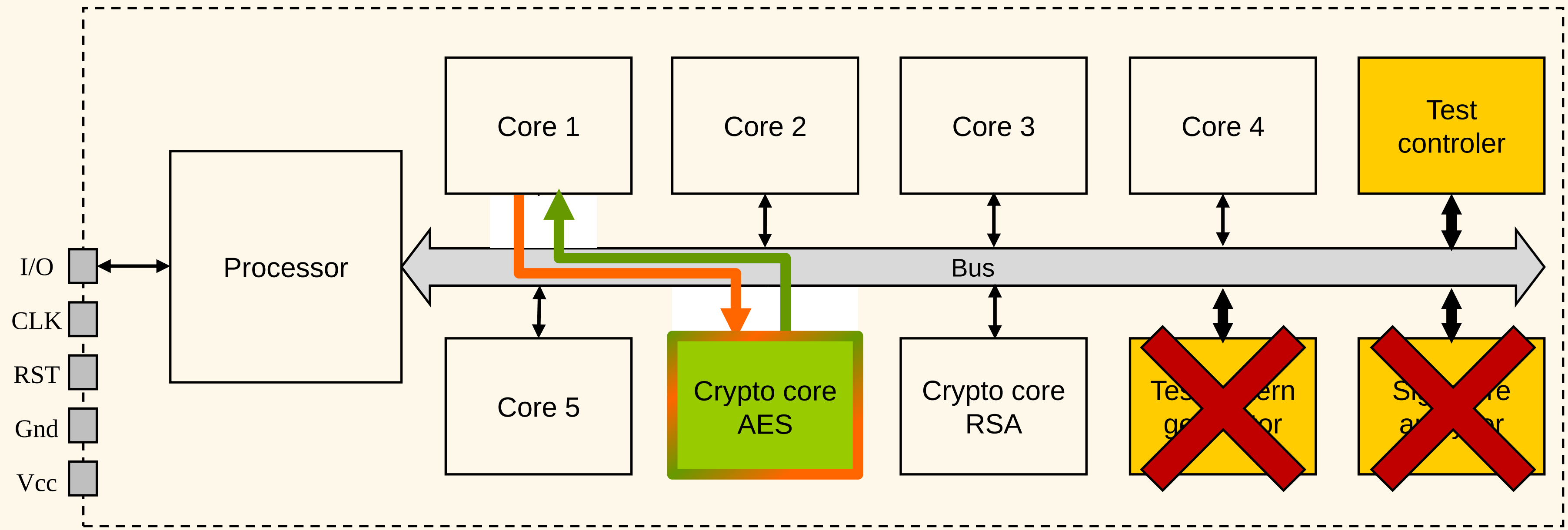
- BIST
  - ✓ Reduced ATE cost
  - ✓ In-situ testing
  - ✓ Reduced external access



- But
  - ✓ Circuitry overhead

Proposal

LIRMM

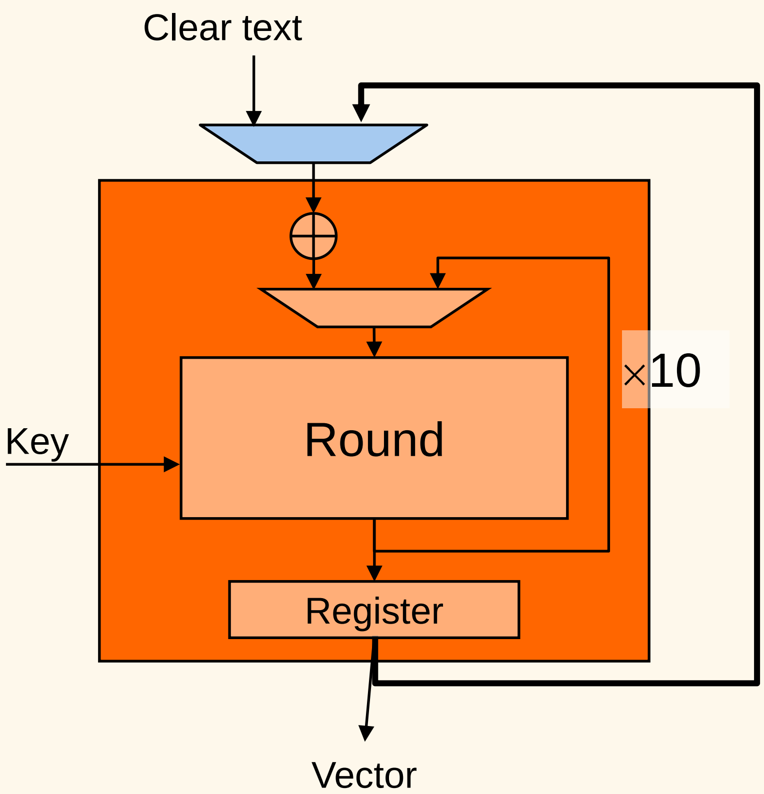


- Self-test of crypto-core
- Use the crypto-core as a test ressource (TPG/SA)
- AES/DES

