Le Hameau de l'Etoile ,
Wednesday, June 20th 2007

Netheos

Reconfigurable cryptographic processors on FPGA

Managers :
Lionel Torres (LIRMM) , Xavier Facelina (NETHEOS)

Authors :
Benoît Badrignans , Augustin Sarr , Jerome Di Battista , Guillaume Denis

UNIVERSITÉ MONTPELLIER 2
SCIENCES ET TECHNIQUES

LIRMM

- Introduction
- Reconfiguration advantages
- Physical security
- Applications
- Conclusion and perspectives

Research organization

- 2 PhD agreements with LIRMM and I3M (UM2)
- 1 ANR research program : ICTeR

  « Integrity and Confidentiality of Reconfigurable Technologies »

TELECOM PARIS
école nationale supérieure des télécommunications

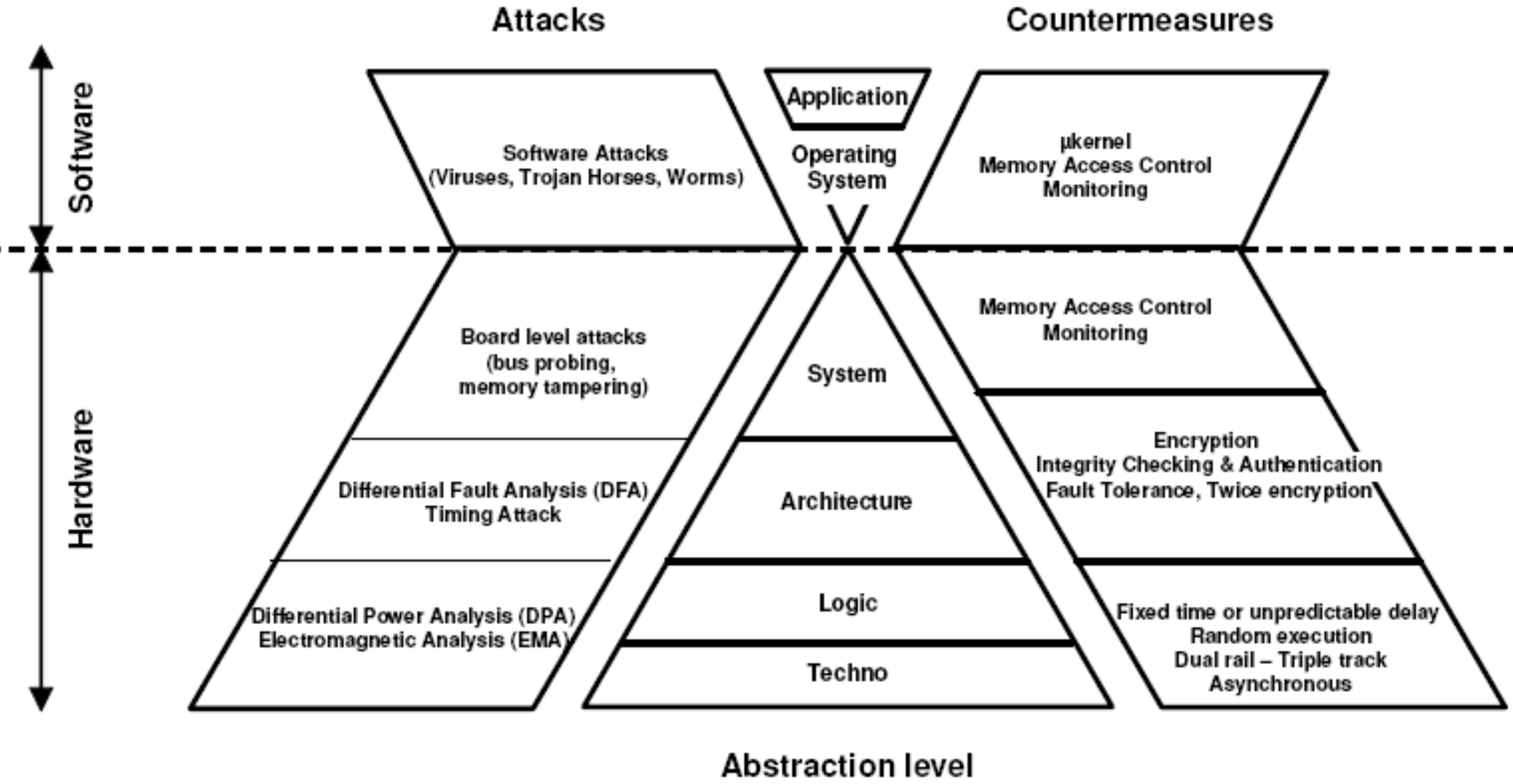
L.E.S.T.E.R
Laboratoire d'Electronique des Systèmes TEmps Réel

LIRMM

LABORATOIRE Hubert CURIEN
UMR • CNRS • 5516 • ST-ÉTIENNE

Netheos
CONFIDENTIALITÉ & SÛRETÉ NUMÉRIQUE

ICTeR « Integrity and Confidentiality of Reconfigurable Technologies »

