SMART SECURITY MANAGEMENT IN SECURE DEVICES

PROOFS'15 – SAINT-MALO

Bruno Robisson (bruno.robisson@cea.fr)

Michel Agoyan (formerly CEA),

Patrick Soquet (formerly Viaccess-Orca) ,

Sébastien Le-Henaff (Viaccess-Orca),

Franck Wajsbürt (UPMC),

Pirouz Bazargan-Sabet (UPMC) and

Guillaume Phan (Trusted Logic)

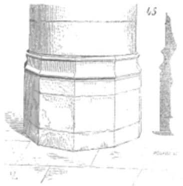
17 SEPTEMBER 2015

PROTECTIONS AGAINST PHYSICAL ATTACKS

Sensors

Light, voltage and frequency sensors ; spatial, temporal or information redundancy ; monitoring execution flow; anti-probing layer; etc…

Counter-measures



Metal shields; power filter ; balanced logic ; balanced place and route; etc.

Noise generator ; random dummy instructions ; memory scrambling ; masking; internal clock, etc.

Mute; reset application or applet; delete data (=kill); etc. } Reactions

Security is achieved by implementing (too) many protections

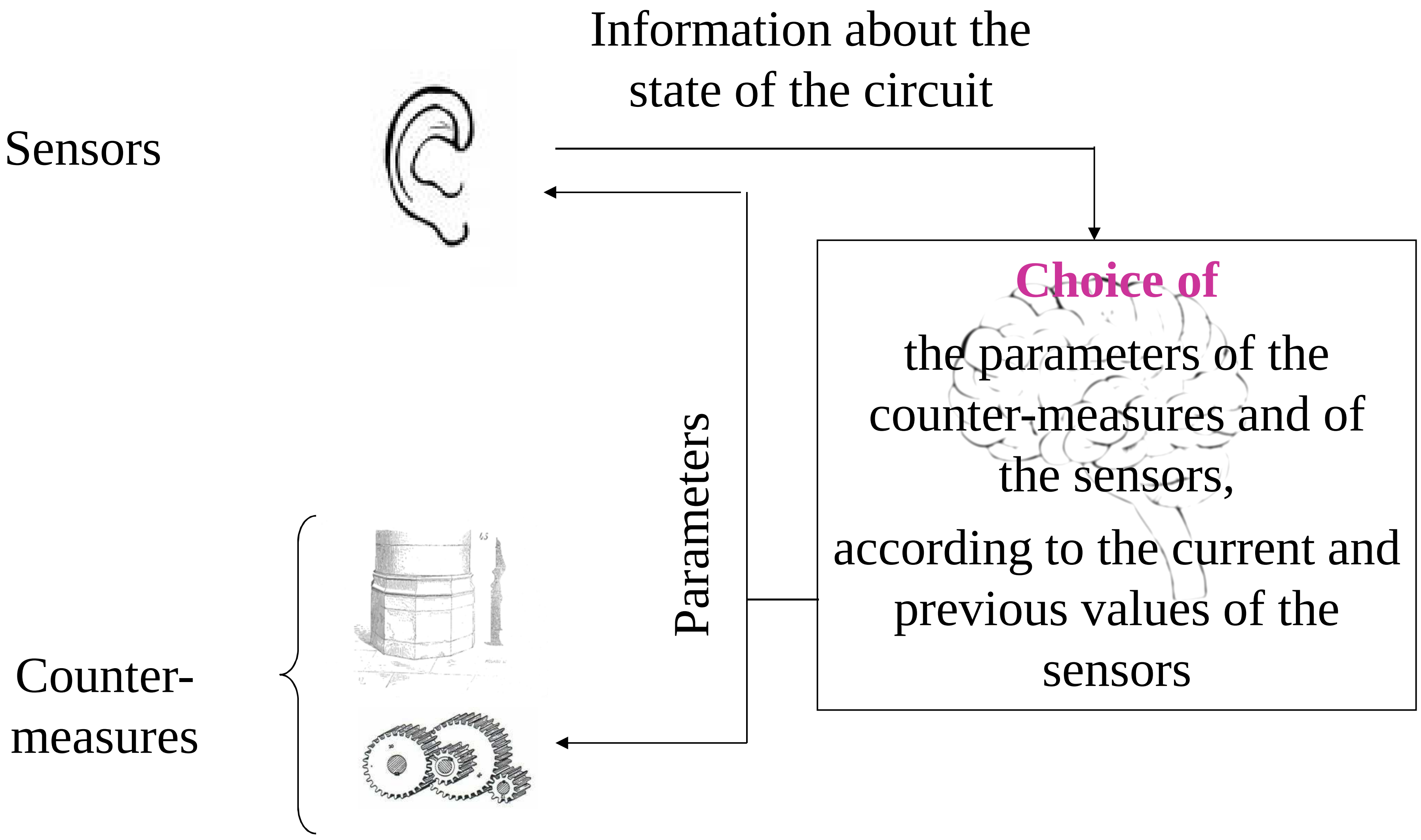
↑security but ↓performances and ↓ availability

Complementary approach: Smart management of protections through the application of a complex “strategy of security”

SCHEDULE

- Strategy of security
  - Definition
  - Main requirement
  - Secondary requirement
- Application
  - Case study: Conditional Access System (CAS) for pay TV
  - Architecture of the Conditional Access System
  - Protections
  - Configurations of protections
  - Example of strategy of security
- Prototyping
  - Architecture of the (Conditional Access System + Strategy of security)
  - FPGA prototype
  - Validation
- Conclusions and perspectives

