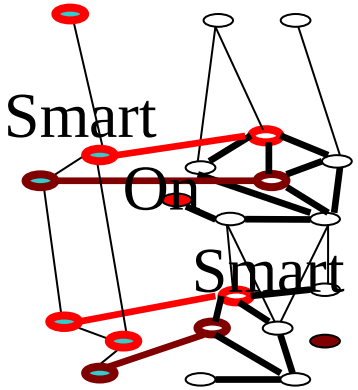
# **S**trategy **O**f **S**ecurity on **S**mart **O**n **S**mart




ANR-SESUR 2007
CEA / LIP6 / Trusted-Logic / Viaccess

P. Soquet, B. Robisson, M. Agoyan, G. Phan

P. Bazargan-Sabet, F. Wajsburt.

Workshop PASTIS, Gardanne
17/06/2010

# Schedule

- Presentation & context
- Strategy of security
- Prototyping
- Conclusion

# Introduction

➢The SOS project : a practical approach to manage the security on a tamper resistant device.

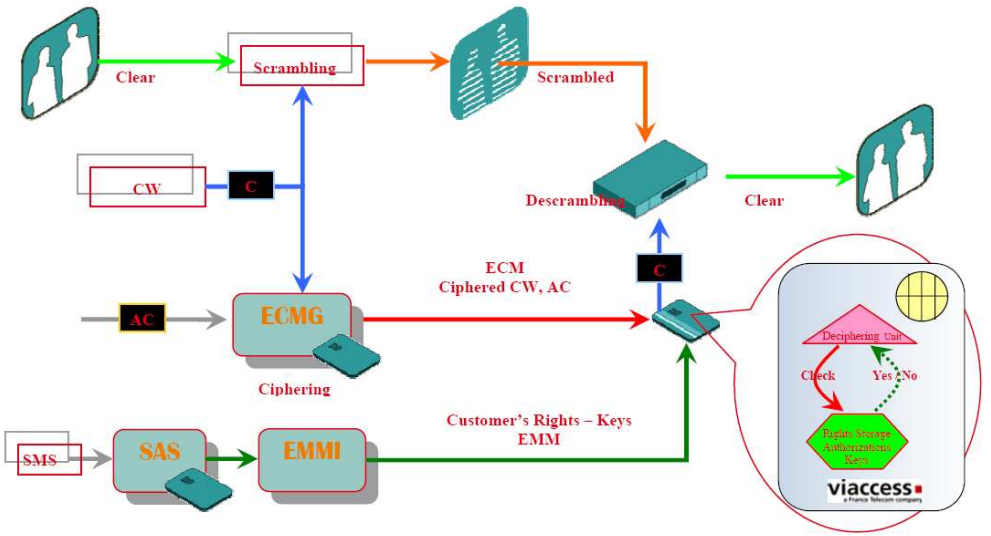
➢Focus on these four properties :

- Security
  - Protect your assets
- Availability
  - Don’t interrupt the service
- Performance
  - Go fast
- Adaptability
  - The system can be easily adapted against new threats

