Physical Attacks and Beyond

Francesco Regazzoni

Contents

What Are Physical Attacks

Design of the Algorithm

What Are Physical Attacks

Design of the Algorithm ‖ Implementation

What Are Physical Attacks

Design of the Algorithm $\|$ Implementation

Physical attacks recover secrets by exploiting the implementation

Types of Physical Attacks

Active	**Passive**
Fault Injection	Power Analysis
	Timing Analysis

Side Channels Are Used in Many Fields

- Pizza Delivery
- Energy Consumption
- Biology
- ...
- Cryptography

Why Physical Security is so Important Today?

Computational Unity

Long Time Ago	**Past**	**Present**
Mainframes	Personal Computer	Pervasive

Physical Access

Mainframes	**Personal Computer**	**Pervasive**
Almost Impossible Physical Access	In a Relatively Protected Environment	Potentially in the Hand of the Attacker

Contents

Two Main Directions...

Countermeasures ‖ Better Attacks

Time line for Attacks

- 1996 Timing Attacks

Paul C. Kocher, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. CRYPTO 1996

Time line for Attacks

- 1996 Timing Attacks
- 1997 Fault Injection Attacks

D. Boneh, R. A. DeMillo, R. J. Lipton: On the Importance of Checking Cryptographic Protocols for Faults. EUROCRYPT 1997

Time line for Attacks

- 1996 Timing Attacks
- 1997 Fault Injection Attacks
- 1999 Power Analysis Attacks

Paul C. Kocher, Joshua Jaffe, Benjamin Jun Differential Power Analysis. CRYPTO 1999

Time line for Attacks

- 1996 Timing Attacks
- 1997 Fault Injection Attacks
- 1999 Power Analysis Attacks
- 2002 Electromagnetic Attacks

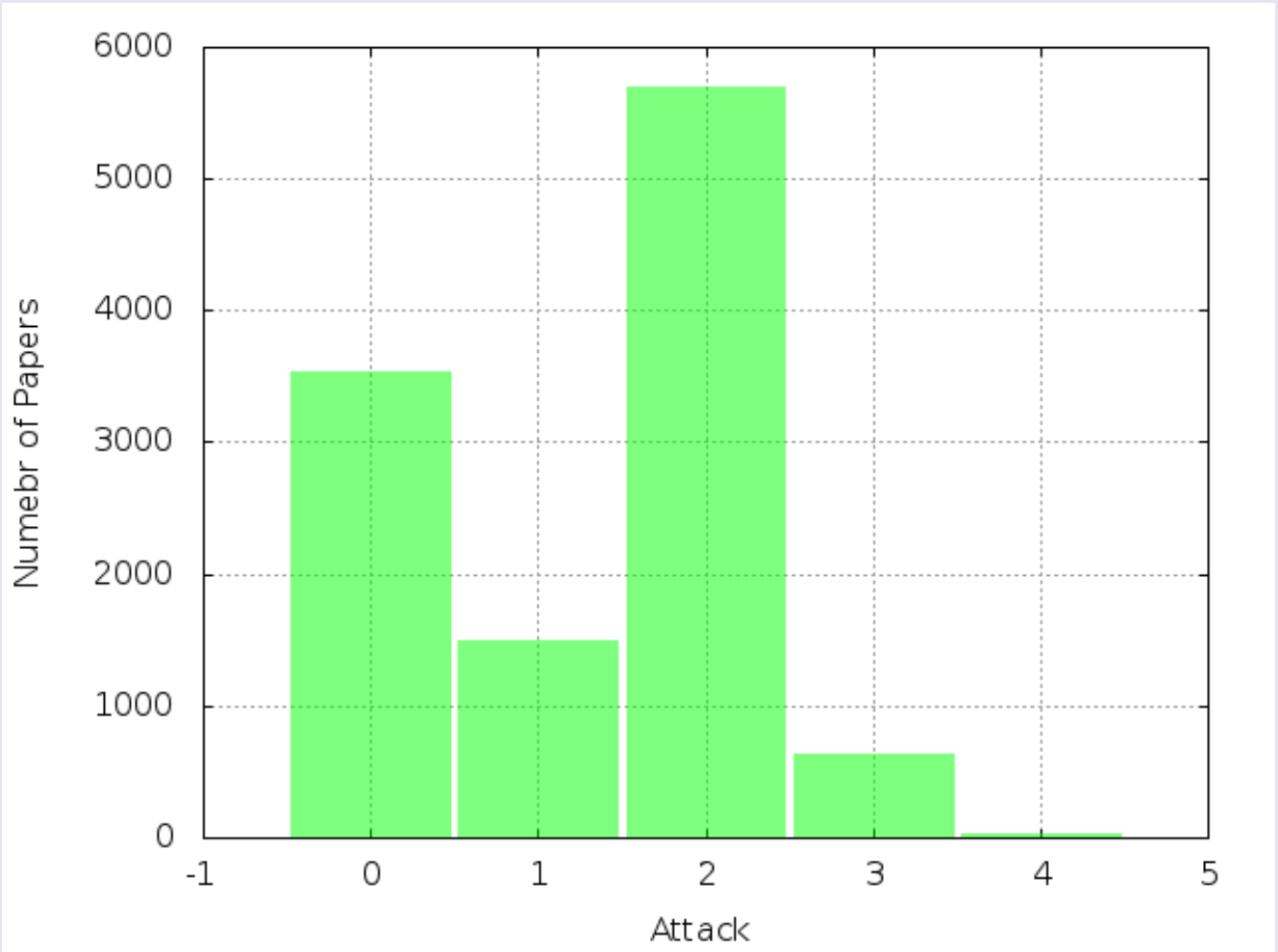
D. Agrawal, B. Archambeault, J. R. Rao, P. Rohatgi: The EM Side-Channel(s). CHES 2002