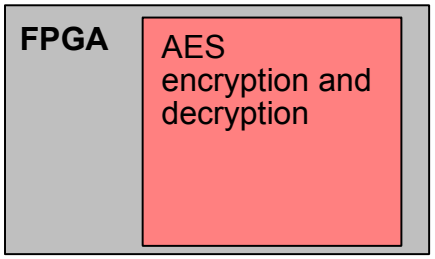


Performance improvement

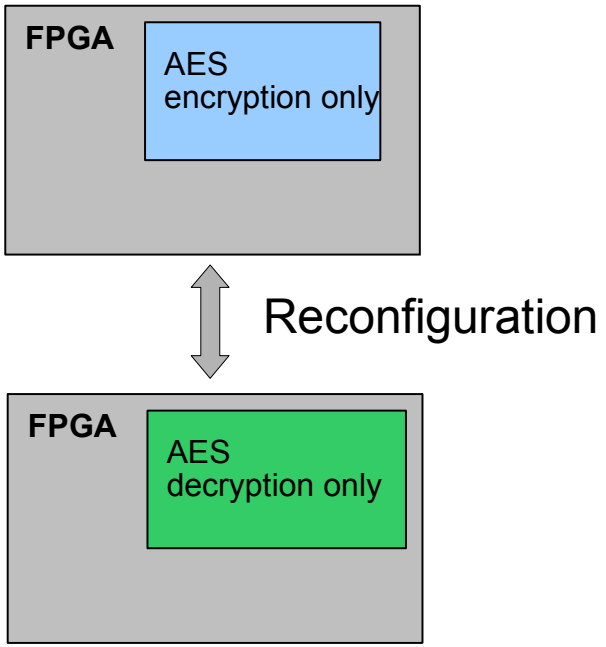
Example : AES implementation on Spartan3

	Max frequency	Area requirements
Encryption and decryption	60 Mhz	1200 Slices
Encryption only	80 Mhz	800 Slices
Decryption only	75 Mhz	1000 Slices

Static implementation



Dynamic implementation



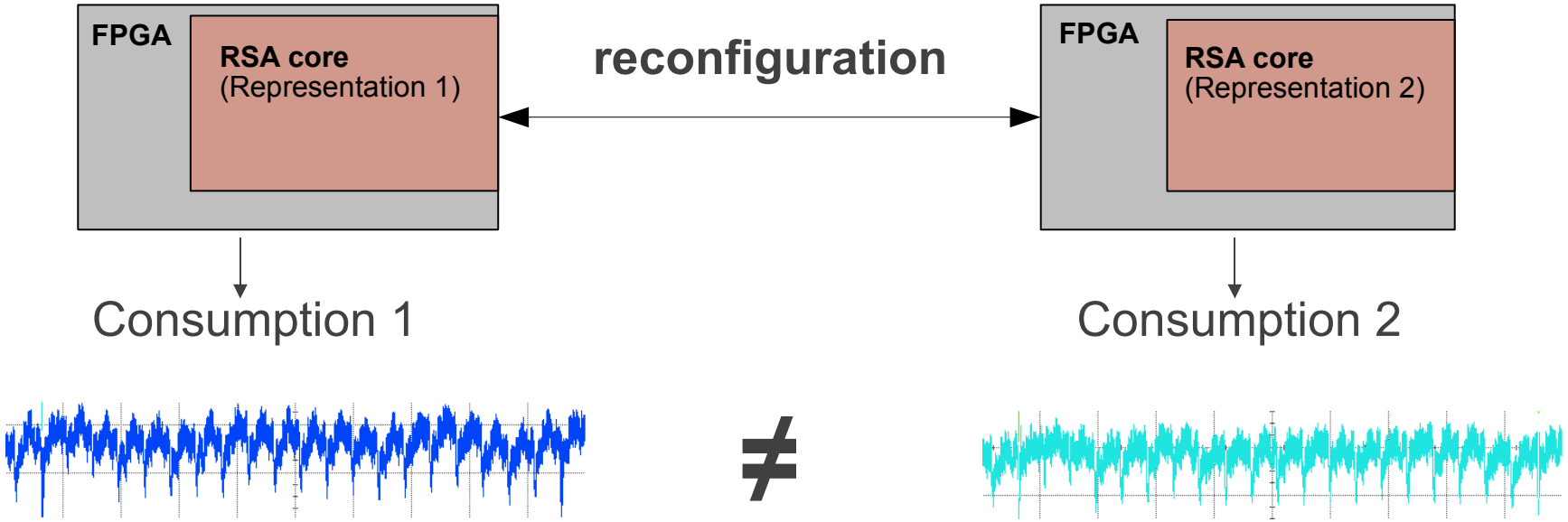
Security improvement

(see Cryptarchi'06)

Example : RSA security

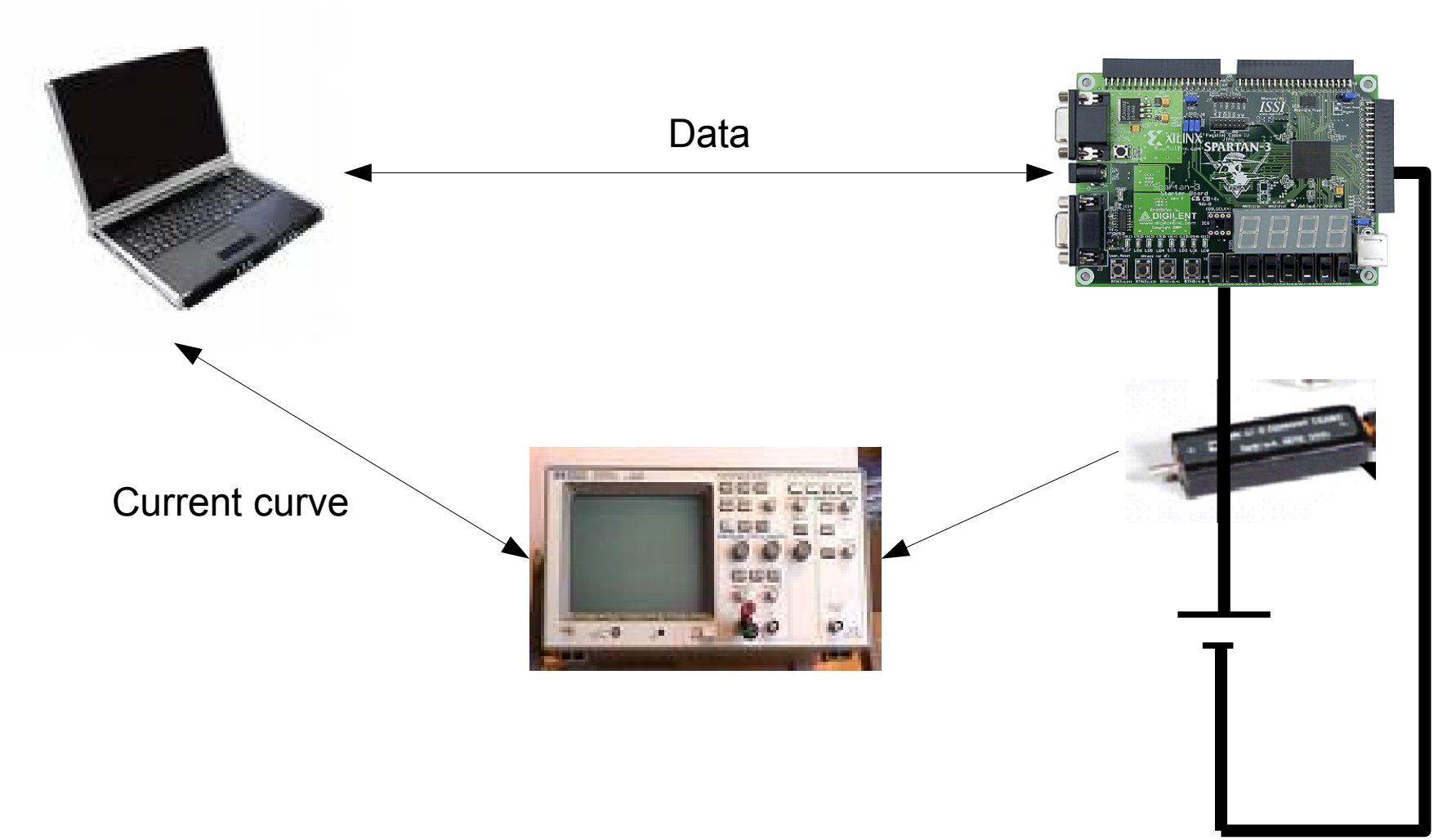
Idea : Reduce correlation between data and power consumption