"Randomness" of cipher

LIRMM

- 1 vector per encryption

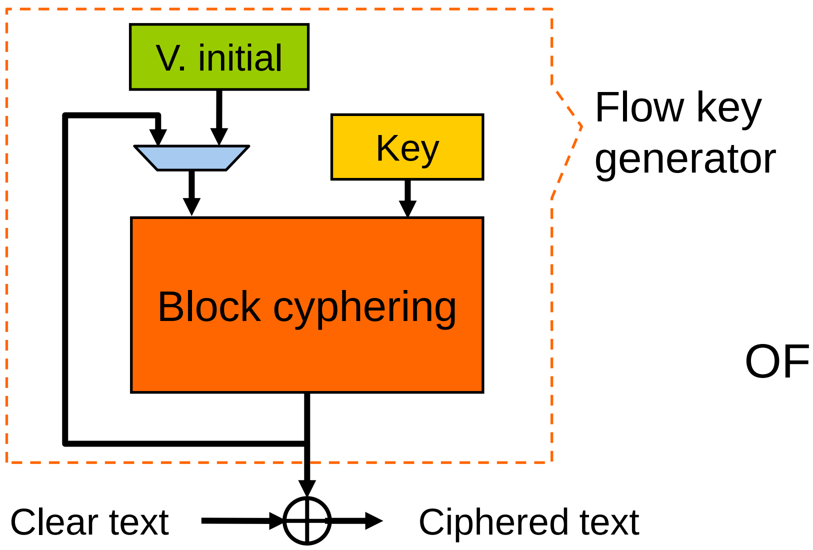


≈ 1 vector every 10 clock cycles

"Randomness" of cipher

LIRMM

- 1 vector per encryption

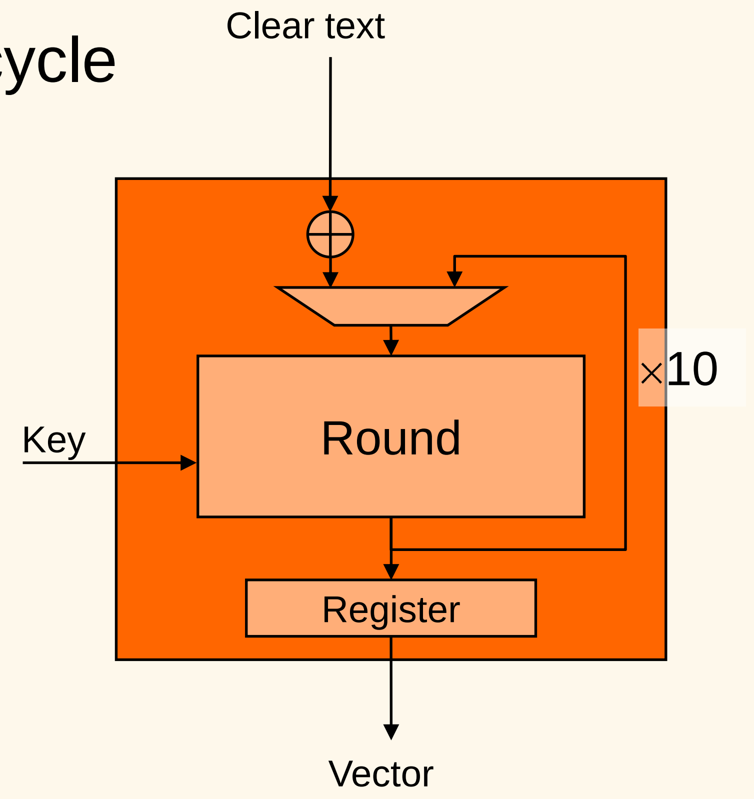


OFB (Output FeedBack)