# Context

➢Target

**Conditionnel access** for paytv.



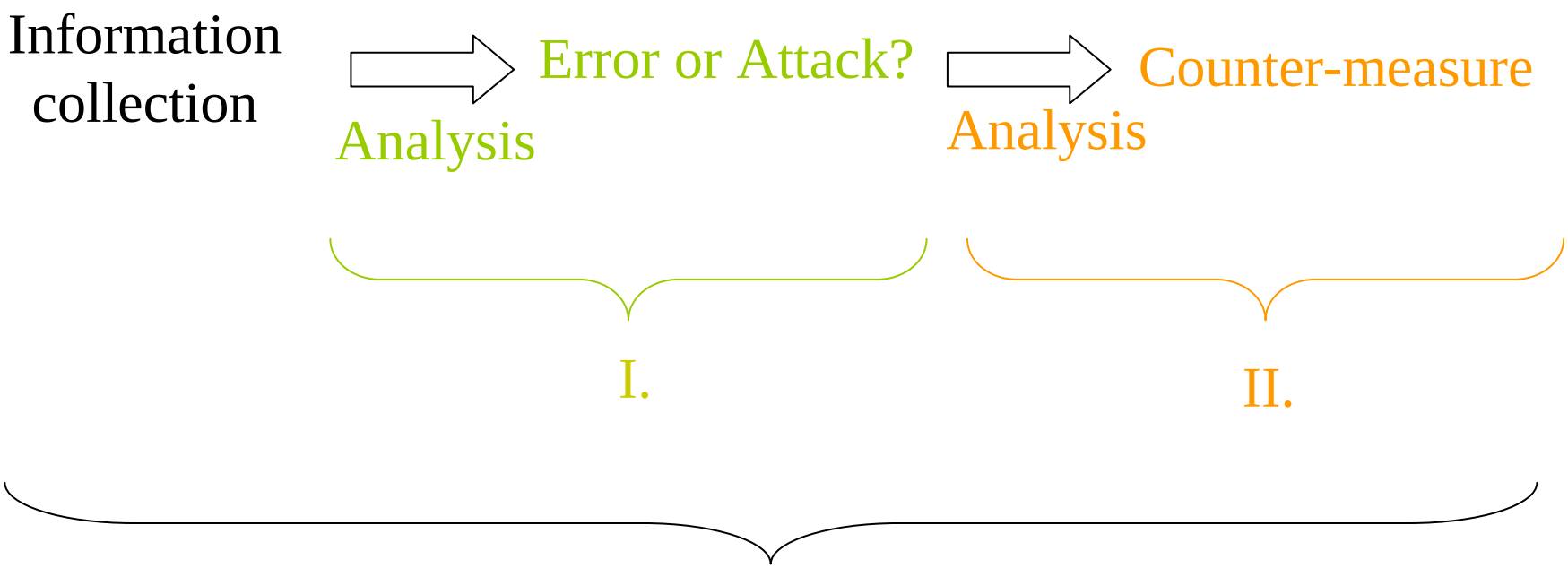

➢Needs

- High level of security
- Real time performance
- Reliability

➢Principle

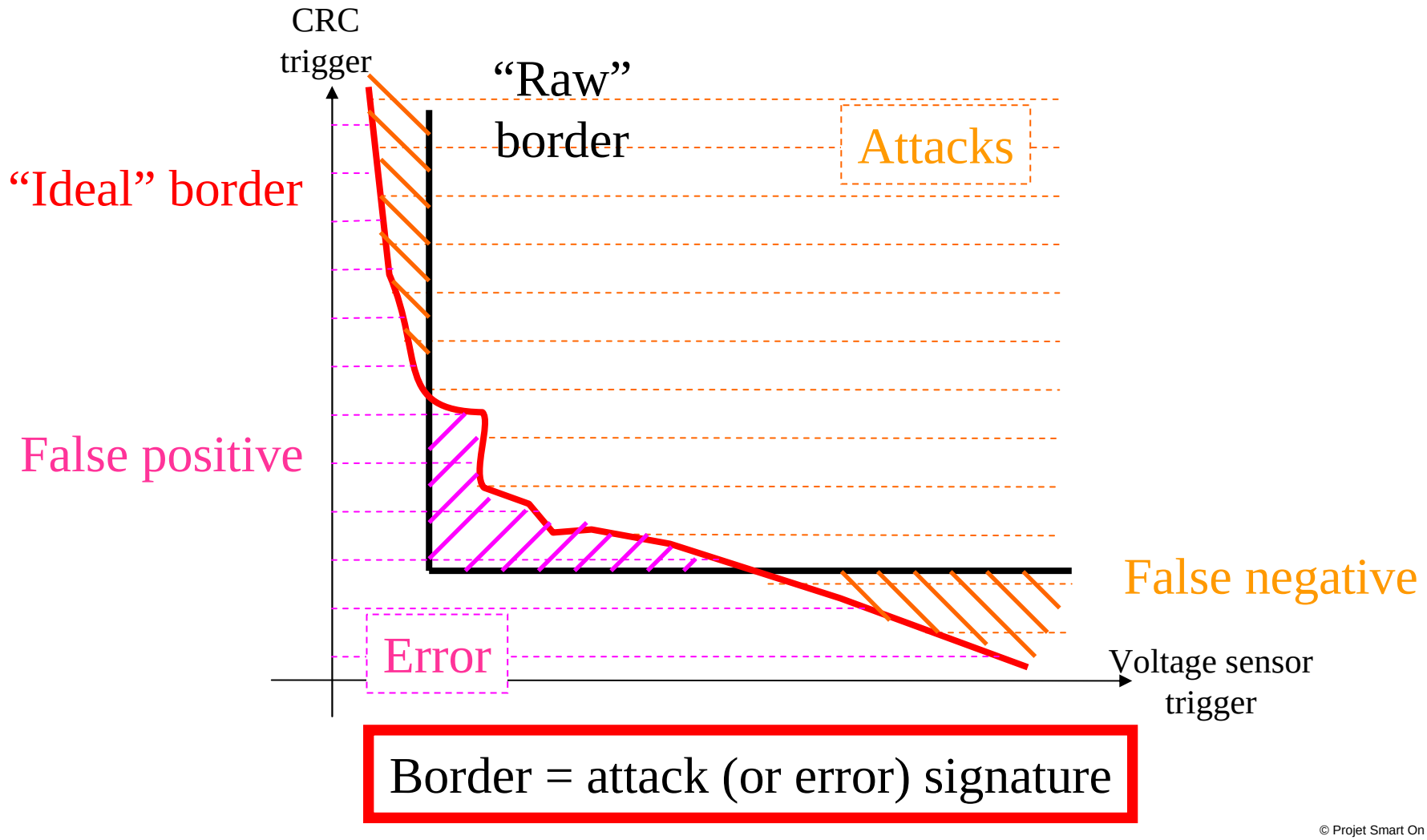
- Based upon the scrambling of an audio/video flow which can be descrambled with a key if and only if the correct right is owned by the smartcard.
- 3 class of commands are used by the system :
  - Subscription writing (Keys, Rights) **Very sensitive**
  - Descrambling (control word) **Sensitive**
  - Subscriber operations (parental control) **Not very sensitive**