MANAGEMENT OF THE SECURITY : DEFINITION



Information about the state of the circuit

Sensors

Parameters

Choice of

the parameters of the counter-measures and of the sensors,

according to the current and previous values of the sensors

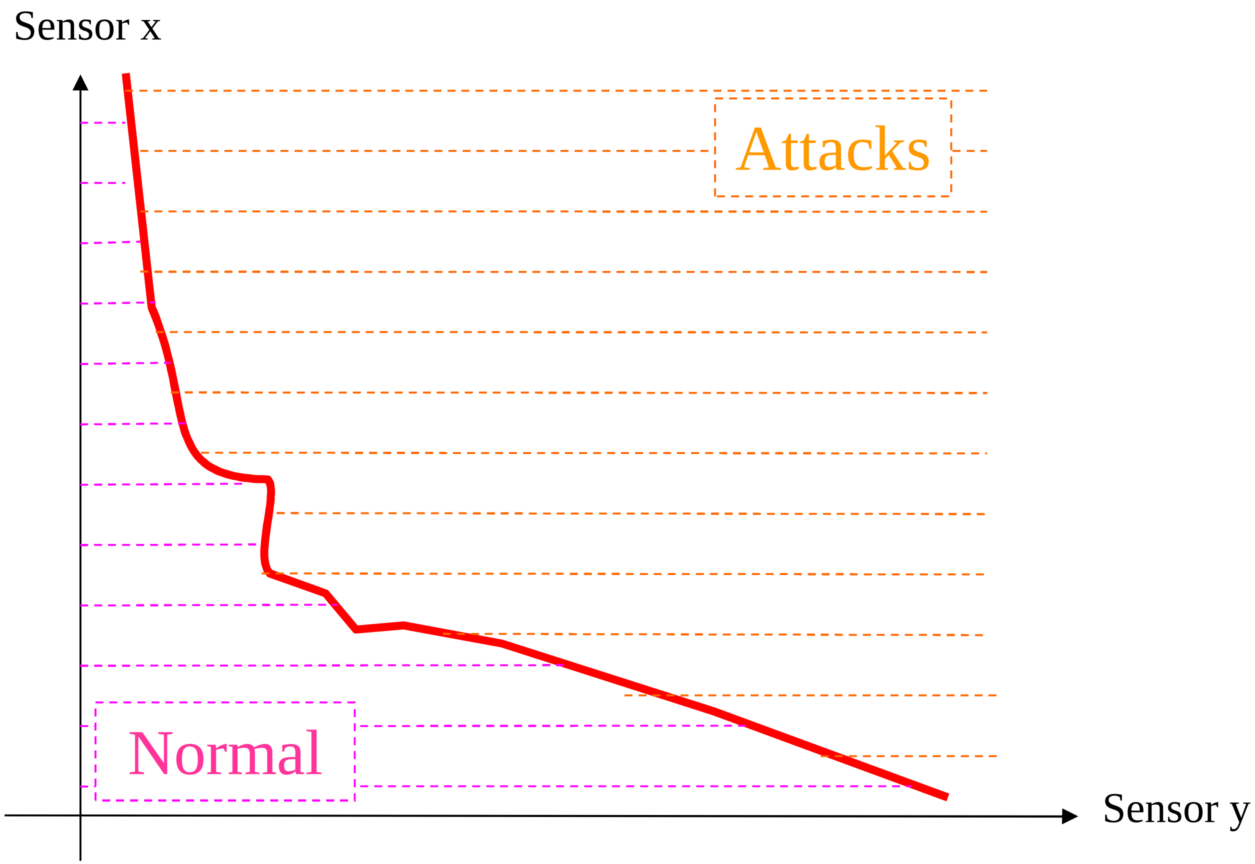