≈ 1 vector every 10 clock cycles

"Randomness" of cipher

- 1 vector per round cycle

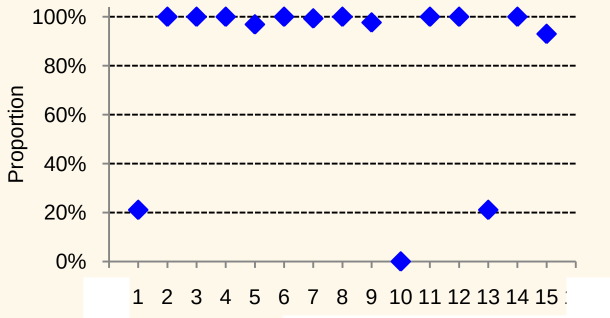


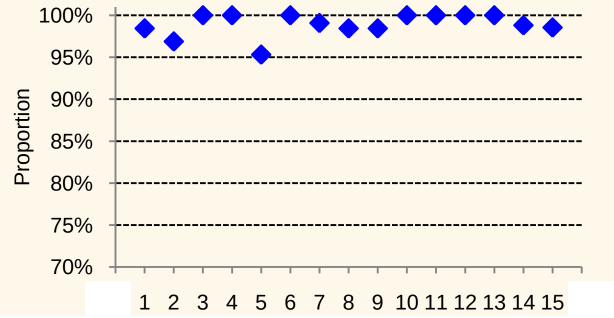
"Randomness" ? (Diffusion, Confusion, Bijection)

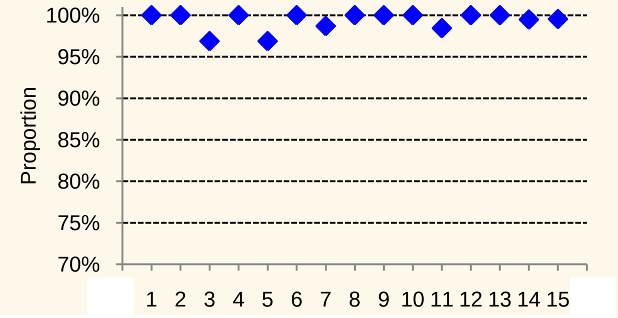
Checked by NIST statistical package suite (15 randomness tests)