Principle : Modification of number representation (Residue Number System)



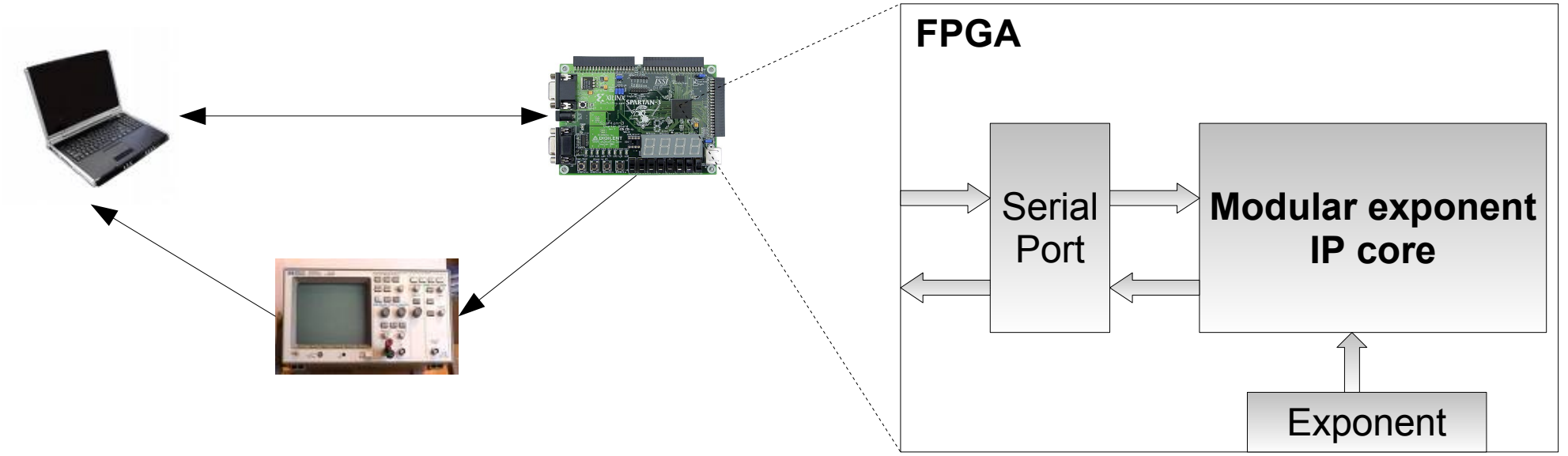
➡ Attack platform needed

Attack platform for FPGA



SPA on RSA

(modular exponent IP)



SPA on RSA

(modular exponent IP)

Data :