Counter-measures

MAIN REQUIREMENT

“Theoretic” border

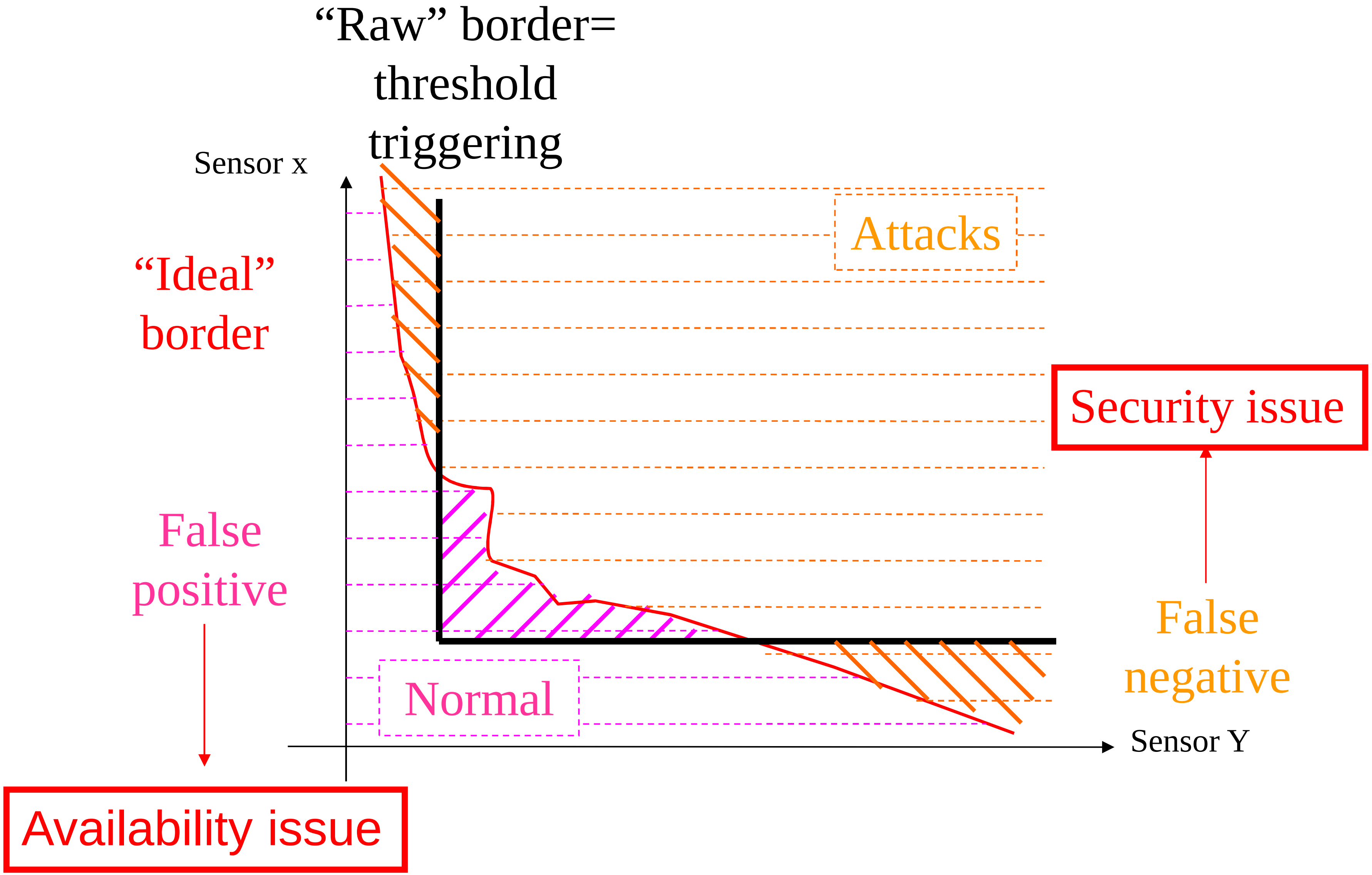
=

“Attack” signature

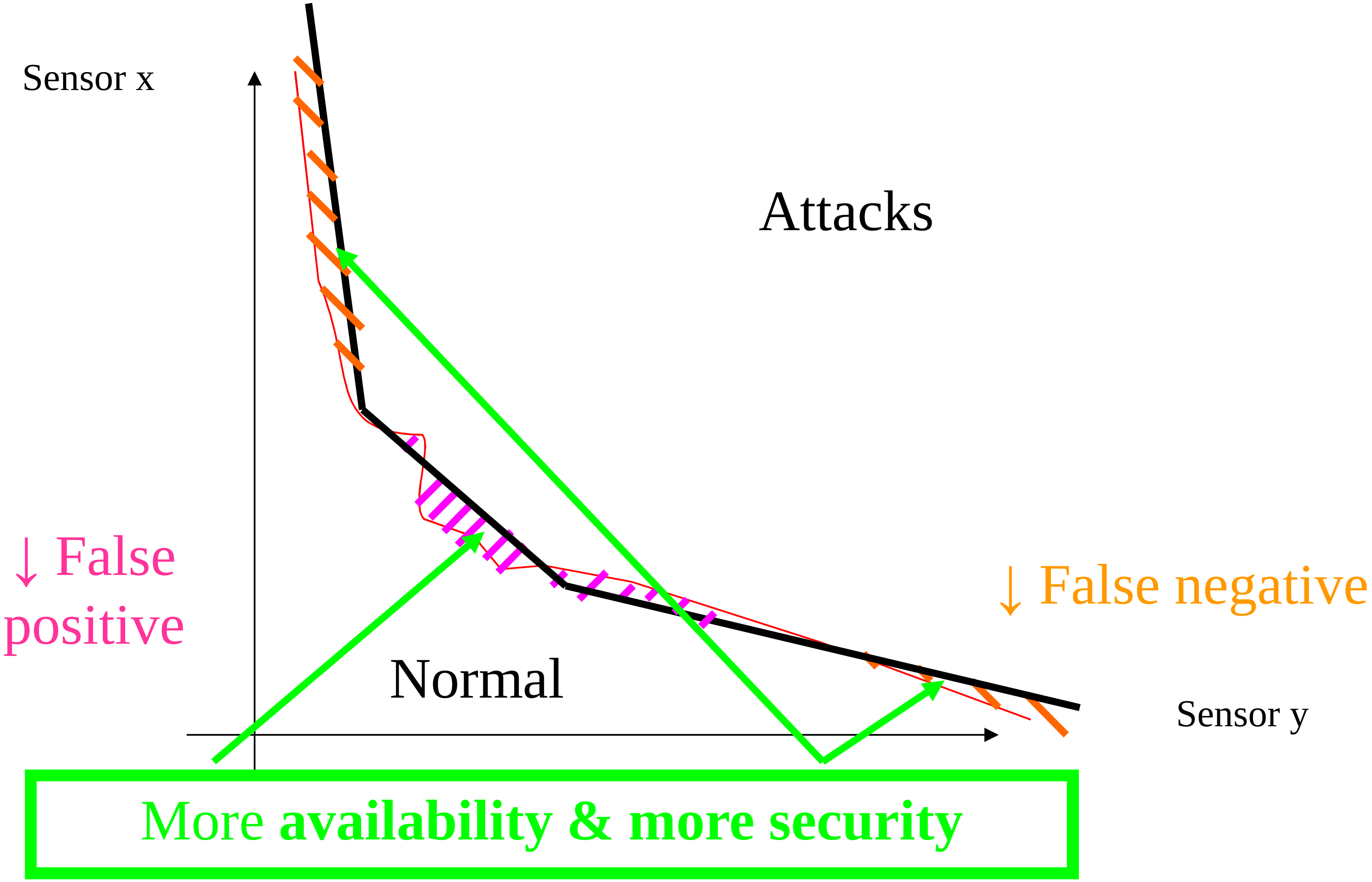


To be able to distinguish attacks from normal behaviors

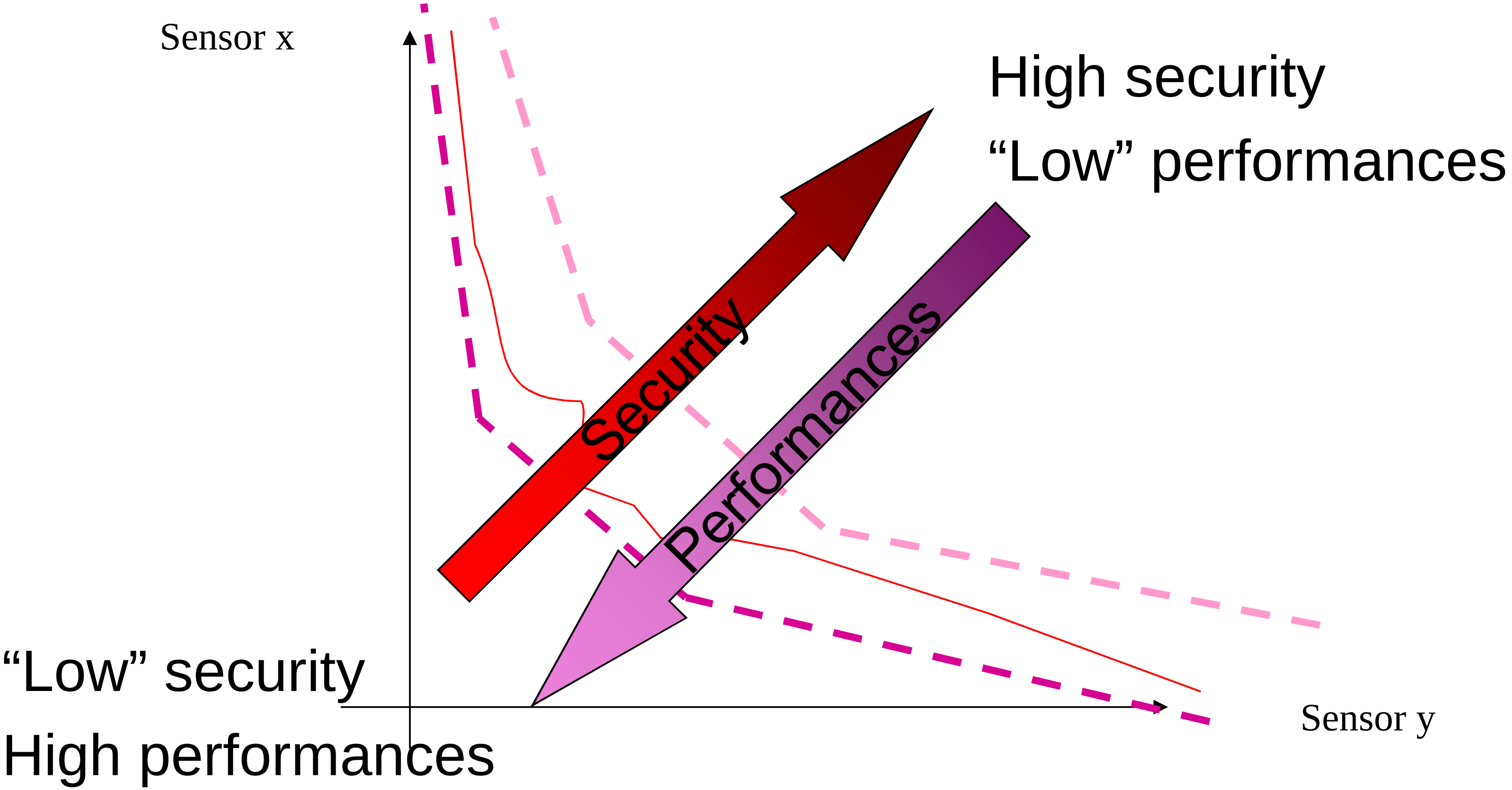
STATE OF THE ART “ATTACK / NORMAL”