Randomness comparison

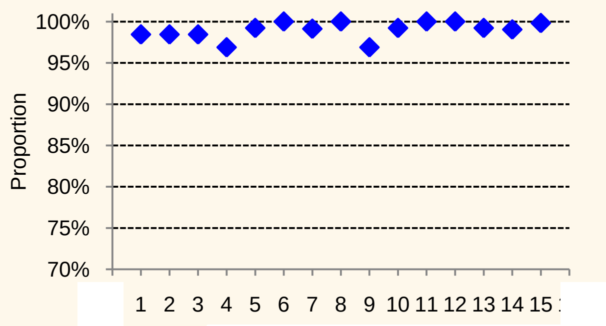
LIRMM

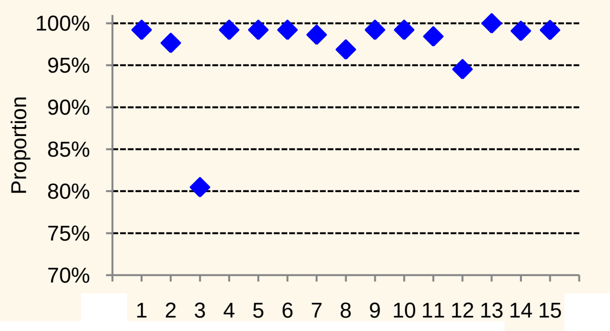






LFSR DES DES round





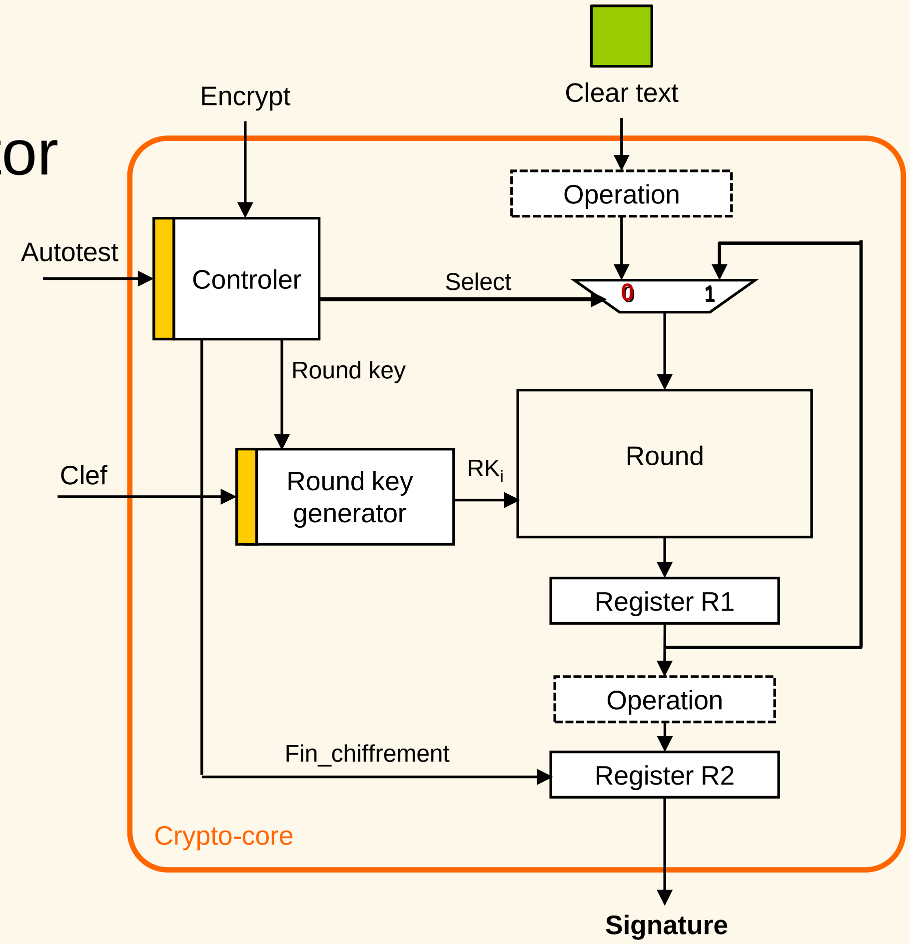
AES AES round

AES round / DES round : as good random pattern generators as LFSRs

Self-test Procedure

LIRMM

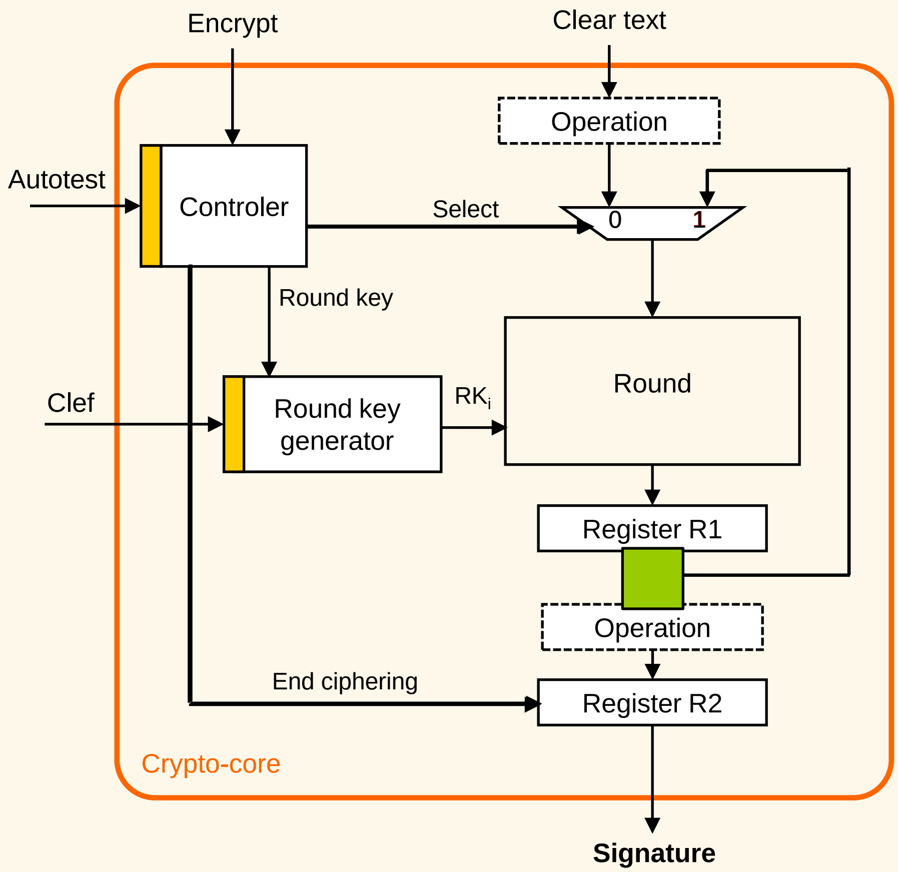
- Looped Crypto-core ⇔ random number generator
- First step
  - ✓ 1^{st} cycle