message : M ;
exponent : E ;
modulus : N

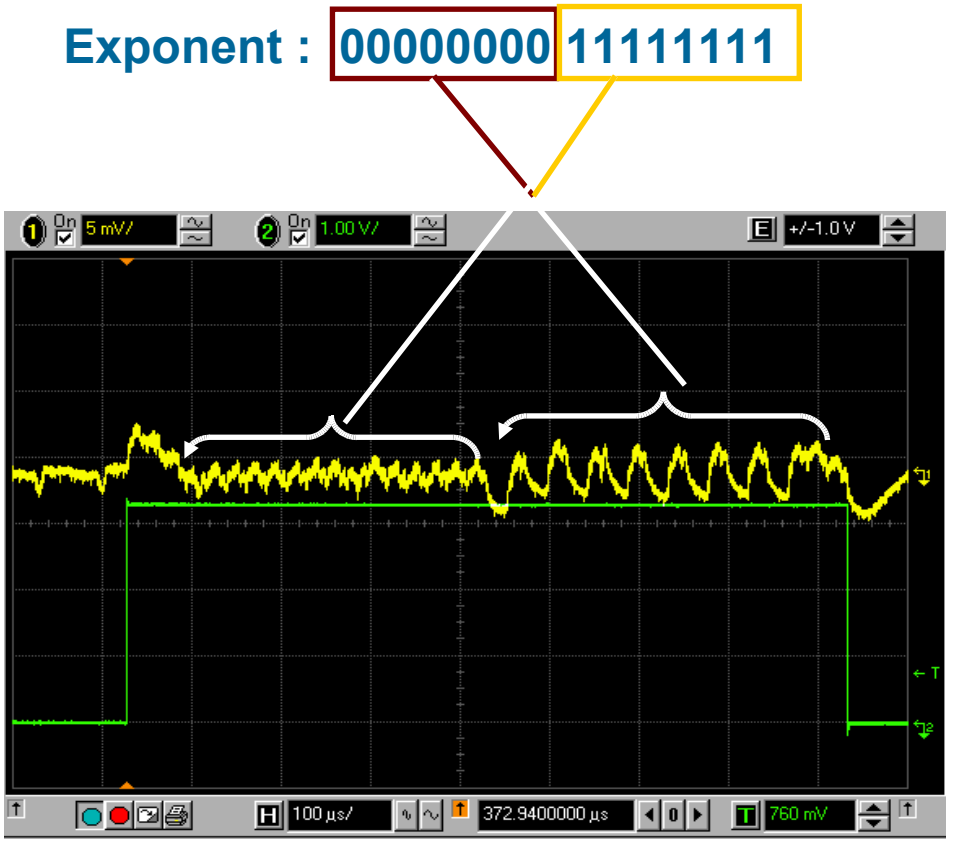
Notation :

exponent size : L ;
i^{th} bit of E : e_i

Result : $R = M^E \bmod N$

Square and multiply Algorithm :

1. R=1 ; P=M
2. for i=0 to L-1
   - 2.2 P=P.P mod N (square)
   - 2.3 if e_i =1 then
   - 2.4 R = R.P mod N (multiply)
   - 2.5 else
   - 2.6 wait; (nothing)
   - 2.5 end if;
3. end for;



SPA on RSA

(modular exponent IP)

Data :

message : M ;
exponent : E ;
modulus : N

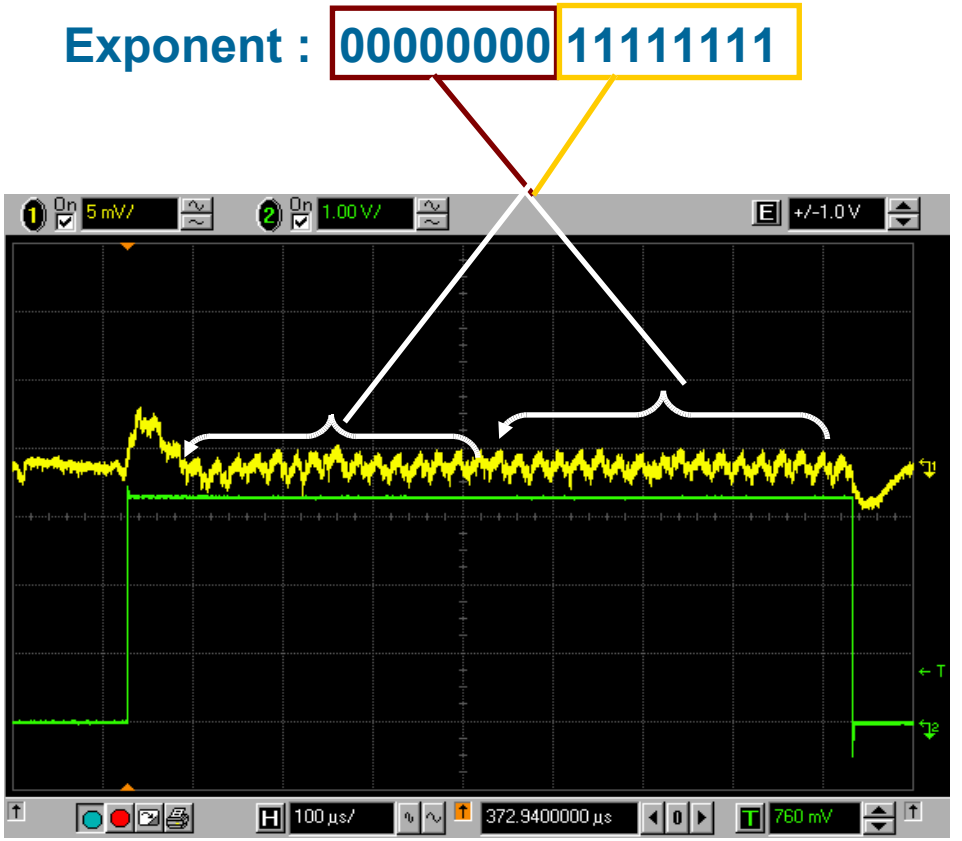
Notation :