Complex management of security



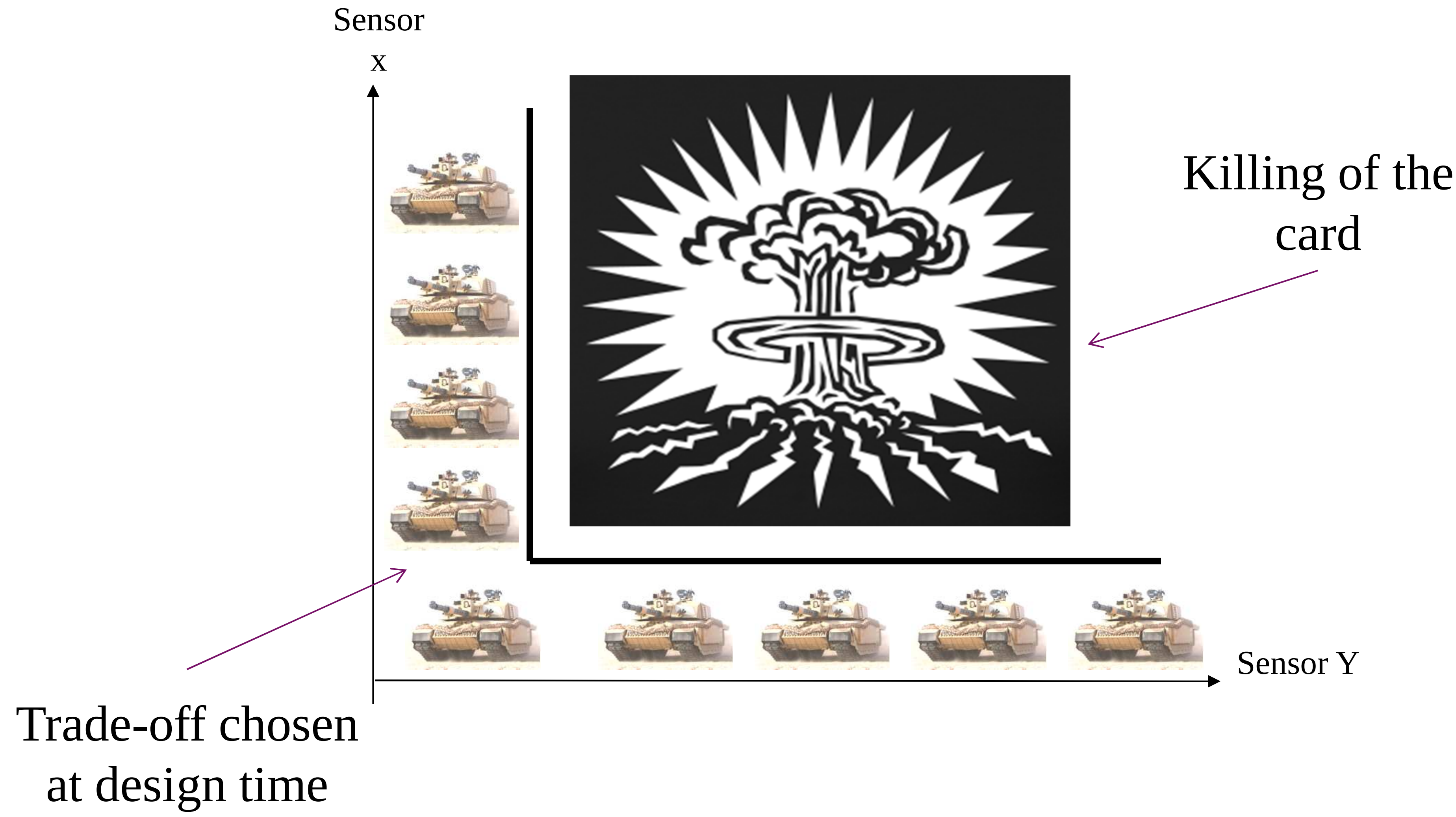
Secondary requirement



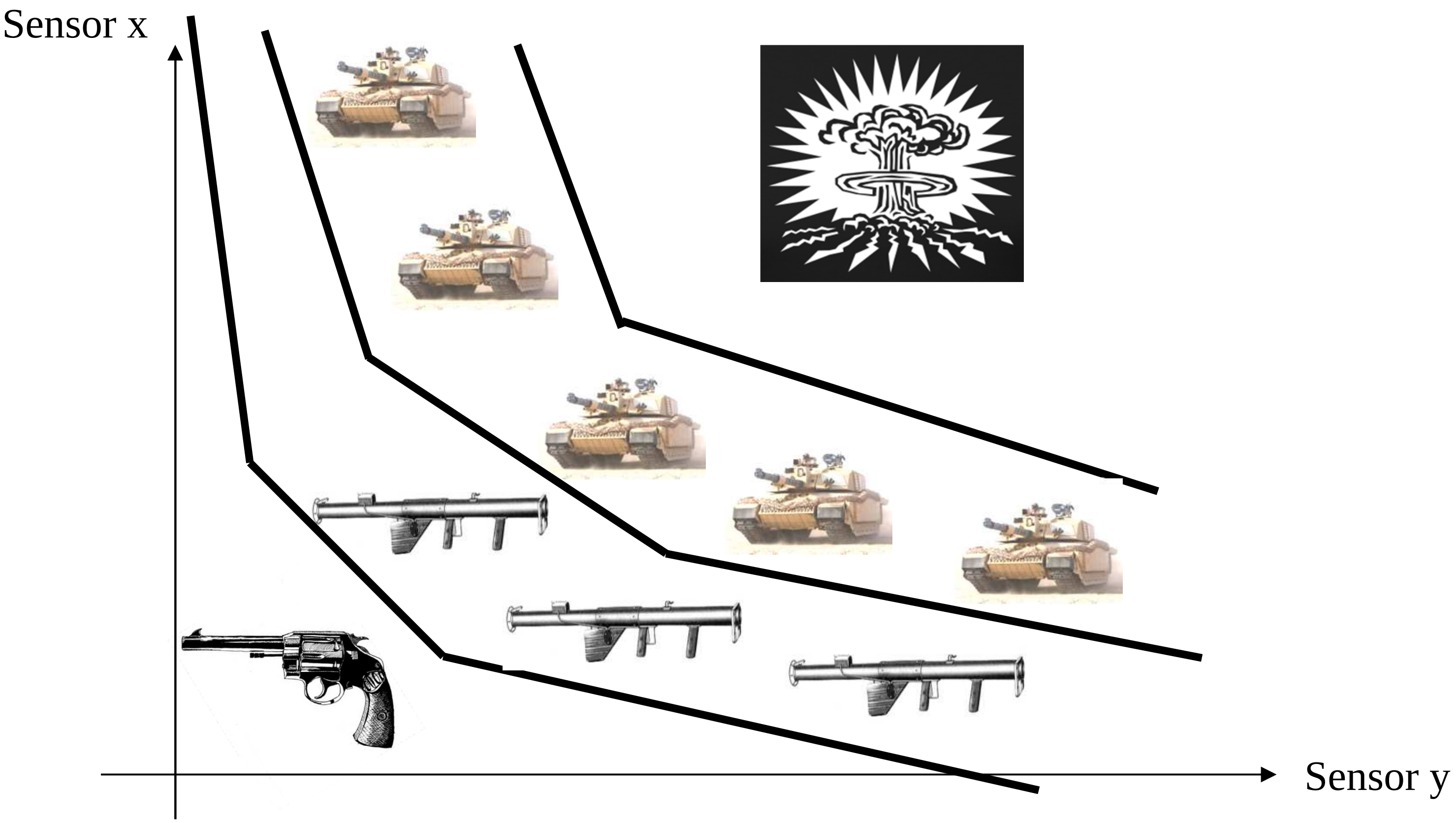
To enable to have dynamical trade-off between performances and security

STATE OF THE ART “DYNAMIC TRADE-OFF“

“basical” configurations



Complex management of security



Increase gradually the security with the risk of attack to obtain optimal performances **without** compromising the security

SCHEDULE

- Strategy of security
  - Definition
  - Main requirement
  - Secondary requirement
- Application
  - Case study: Conditional Access System (CAS) for pay TV
  - Architecture of the Conditional Access System
  - Protections
  - Configurations of protections