Time line for Attacks

- 1996 Timing Attacks
- 1997 Fault Injection Attacks
- 1999 Power Analysis Attacks
- 2002 Electromagnetic Attacks
- 2012 Photon Emission

A. Schlosser, D. Nedospasov, J. Kramer, S. Orlic, J. P. Seifert: Simple Photonic Emission Analysis of AES. CHES 2012

Research Activity per Attack (approx)



Highlight on Countermeasures

- Timing Attacks Constant Time
- Masking
- Protected Logic Styles
- Metrics for Comparison
- Redundancy Against Fault Attacks

Contents

Challenge One

What is the Attacker Goal

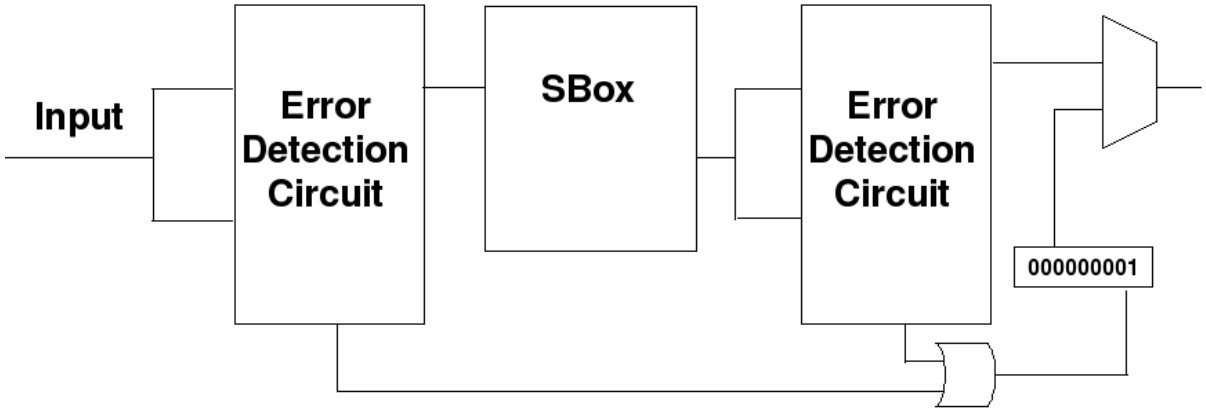
To Access Secret Information

Types of Physical Attacks

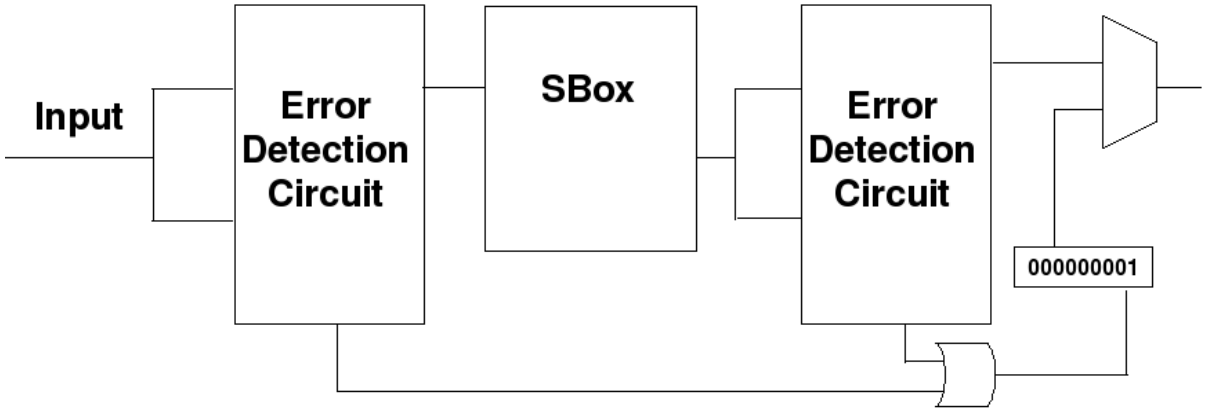
Active	**Passive**
Fault Injection	Power Analysis
	Timing Analysis