exponent size : L ;
i^{th} bit of E : e_i

Result : $R = M^E \bmod N$

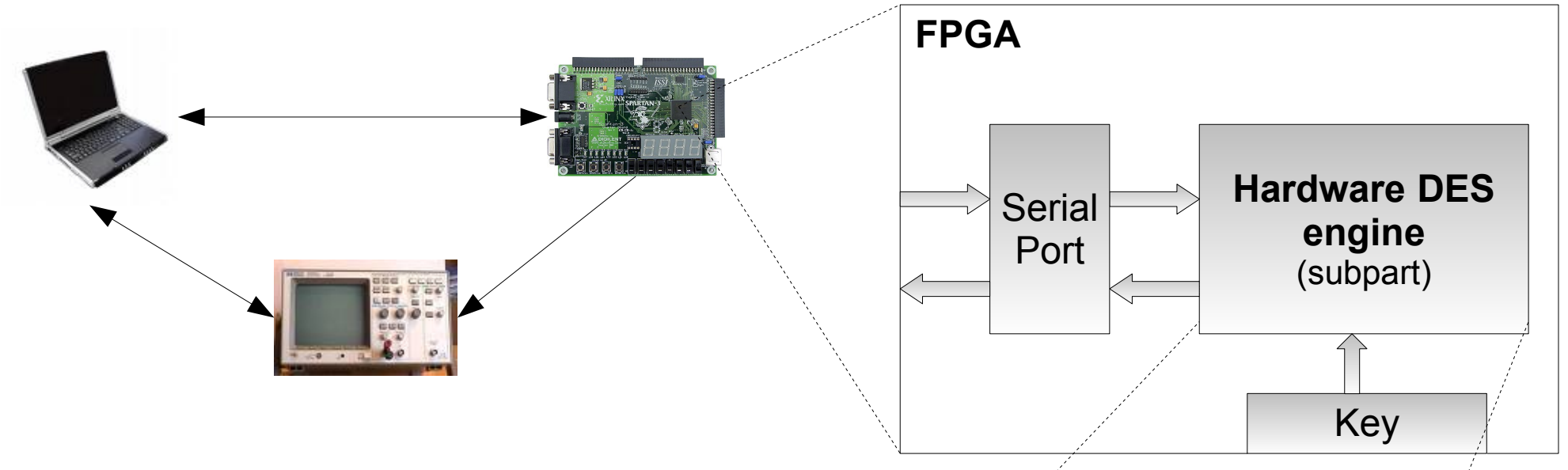
Square and multiply Algorithm :

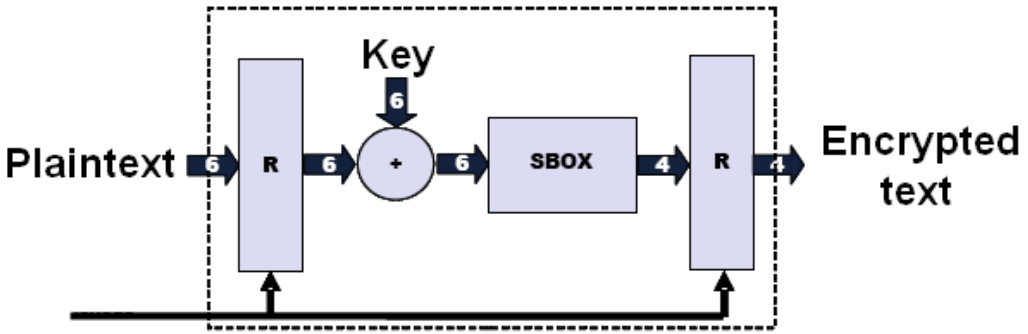
1. R=1 ; P=M
2. for i=0 to L-1
   2.2 P=P.P mod N (square)
   2.3 if e_i =1 then
   2.4 R = R.P mod N (multiply)
   2.5 else
   2.6 X = R.P mod N (dummy multiply)
   2.5 end if;
3. end for;



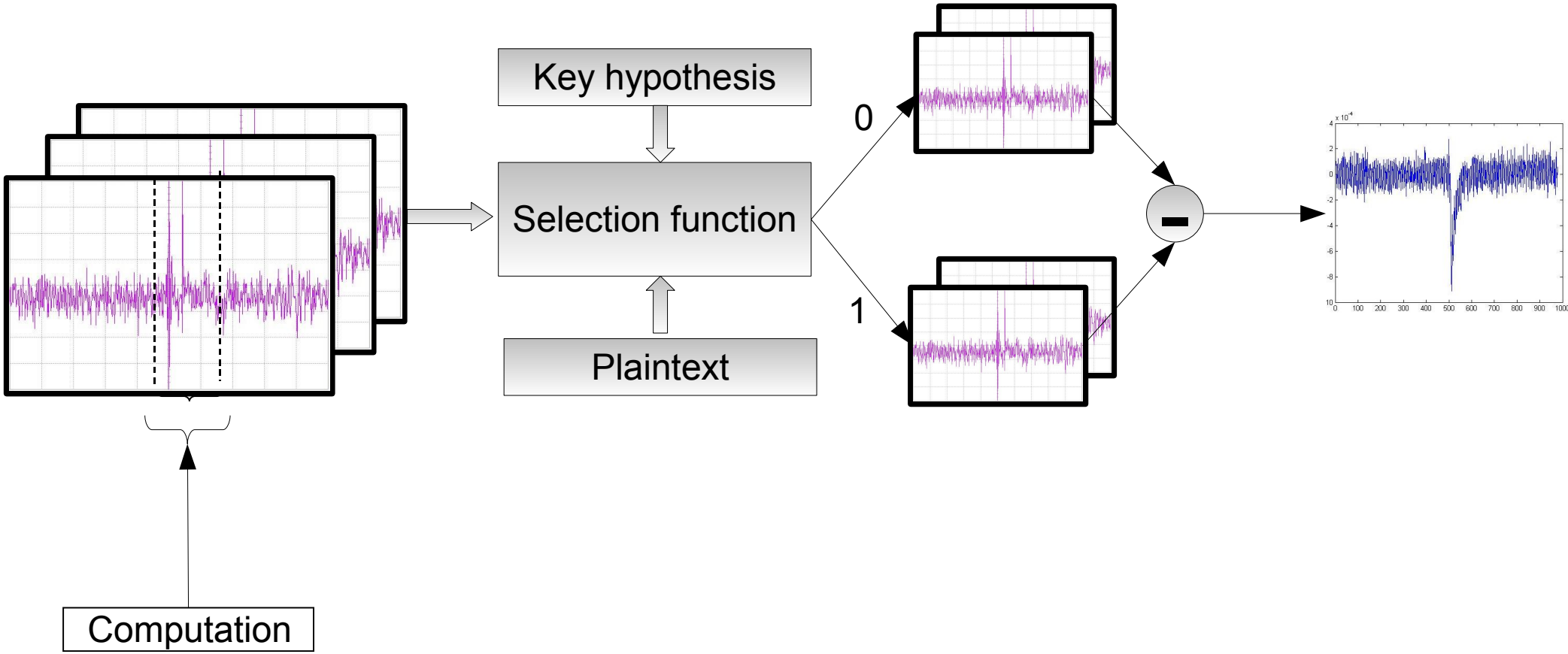
DPA on DES

(only a subpart)