  - Example of strategy of security
- Prototyping
  - Architecture of the (Conditional Access System + Strategy of security)
  - FPGA prototype
  - Validation
- Conclusions and perspectives

APPLICATION: CONDITIONAL ACCESS FOR PAY-TV

Principle

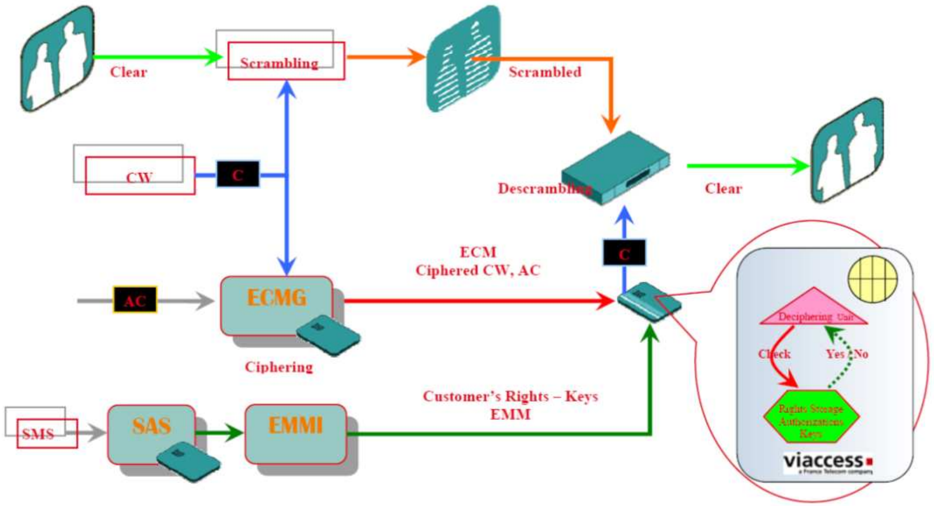
- Conditional Access Systems (CAS) protect a content (such as radio, TV, data stream) by requiring certain criteria to be met before granting access to this content.
- One criteria : Own a smartcard which stores “secret” information
- 3 class of commands are used by the system :
  - Subscription management (Keys, Rights) **Very sensitive**
  - Descrambling (control word) **Sensitive**
  - Subscriber operations (parental control) **Not very sensitive**