Self-test Procedure

LIRMM

- Second step
  - ✓ Cycles 2, 3, ……, n



Self-Test

LIRMM

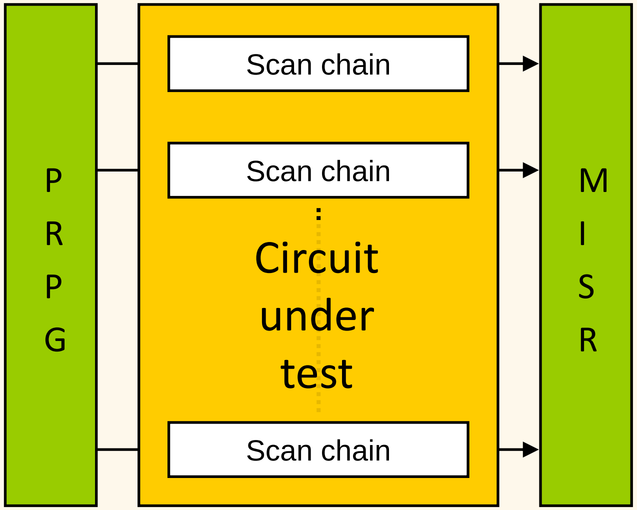
- Theorical result
  - AES : Fault-coverage = 100% after $n \in \{2520,...,2590\}$ clock cycles
  - DES : Fault-coverage = 100% after $n \in \{620,...,710\}$ clock cycles

- In practice
  - AES
    - Fault-coverage = 100% after 2200 clock cycles (∀key, ∀ clear text)
  - DES
    - Fault-coverage = 100% after 560 clock cycles (∀key (not wk), ∀clear text)