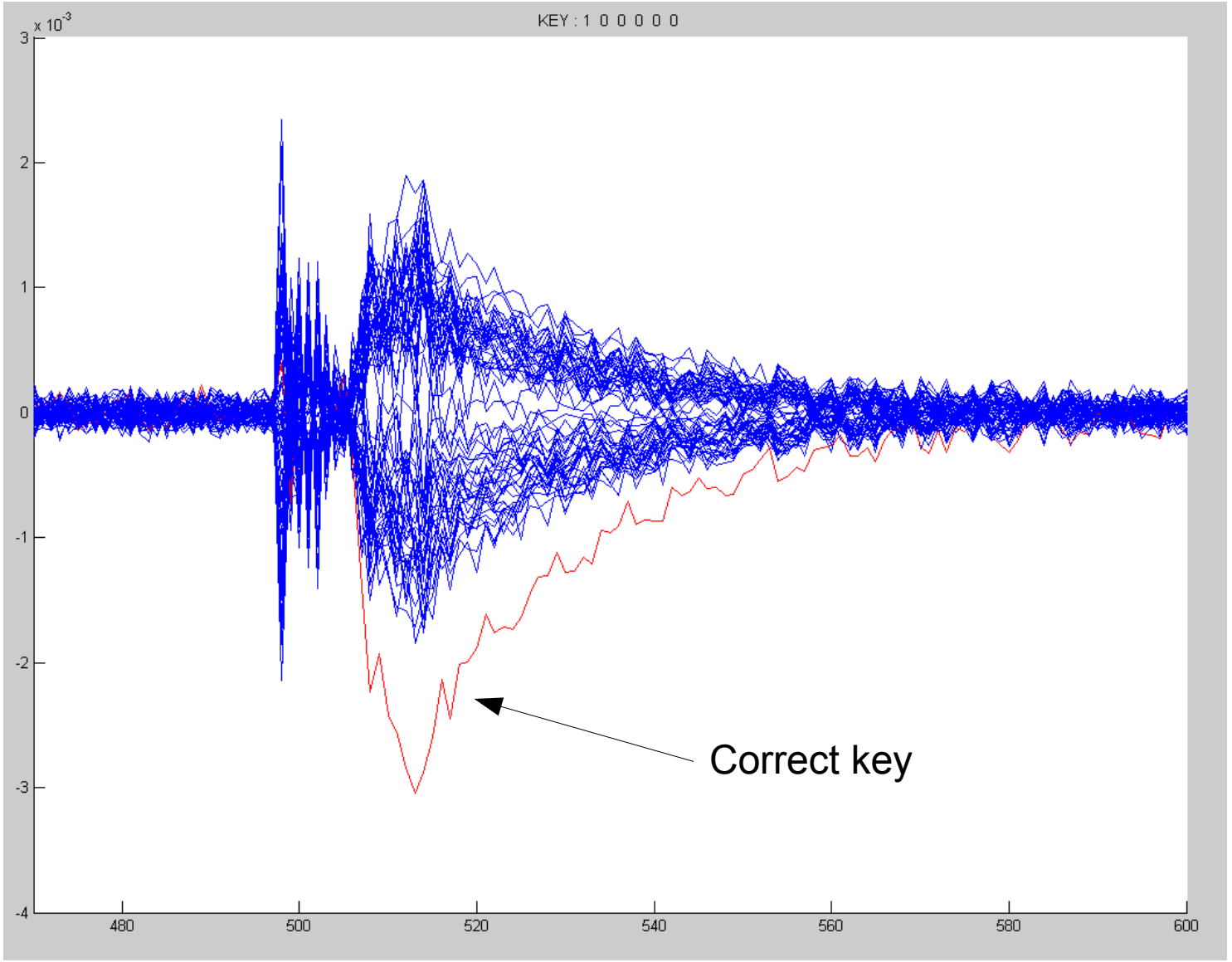




DPA on DES


Key hypothesis
Selection function
Plaintext
Computation
0
1

DPA on DES



Obtained with 1000 samples

Attack platform perspectives

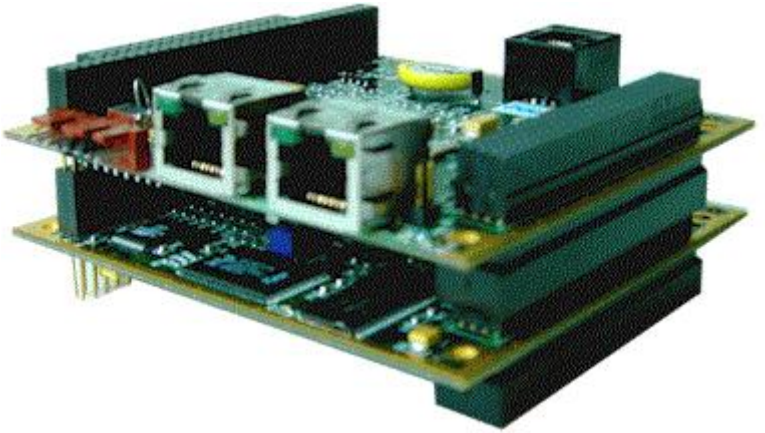
- Extension to other algorithms
- Counter-measures testing
- Comparison with ASIC

Applications

Hardware Security Module (HSM)