Needs

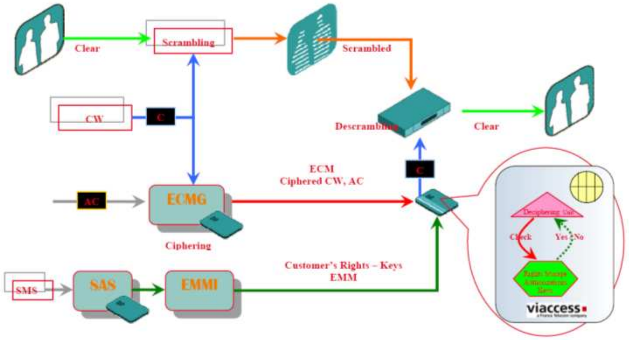
- High level of security
- Real time performance
- High level of availability

Extra needs

- Low power for integration in mobile phones



CAS CARD SYSTEM = “HOST”

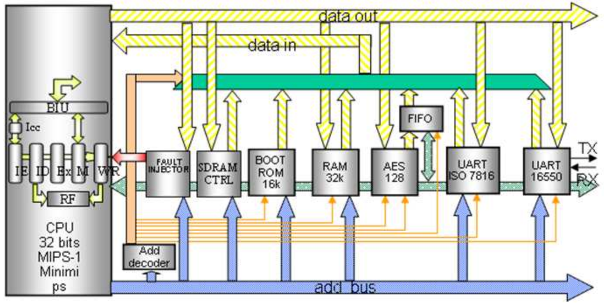


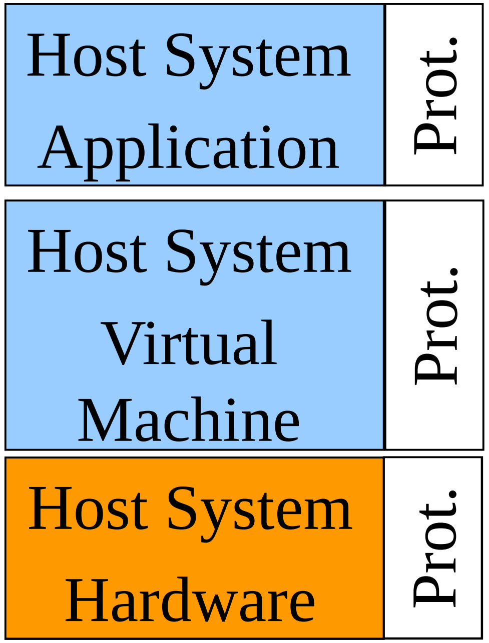
Conditionnal Access

JavaCard 2.2

GlobalPlatform API

MiniMips



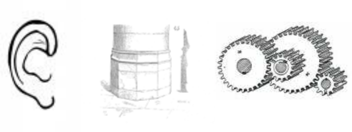


Host System Application | Prot.

Host System Virtual Machine | Prot.

Host System Hardware | Prot.

SW protections



HW protections

PROTECTIONS

Redundancy (HW): Execute **RL** (for Redundancy Level) times the same computation and compare the results.

If a difference is observed the number of corrupted execution (noted **CE**), is increased.

Sensors (HW) : Emulation of voltage (**VS**) and light (**LS**) sensors

Sensors (SW) : # of wrong PIN (**PE**), # of cryptographic execution (**CO**), # of corrupted execution flow (**EFE**), # of methods processed without error (**NE**), sensitivity of data (**DS**), MAC error message (**ME**), etc.

Insert randomly **D**ummy random **I**nstructions (parameters

D: max # of consecutive usefull instructions

N: max # of consecutive dummy instructions)

Random **P**ower **G**enerator (parameter

R: # of activated PRNG)

Mute/reset

Kill

CHOICE OF COUNTER-MEASURES CONFIGURATIONS

Configuration	Safe (ref)	Unsafe	Critical	Fatal
Security against observation	1.0	122.5	1346.7	-
Security against perturbation	1.0	6270.5	1.10^{8}	-
Time	1.0	4.0	7.8	-
Energy consumption	1.0	5.2	15.6	-
Sensors	ON	ON	ON	-
Redundancy	RL=1	RL=2	RL=3	-
Random Power Generator	R=0	R=3	R=10	-
Insertion Dummy Instruction	D=2;N=0	D=3;N=4	D=4;N=8	-
Mute/reset	No	No	Yes	-
Kill	No	No	No	Yes

Wide range of trade-off between:

Security

AND

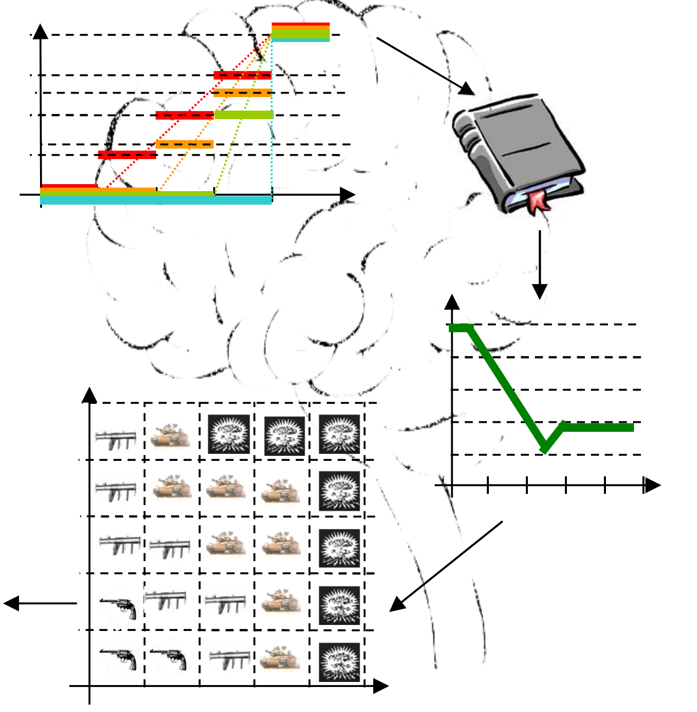
Performance

SECURITY STRATEGY

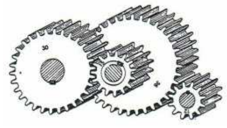
Sensors

Name of sensors	Values
LS	{0,…,5}
VS	{0,…,10}
PE	{0,…,10}
NE	{0,…,1000}
…	…

Fuzzy logic reasoning



Counter-measures



Safe	Unsafe	Critical	Fatal
ON	ON	ON	-
RL=1	RL=2	RL=3	-
R=0	R=3	R=10	-
D=2;N=0	D=3;N=4	D=4;N=8	-
No	No	Yes	-
No	No	No	Yes