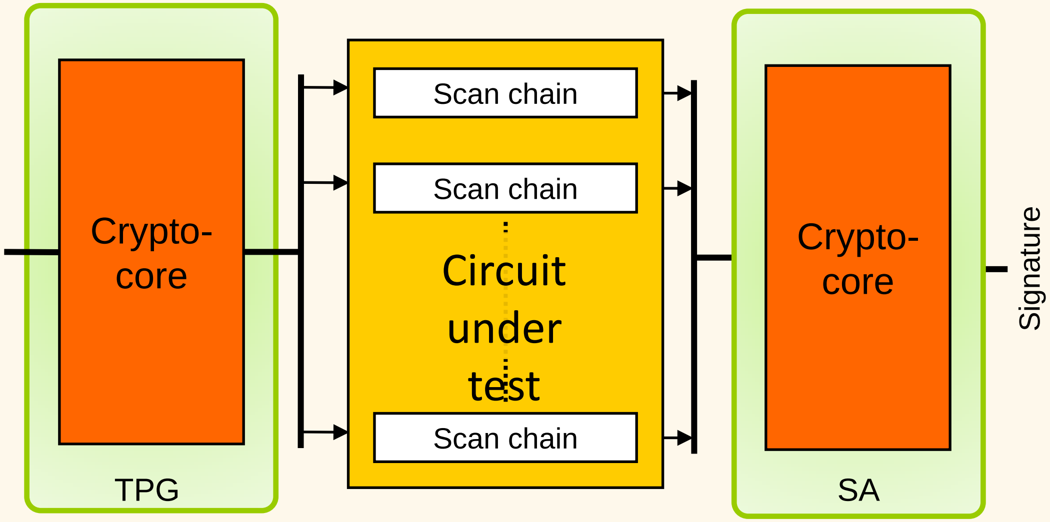
Crypto-core as TPG/SA

LIRMM

- STUMPS Architecture



- Proposed solution



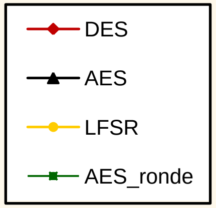
TPG for other cores

Test response compactor for other cores

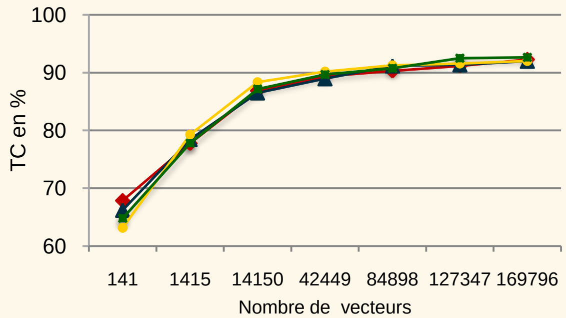
TPG : ISCAS'89 benchmarks

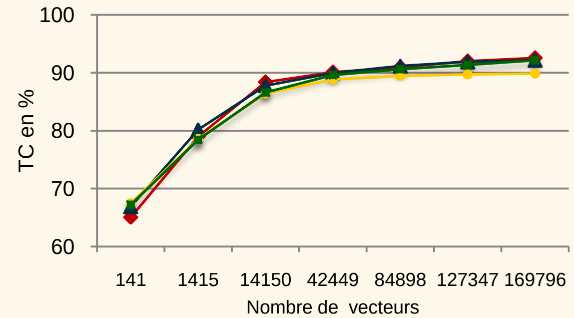
LIRMM

1 Scan chain | 64 Scan chains

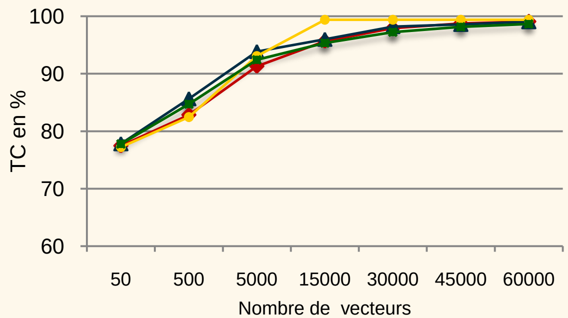


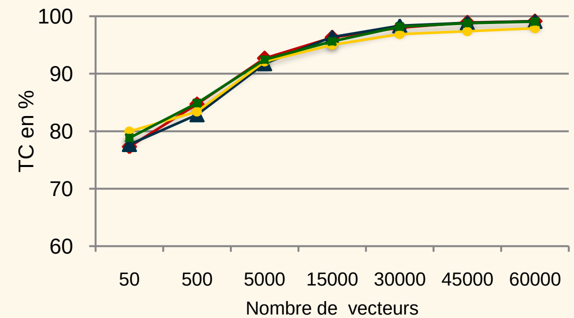
- s9234



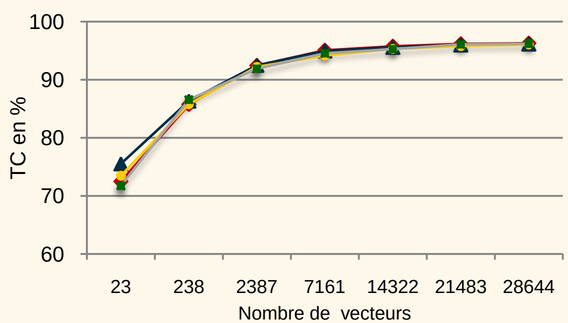


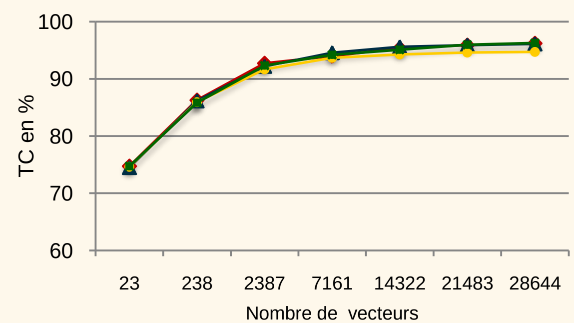
- s13207





- s38548

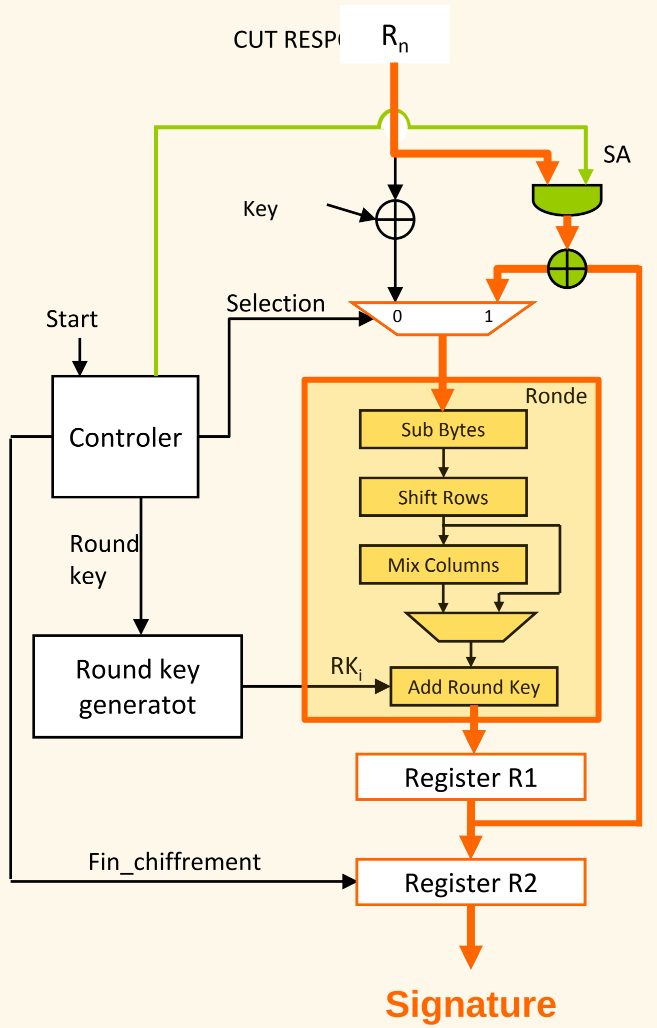




Response compaction

- Response compaction mode :
  - ✓ SA = Selection = 1

- Functional mode
  - ✓ SA=0



Fault-masking probability

- AES/DES

$$P(M_n) = \frac{1}{2^m} - \left(\frac{1}{2^m}\right)^n \qquad P(M_{128}) \xrightarrow{n\to\infty} \frac{1}{2^{128}} \approx 10^{-40}$$