# Strategy of security



# Strategy of Security

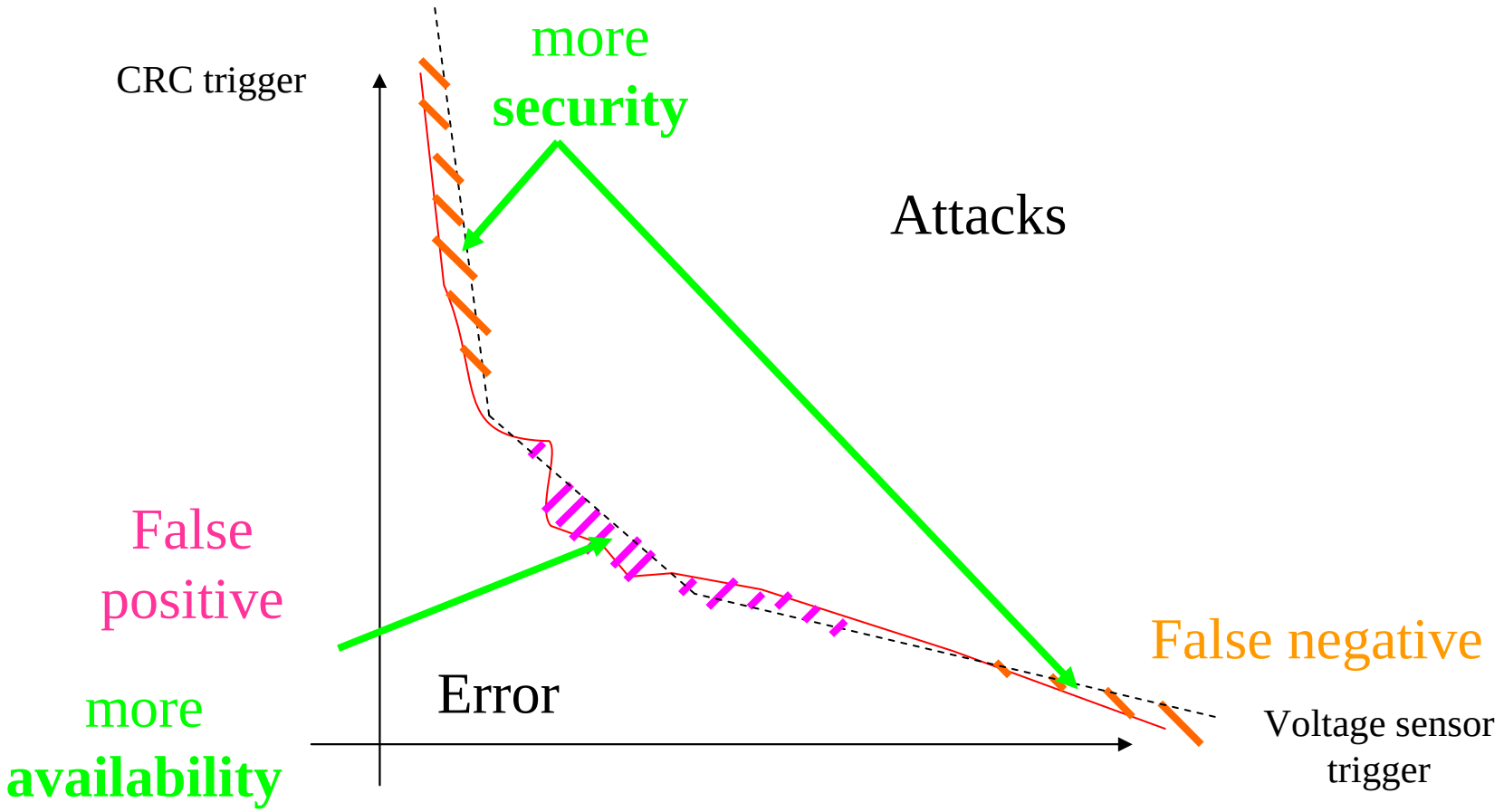
"Naive approach" : Hardcoded "attack/error" border