R0: "IF the number of methods that have processed without error (NE) is VERY HIGH THEN the attack is LOW "

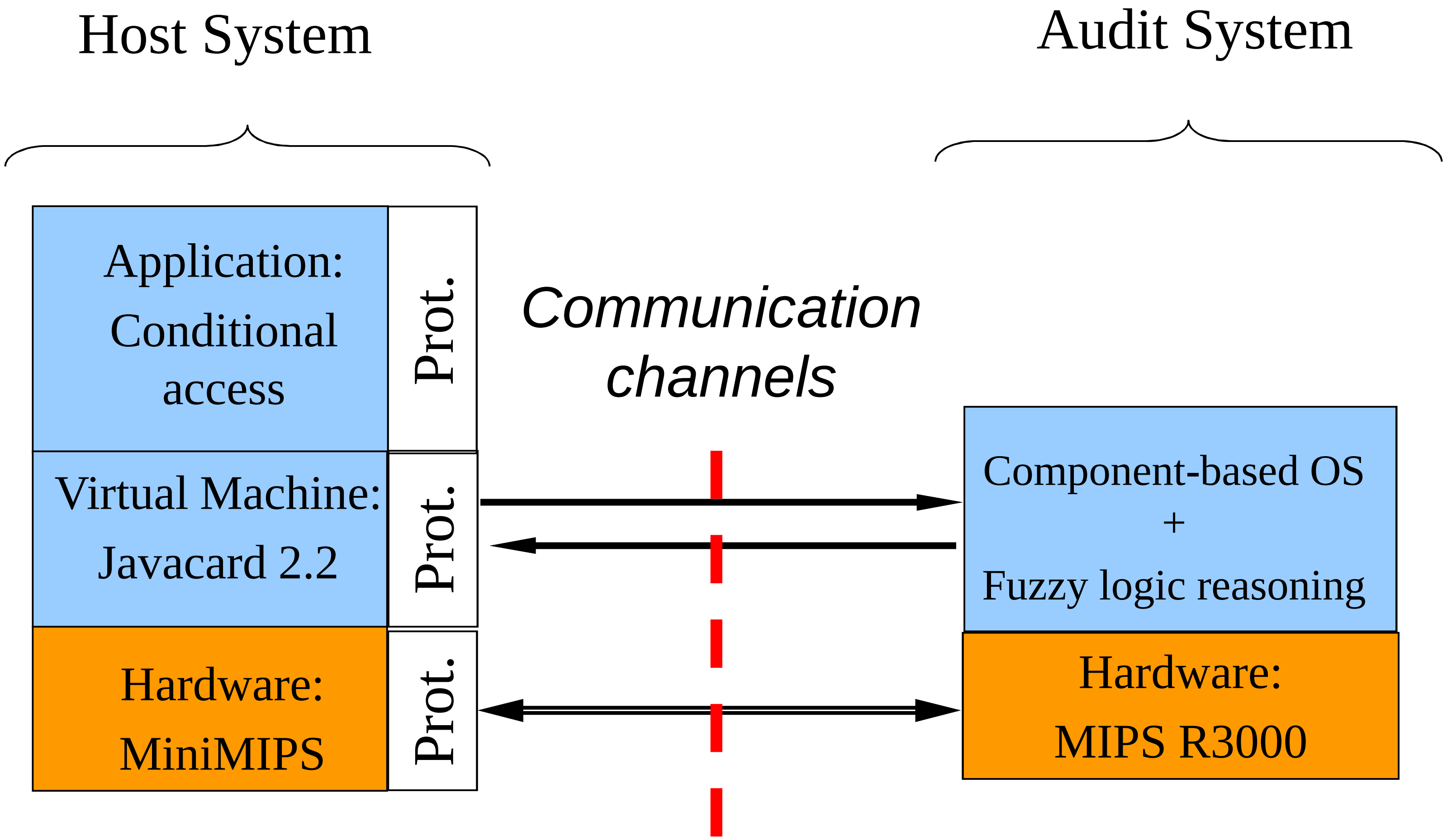
R1: "IF the voltage (VS) is RATHER HIGH and the light (LS) is HIGH THEN the attack is HIGH "

R2: "IF the number of cryptographic errors (CO) is RATHER HIGH THEN the attack is HIGH "…

SCHEDULE

- Strategy of security
  - Definition
  - Main requirement
  - Secondary requirement
- Application
  - Case study: Conditional Access System (CAS) for pay TV
  - Architecture of the Conditional Access System
  - Protections
  - Configurations of protections
  - Example of strategy of security
- Prototyping
  - Architecture of the (Conditional Access System + Strategy of security)
  - FPGA prototype
  - Validation
- Conclusions and perspectives

PROTOTYPE ARCHITECTURE



Transfers of sensor values and of parameters of protection

BUT NO TRANSFERT OF SENSITIVE DATA!!

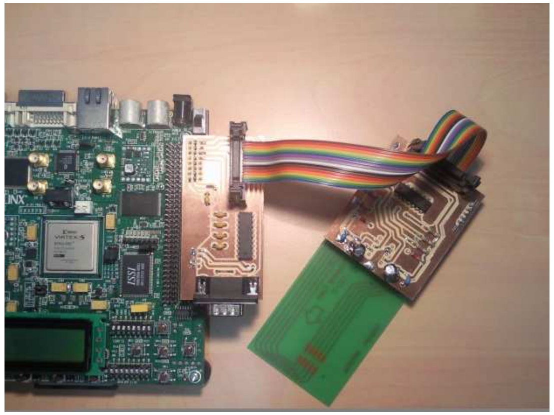
PROTOTYPE ON FPGA

Based on Xilinx® ML501 virtex5 board

- Host System :
  - 32-bit µprocessor @ 50 MHz
  - MIPS-1 instruction set
  - 5-stage pipeline
  - Harvard architecture
  - 128 KB E2 emulation
  - 896 KB Data/Instruction
  - AES-128
  - ISO 7816-3 UART + connector
  - UART (111520 bauds) + DB9
  - Embedded software stubs for remote debugging
  - Embedded fault injection emulation