Crypto accelerator

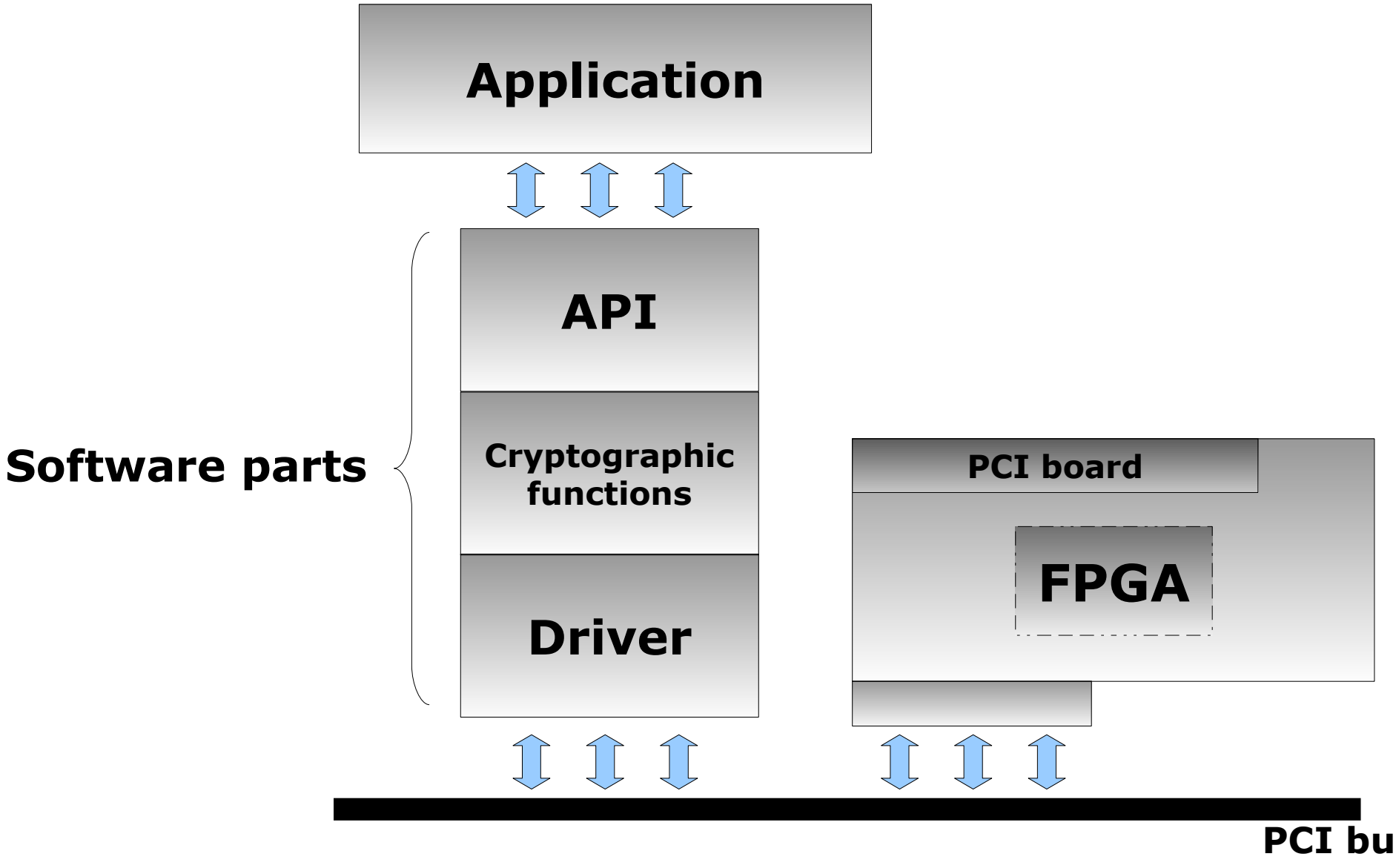
Hardware disk encryption



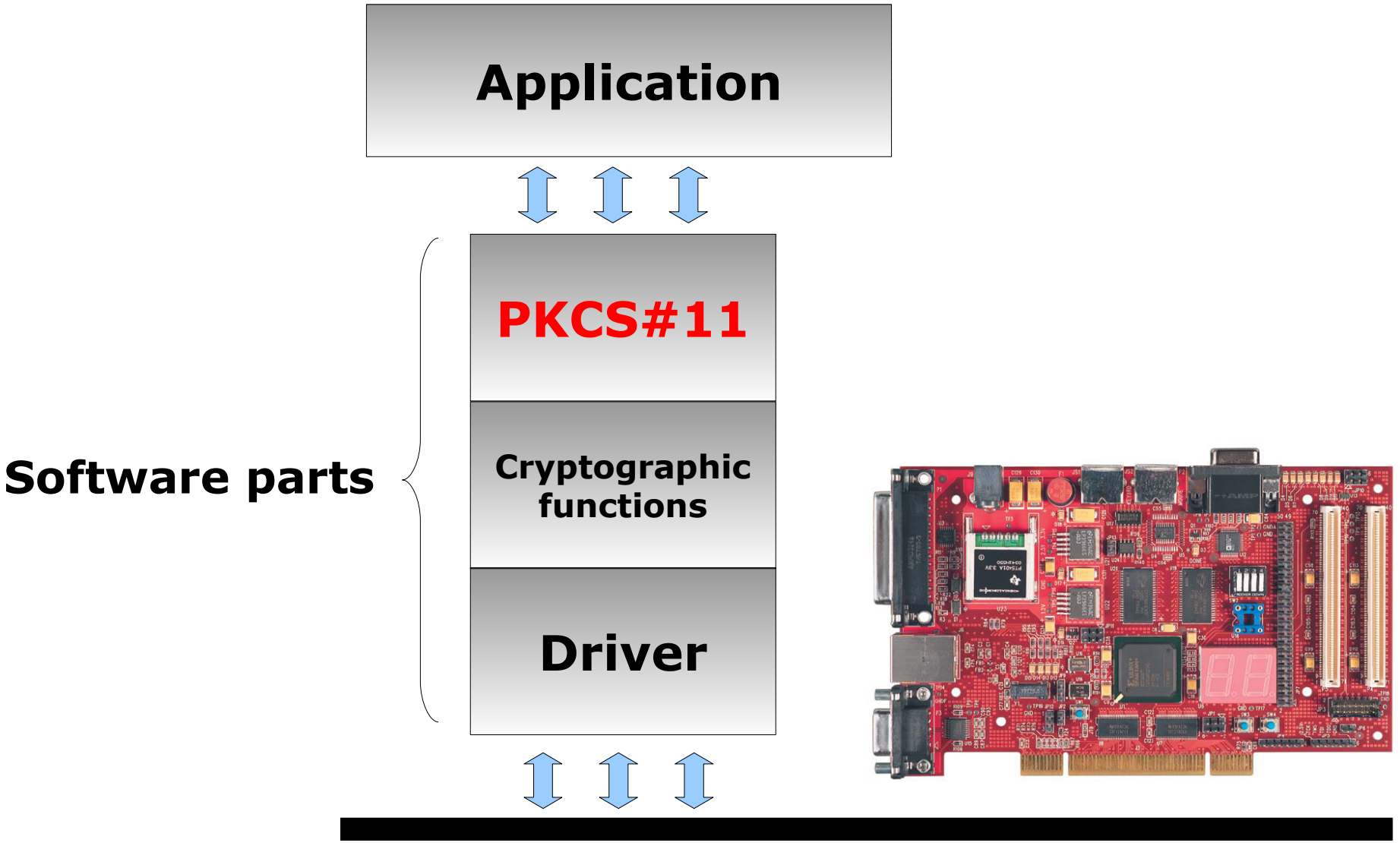
Network encryption

➔Maybe custom products

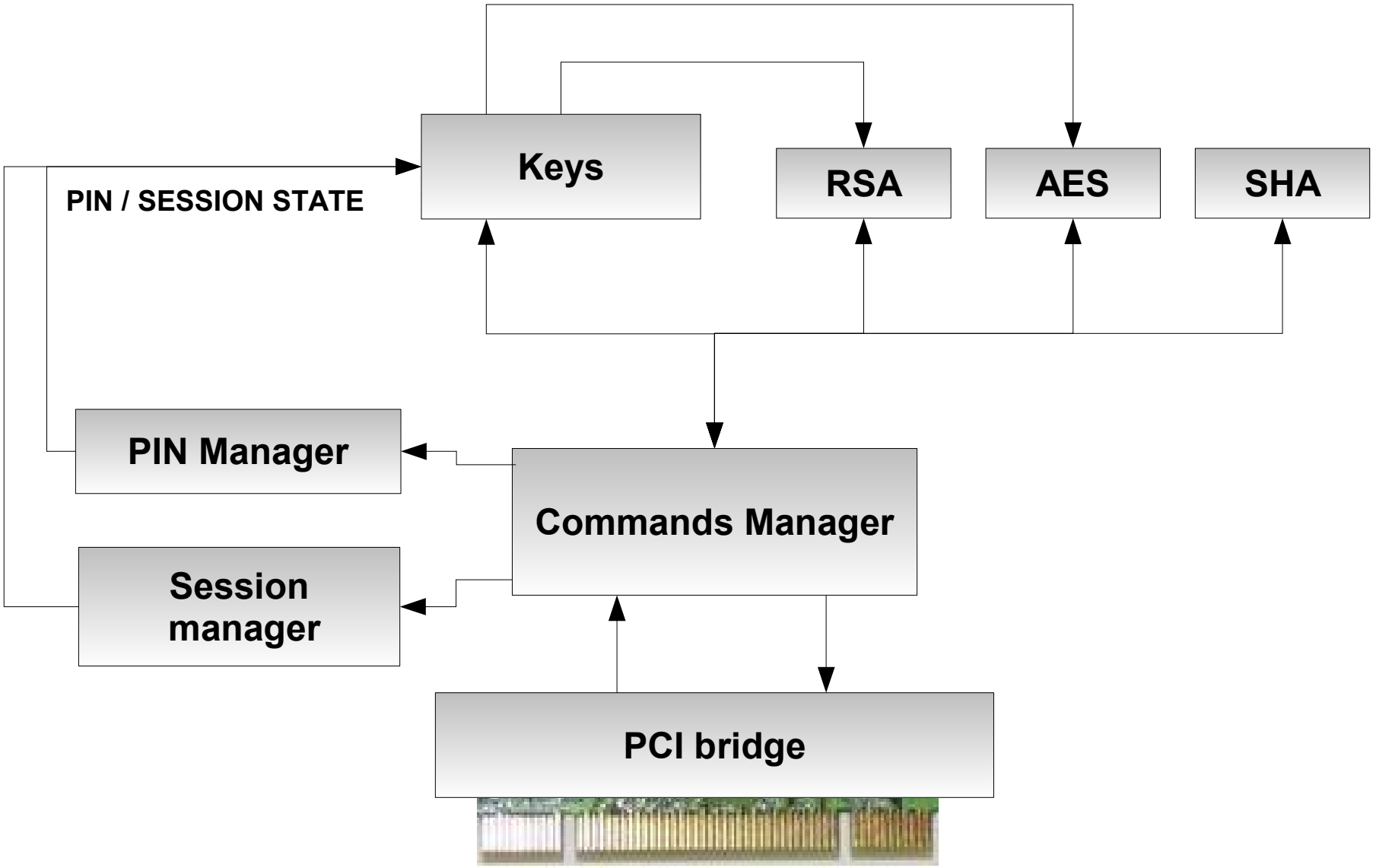
HSM prototype



HSM prototype



HSM prototype



Why FPGA ?

- Small volume markets and custom products ;
- Efficient cryptographic core could be developed ;
- Reconfigurable → hardware update ;
- Short design cycle → prototypes and experiments ;
- Dynamic reconfiguration → counter-measures and area optimization

But ...

- Hardware functions are limited (no true RNG) ;
- Investigations on physical security needed ;
- Security features are limited.