Border = attack (or error) signature

# Smart On Smart

More complex borders : more security and more availability

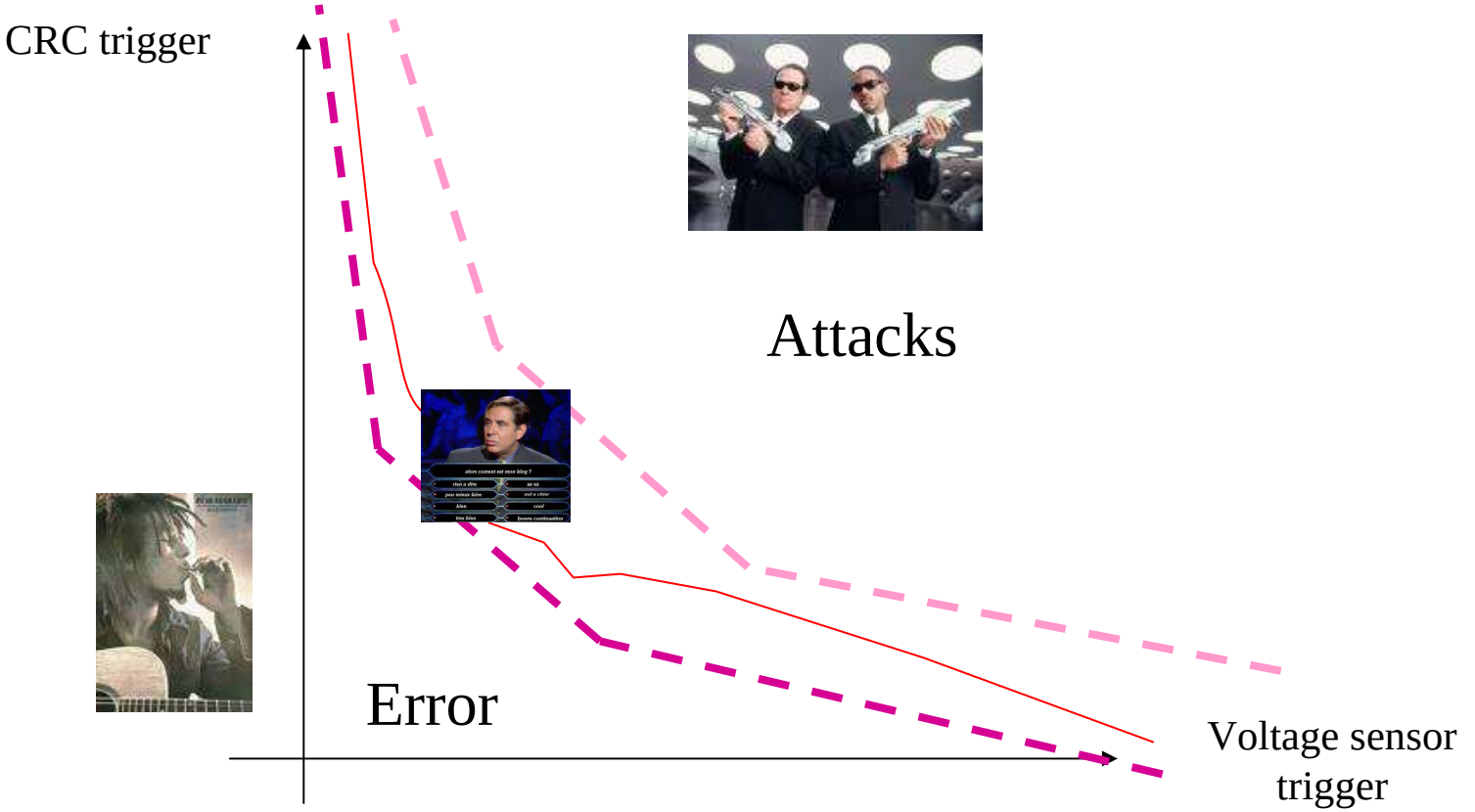



SOS project proposes a method to approximate the “ideal” border

# Smart On Smart

Response which depends on the distance from the borders

→ more flexibility and performance optimization




SOS project proposes an architecture which enable to graduate response

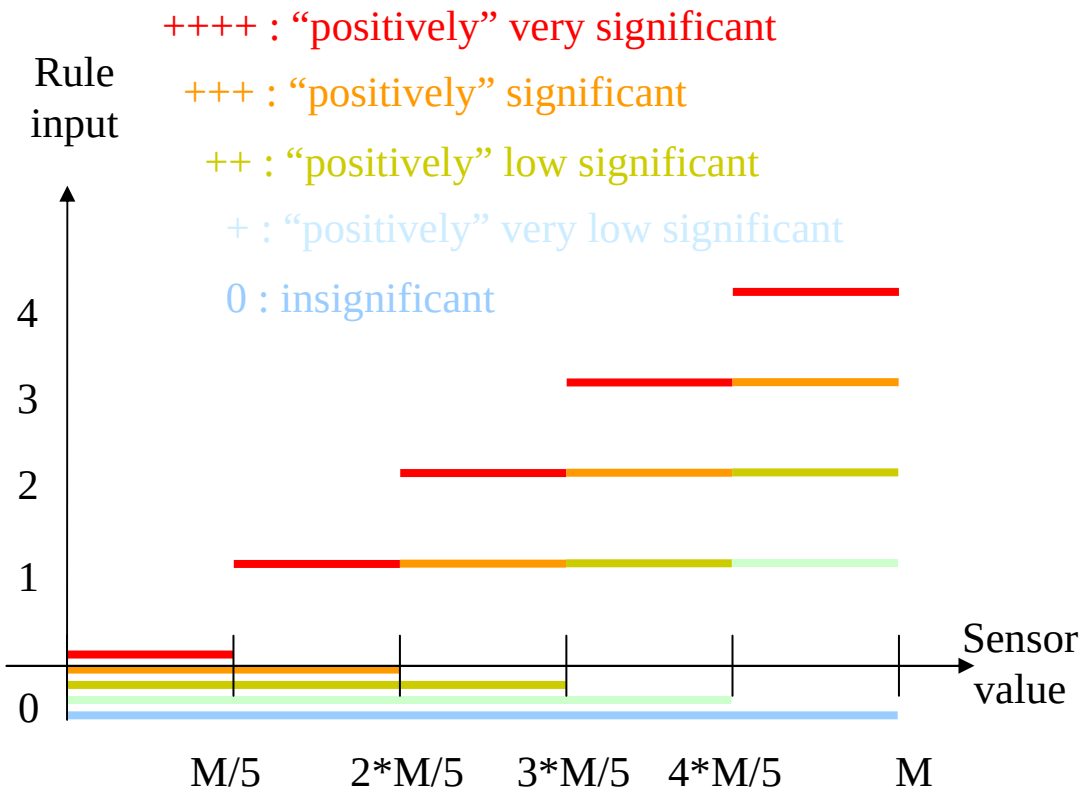
# Method to approximate “ideal” border 1/3

Choice : Definition of borders with fuzzy logic

Counters ⇨ Discretization ⇨ Ponderation ⇨ Rule evaluation

“Sensors” count events

| Numéro de compteur | Compteurs équivalents | Valeurs |
|---|---|---|
| 1 | CPT_COMMAND_WRONG_MAC | 0..65535 |
| 2 | CPT_COMMAND_WRONG_PIN | 0..65535 |
| 3 | CPT_COMMAND_WRONG_DATA | 0..65535 |
| 4 | CPT_COMMAND_UNEXPECTED | 0..65535 |
| 5 | CPT_COMMAND_REPLAY | 0..65535 |
| 6 | CPT_COMMAND_USELESS | 0..65535 |
| 7 | CPT_USER_DATA_INTEGRITY | 0..65535 |
| 8 | CPT_EXECUTION_FLOW | 0..65535 |
| 9 | CPT_REDUNDANCY | 0..65535 |
| 10 | CPT_VM_RUNTIME_INTEGRITY | 0..65535 |
| 11 | CPT_VM_REGISTRY_INTEGRITY | 0..65535 |
| 12 | CPT_VM_CRYPTO_INTEGRITY | 0..65535 |
| 13 | CPT_VM_CODE_INTEGRITY | 0..65535 |
| 14 | CPT_VM_EXECUTION_FLOW | 0..65535 |
| 100 | CPT_HW_SENSOR_LIGHT | 0..65535 |
| 101 | CPT_HW_SENSOR_GLITCH | 0..65535 |
| 102 | CPT_HW_SENSOR_FREQUENCY | 0..65535 |