Host System only :

Number of Slices	2462 out of 7200	34%
Number of Slice Registers	2421 out of 28800	8%



Audit System (+5 to +20%)

- Audit system :
  - Mips like cpu @50MHz
  - 4KB Data
  - 32 KB Instruction
  - Simple UART + DB9
  - ICU + comm FIFO

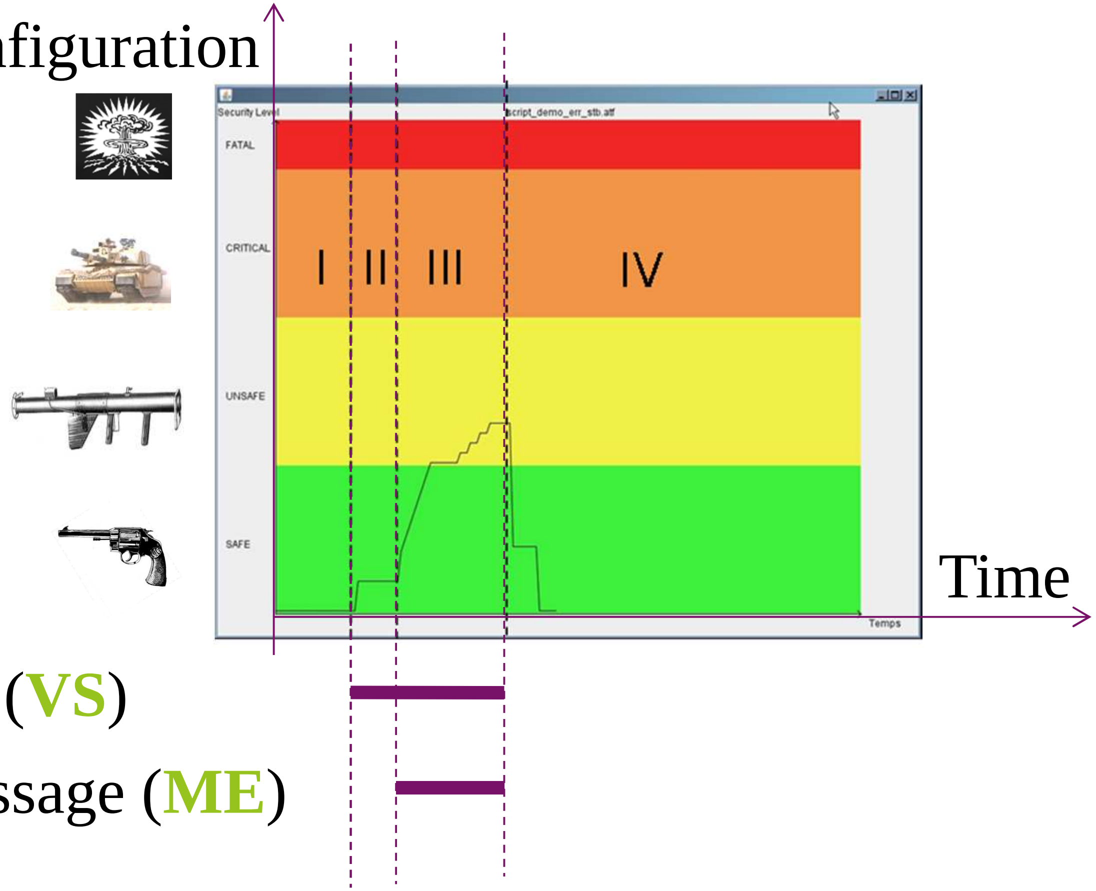
Host System + Audit system :

Number of Slices	3490 out of 7200	48%
Number of Slice Registers	4534 out of 28800	15%

VALIDATION

Theoretical analysis (cf paper)

Simulation of scenario : low quality card reader



Configuration

Time

Voltage sensor (**VS**)

MAC error message (**ME**)

SCHEDULE

- Strategy of security
  - Definition
  - Main requirement
  - Secondary requirement
- Application
  - Case study: Conditional Access System (CAS) for pay TV
  - Architecture of the Conditional Access System
  - Protections
  - Configurations of protections
  - Example of strategy of security
- Prototyping
  - Architecture of the (Conditional Access System + Strategy of security)
  - FPGA prototype
  - Validation
- Conclusions and perspectives

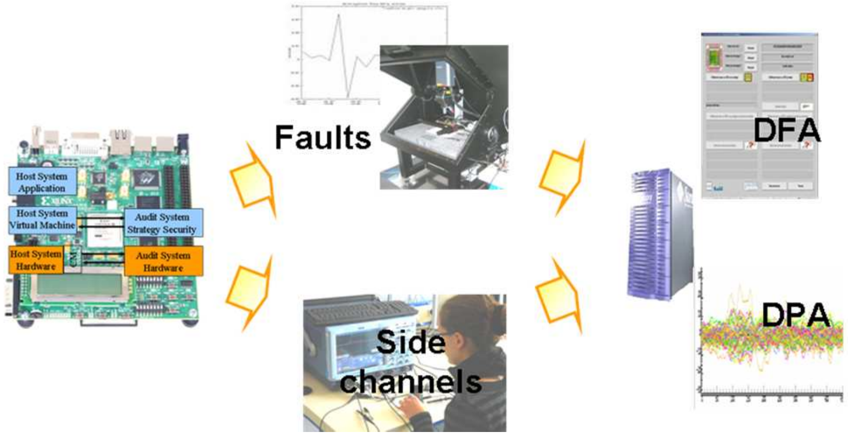
CONCLUSION

Our work constitutes a first step towards the implementation of complex strategies of security

- Re-organization of security features thought the entire system
- Proposal of an architecture enabling the execution of complex strategies of security
- Innovative strategy of security based on fuzzy logic
- Set up of a dedicated HW/SW design methodology (including debugging tools and built-in security estimation capabilities)

FUTURE WORKS

- Fine tuning of the current rules set
- Security characterization of the prototype with ENSMSE-CMP benches at Gardanne



Distinguish “normal functioning” and “attack”

==

MODEL USER **AND** ATTACKER

⇒ Which formalism ?

⇒ Data bases of attacker and user behavior & learning algorithms?

⇒ Are the current sensors suitable?

⇒ etc…

Thank you for your attention!

Questions?