- MISR

$$P(M_n) = \frac{2^{n-1} - 1}{2^{m+n-1} - 1} \qquad P(M_{128}) \xrightarrow{n\to\infty} \frac{1}{2^{128}} \approx 10^{-40}$$

n = #test responses and m = 128 or 64

To resume

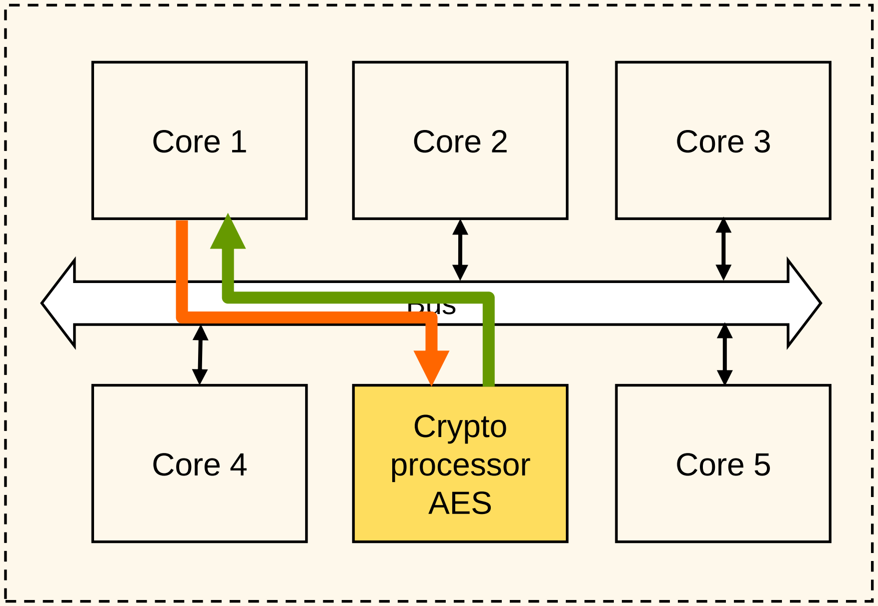
Cryto-core (AES/DES) as a test ressource:

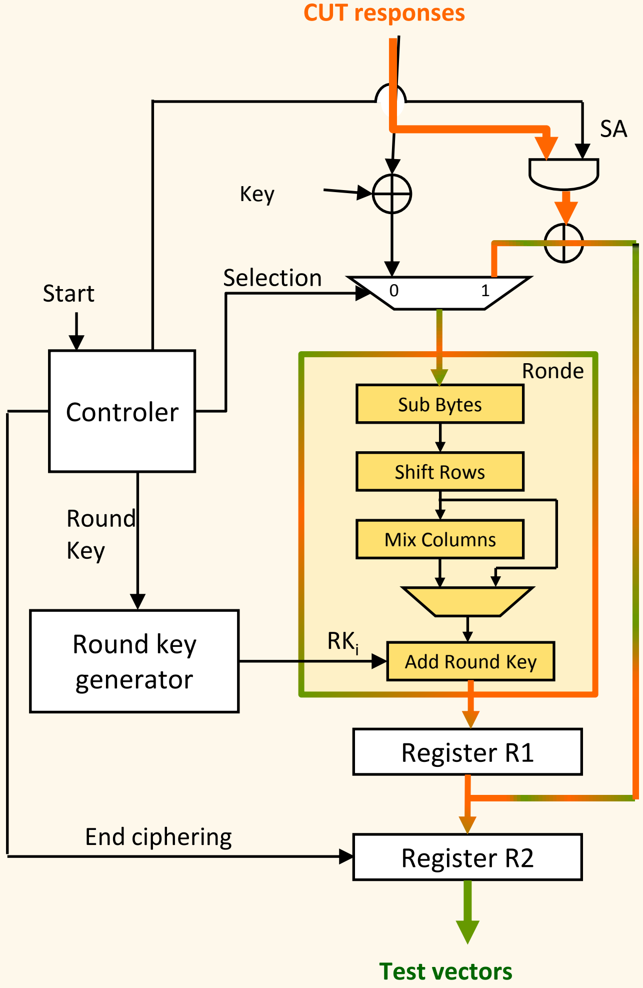
- ☺ Test Fault Coverage: ≅ LFSR
- ☺ Error Masking Probability: ≅ MISR
- ☺ Reduced area overhead
- ☺ No impact on ciphering frequency/latency