# Method to approximate “ideal” border 2/3

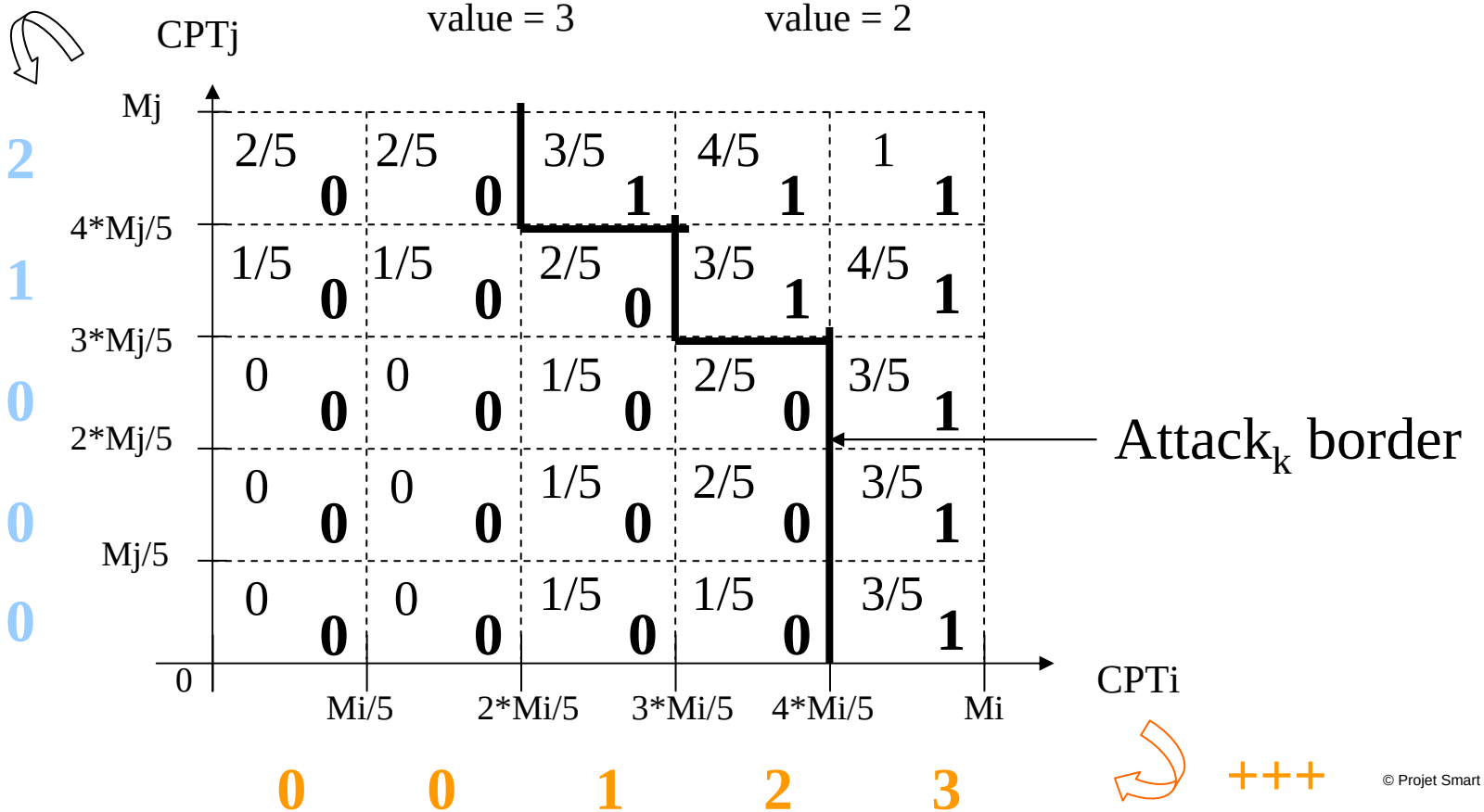
Example of rule evaluation :

“Probability”

$$Attack_k = \{ ( ( CPT_i +++ ) + ( CPT_j ++ ) ) / 5 \} > 0.4$$

Rule input max value = 3

Rule input max value = 2



# Method to approximate “ideal” border 3/3

Counters
#CT =11

Rules refinement

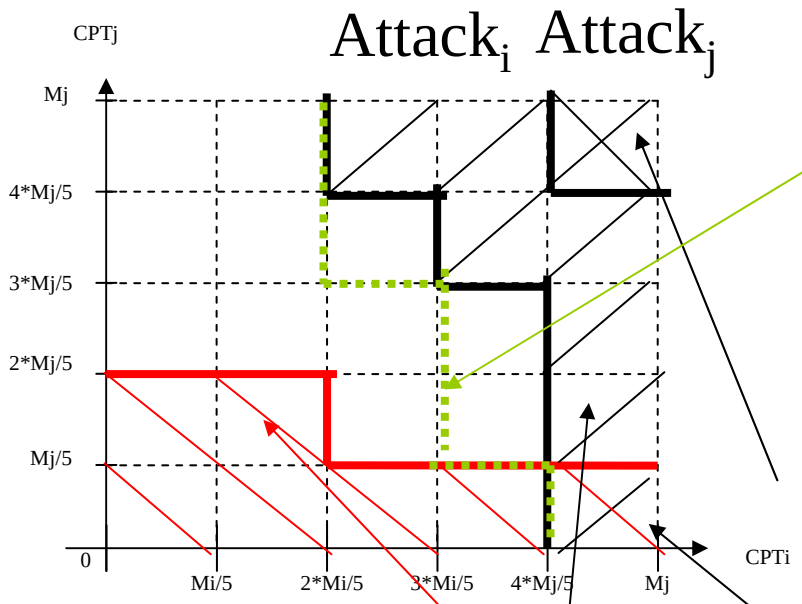
Rule Errors

Rule Attacks

Generation of the whole space
($5^{\#CT} = 5^{11}$ elements)