- ☹ Potential attacks (2 succesive round results observable))

  ⇒ use a specifc key for test ☺

Simultaneous TPG and Compaction





Area overhead

		AES	AES generator	AES compactor	AES Self-test	AES 4 modes
Round	-SubBytes	803 734	.	.	.	.
	-ShiftRows	0	.	.	.	.
	-MixColumns	59 847	.	.	.	.
	- AddRoundKey	49 945	.	.	.	.
Controler		6 345	+ 5.72%	+ 8.72%	+ 6.58%	+ 9.58%
Key generator		301 162	+ 0.015%	+ 0.015%	+ 0.015%	+ 0.015%
Glue logic		153 620	+ 0.04%	+ 17.95%	+ 0.04%	+ 18.36%
TOTAL		**1 374 655**	**+0.03%**	**+2.05%**	**+0.04%**	**+2.10%**

Overhead 2.1%

Synthesis: VHDL + Synopsys Design Compiler
Technology: 0.35 um CMOS librairies (AMS)

Area overhead

		AES	AES generator	AES compactor	AES Self-test	AES 4 modes
Round	-SubBytes	803 734	.	.	.	.
	-ShiftRows	0	.	.	.	.
Contro						58%
Key ge						15%
Glue l						36%
TOTAL		**1 374 655**	**+0.03%**	**+2.05%**	**+0.04%**	**+2.10%**

For comparaison :

Implementing a LFSR ⇒ 3.67%

Implementing a BILBO ⇒ 7.64%

Overhead 2.1%

Synthesis: VHDL + Synopsys Design Compiler
Technology: 0.35 um CMOS librairies (AMS)

Conclusion

Special attention must be paid when testing secure circuits

- ✓ Scan-based designs
  - Counter-measures
- ✓ Bist (random test)
  - Self-test
  - Test resource
  - ECC ?

Publications

- SCAN
  - ✓ [Jetta 07] ***"Securing Scan Control in Crypto Chips" .*** Journal of Electronic Testing and Applications **23**, 5 (2007) 457-464
  - ✓ [IOLTS'06] ***"Secure Scan Techniques: a Comparison"*** 12th International On-Line Testing
  - ✓ [DATE'06] ***"Secure Scan Design"*** Design, Automation and Test in Europe, 2006
  - ✓ [ETS'05] ***"Test Control for Secure Scan Designs"*** European Test Symposium, 2005
  - ✓ [IOLTS'04] ***"Scan Design and Secure Chip"*** On-Line Testing Symposium, 2004
- BIST
  - ✓ [LATW'07] : ***"AES vs LFSR Based Test Pattern Generation: A Comparative Study".*** *LATW'07: 8th IEEE Latin*-American Test Workshop, Cuzco, Peru, 2007.
  - ✓ [DELTA'08] : ***"AES-based BIST: Self-test, TestPattern Generation and Signature Analysis"***, DELTA'08: 4th IEEE International Symposium on Electronic Design, Test & Applications, 2008.
  - ✓ [WDSN'08] : "***Low Cost Self- Test of Crypto-Devices***", *WDSN'08: 2nd Workshop on Dependable and* Secure Nanocomputing, 2008.
  - ✓ [TVLSI] : "***Self-Test, Techniques for Crypto-Devices***", TVLSI: IEEE Transactions on Very Large Scale Integration Systems, to appear.

References

- **[Yan04]:** B. Yang, K. Wu, R. Karri, Polytechnic University, "Scan-based Side-Channel Attack on Dedicated Hardware Implementations on Data Encryption Standard", International Test Conference (ITC 2004), Charlottes, USA, October 26-28, pp 339-344
- **[Yan05]:** B. Yang, K. Wu and R. Karri, Secure Scan: A Design-for-Test Architecture for Crypto Chips, Design Automation Conference (DAC 2005), Anaheim, July 12-14 pp 135-140, 2005
- **[Yan, FDTC 05]**: B. Yang & R. Karri, "Crypto BIST: A Built-In Self Test Architecture for Crypto Chips", 2nd Workshop on fault diagnosis and tolerance in cryptography (FDTC 2005), pp 95-108
- **[NIST 800-22]**: A statistical test suite for random and pseudorandom number generators for cryptographic applications NIST Special Publication 800-22 (with revisions dated May 15, 2001)