| 1 | 2 | 3 | 4 | 5 | \|1 | 2 | 3 | 4 | \|0 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|
| I1 | I1 | I1 | I1 | I1 | \|0.0 | 1.0 | 0.0 | 1.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I1 | I2 | \|0.0 | 1.0 | 0.0 | 1.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I1 | I3 | \|0.0 | 1.0 | 0.0 | 1.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I1 | I4 | \|0.0 | 1.0 | 0.0 | 1.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I1 | I5 | \|0.0 | 1.0 | 0.0 | 0.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I2 | I1 | \|0.0 | 1.0 | 0.0 | 1.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I2 | I2 | \|0.0 | 1.0 | 0.0 | 1.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I2 | I3 | \|0.0 | 1.0 | 0.0 | 1.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I2 | I4 | \|0.0 | 1.0 | 0.0 | 1.0 | \|0.0 | 1.0 |
| I1 | I1 | I1 | I2 | I5 | \|0.0 | 1.0 | 0.0 | 0.0 | \|0.0 | 1.0 |

Statistics tools




Ideal border

Subsumption

Error$_k$

Coverage rates

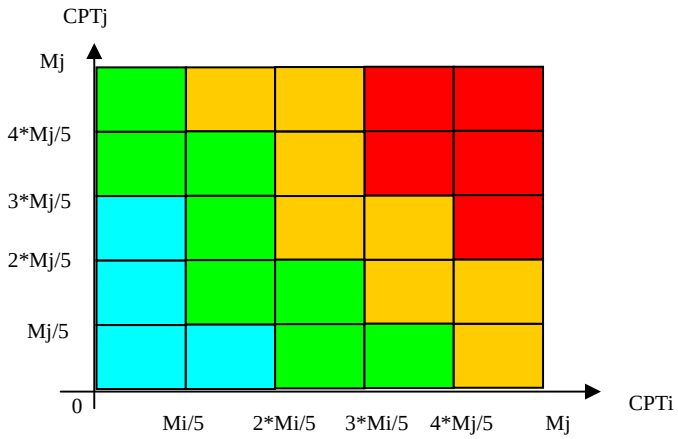
Potential conflicts

…

# Graduated Reaction

Rule evaluation ⇨ Security levels ⇨ Control of counter-measures

| | Redundancy | Scrambling | Reset | Kill |
|---|---|---|---|---|
| Safe | No | No | No | No |
| Unsafe | *2 | L1 | No | No |
| Critical | *3 | L2 | Yes | No |
| Final | - | - | - | Yes |

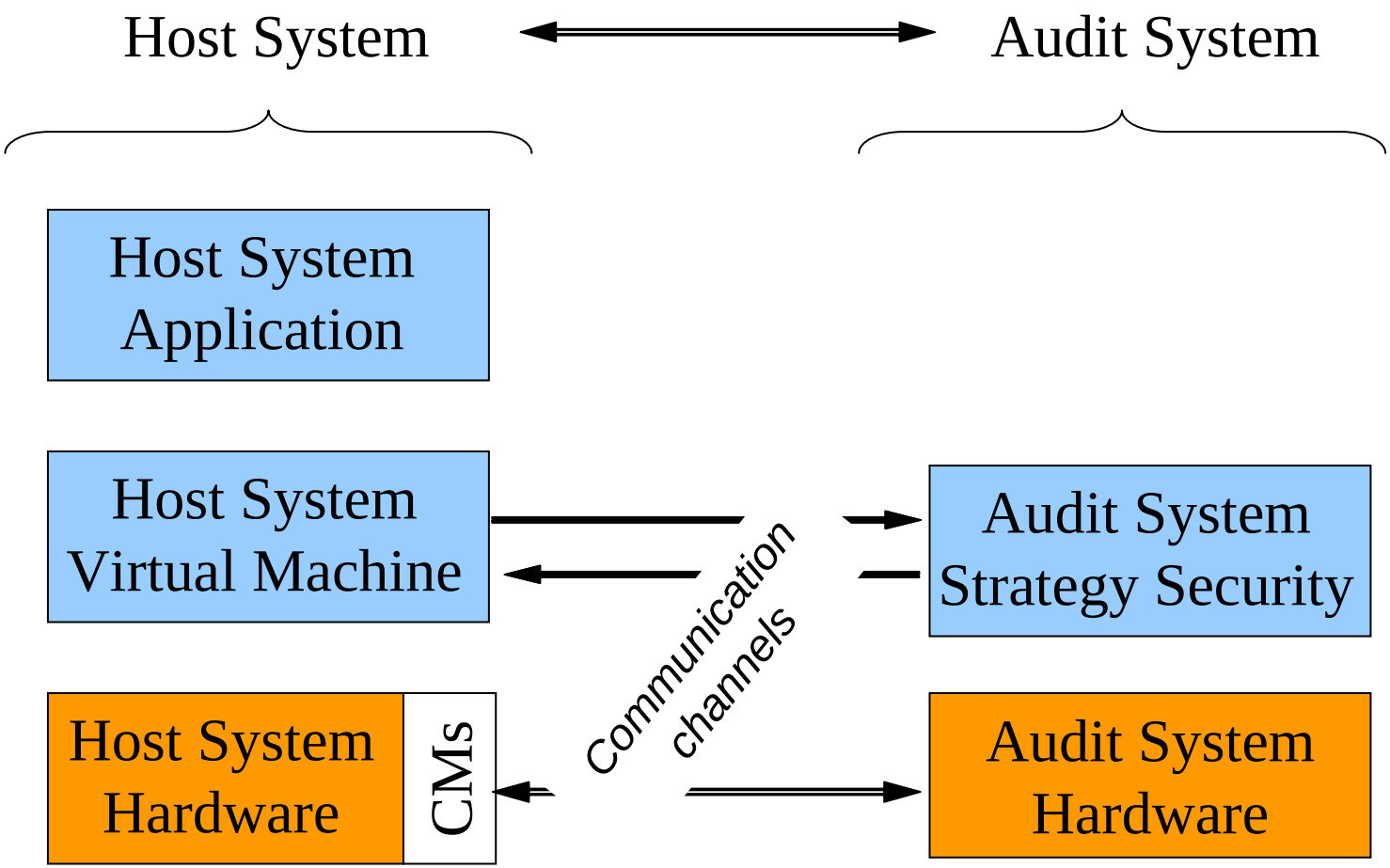



Performance

Security

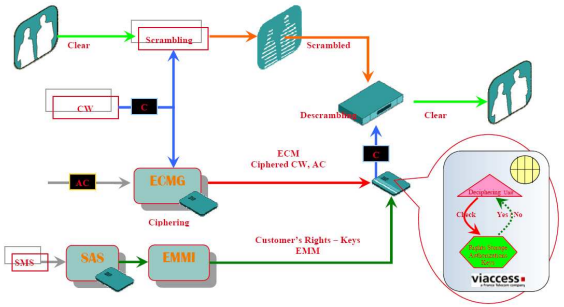
# Architecture

Fault attacks → Dedicated hardware = Audit System

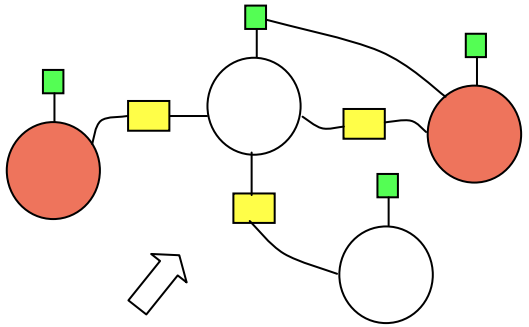


# Prototyping 1/2

Conditional access




Ad-hoc OS



Host System Application

TL JavaCard 2.2

Host System Virtual Machine

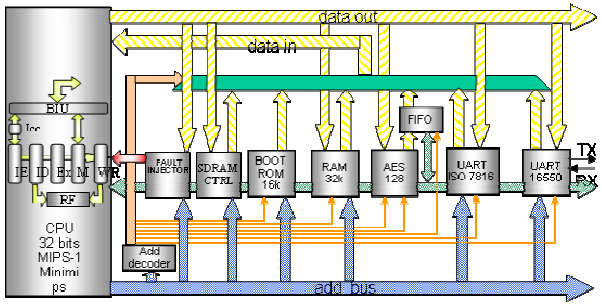
Audit System Strategy Security

Host System Hardware

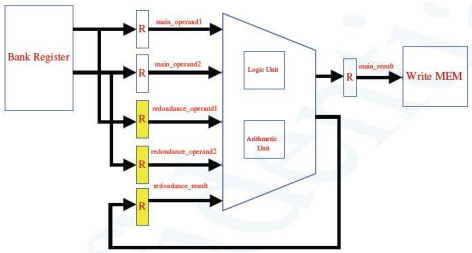
CM

Audit System Hardware

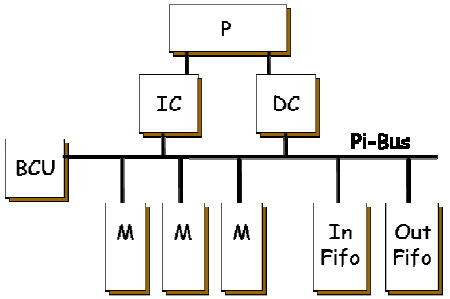
Mips




Redundancy, scrambling, etc…




Mips – R3000

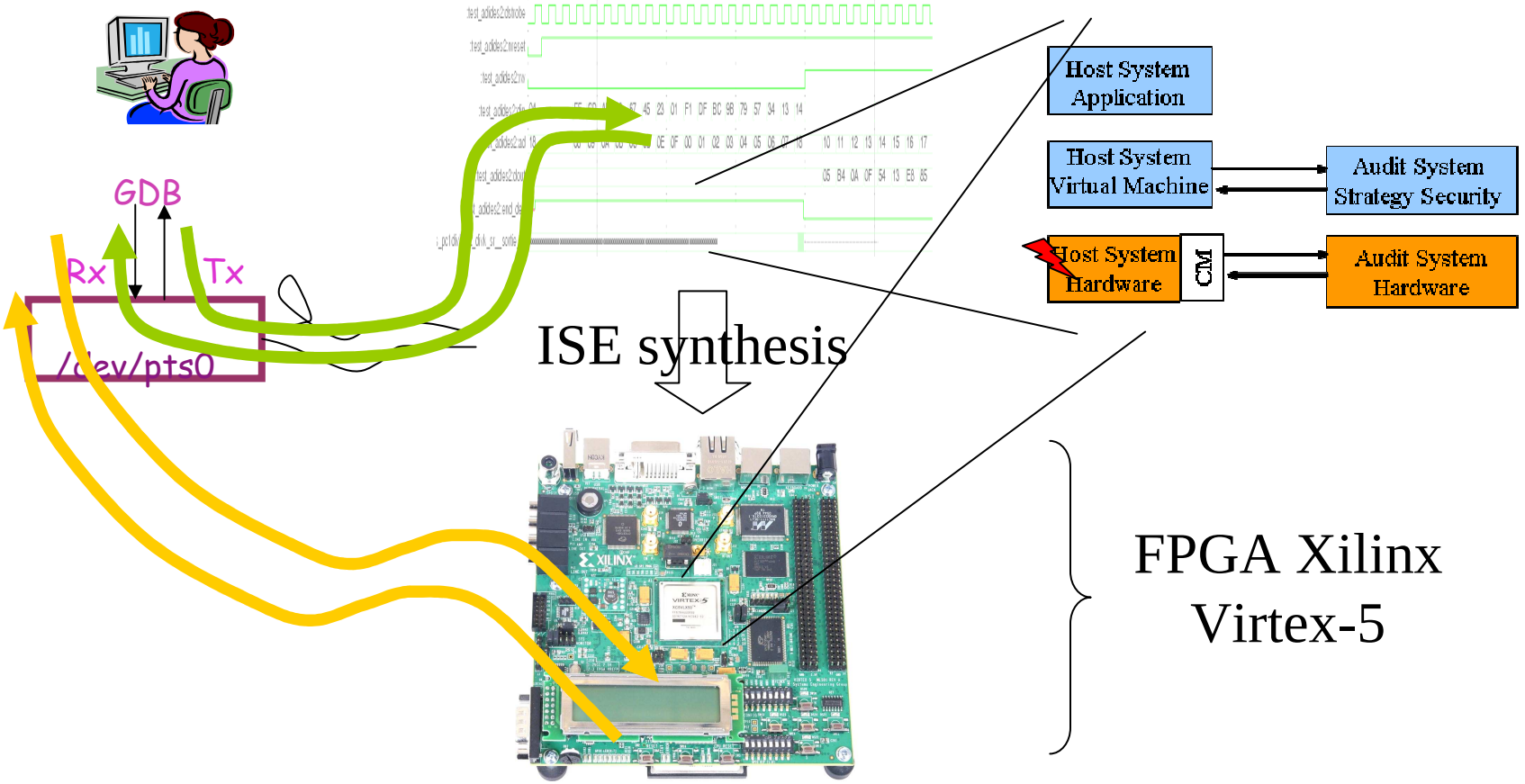


# Prototyping 2/2

Debug on simulation on target

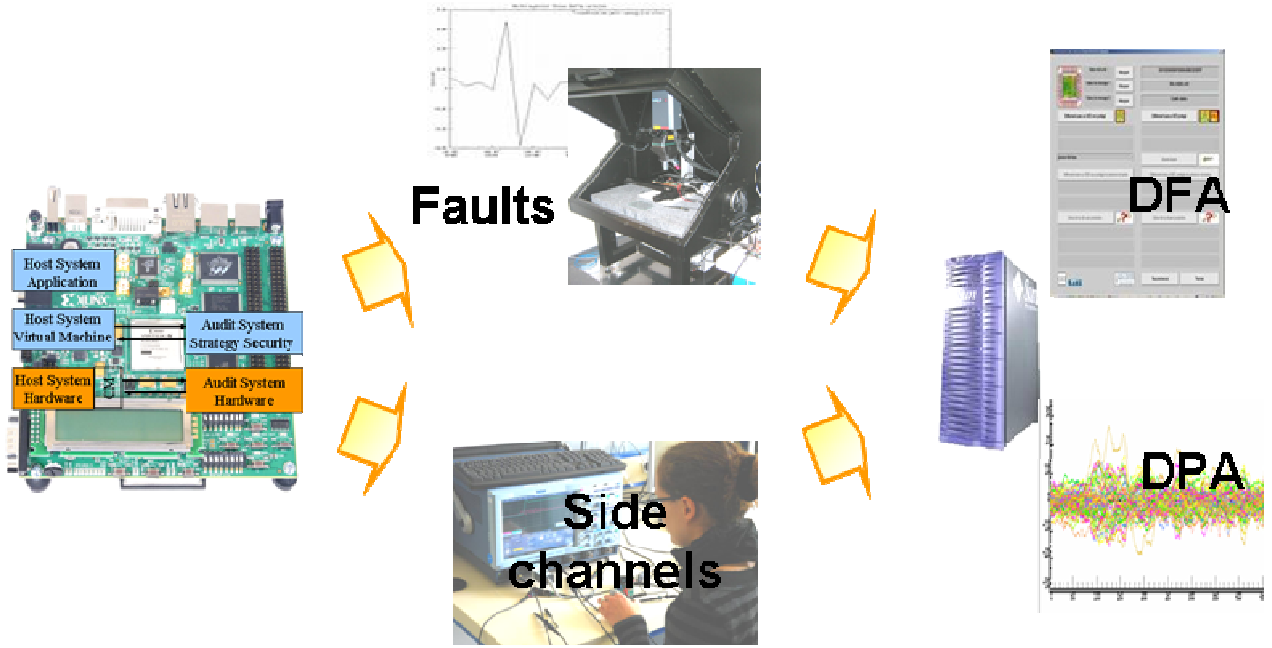
Simulation

VHDL SOS models



# Testing SOS

The Strategy of Security must be tested. Hardware simulation failure is used for a realistic approach. The “attacks decision rules” will be refined with these tests.



# Conclusion on SOS/SOS

- New approach to deduct attacks from errors counters.
- New approach to react against these attacks
- Dynamic counter measures : the behavior of the device evolve.
  → to use long scripts is more difficult
- The counter measures are dissociated from the errors/attacks detected.
  → the attacker gains less information