Physical attacks are considered independently!

Effects of Error Correcting Codes on DPA

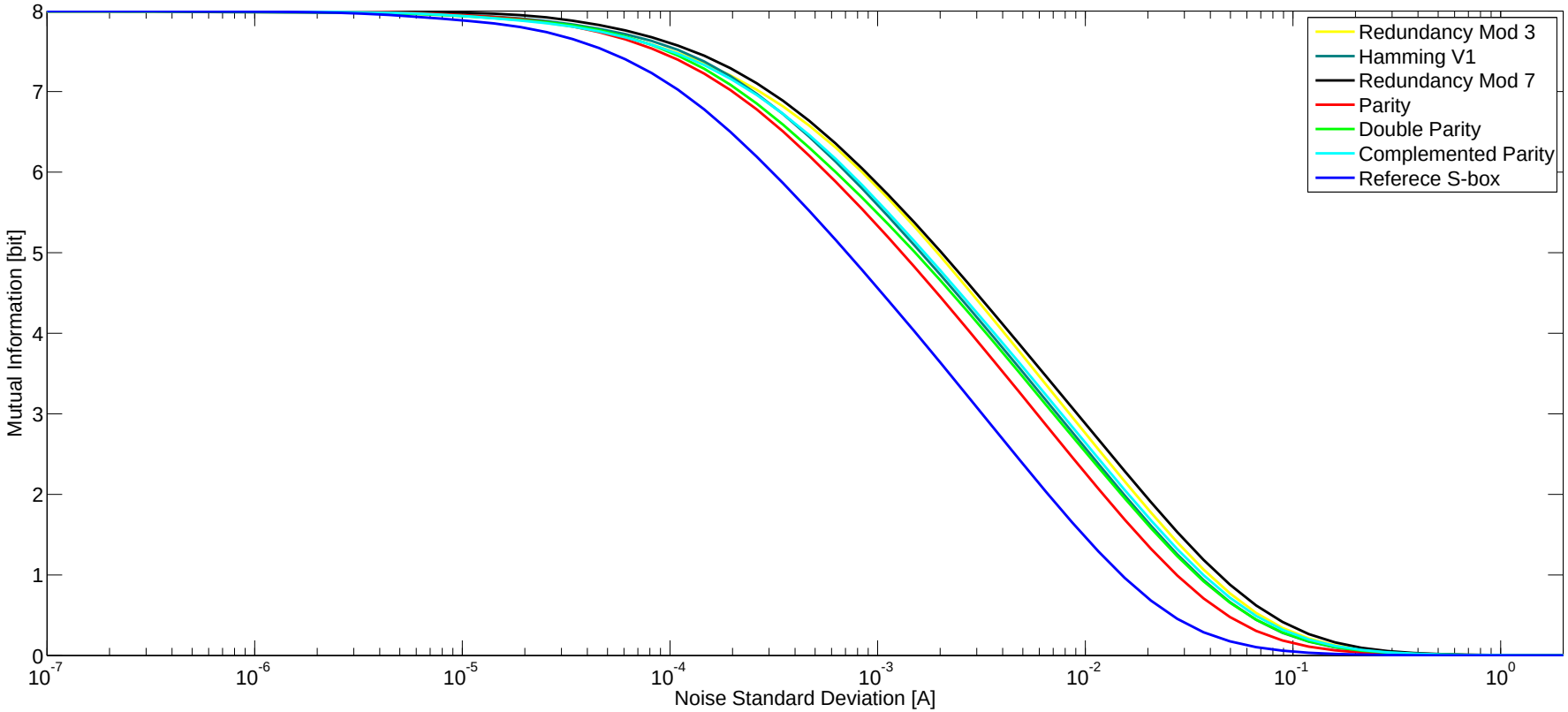


Effects of Error Correcting Codes on DPA

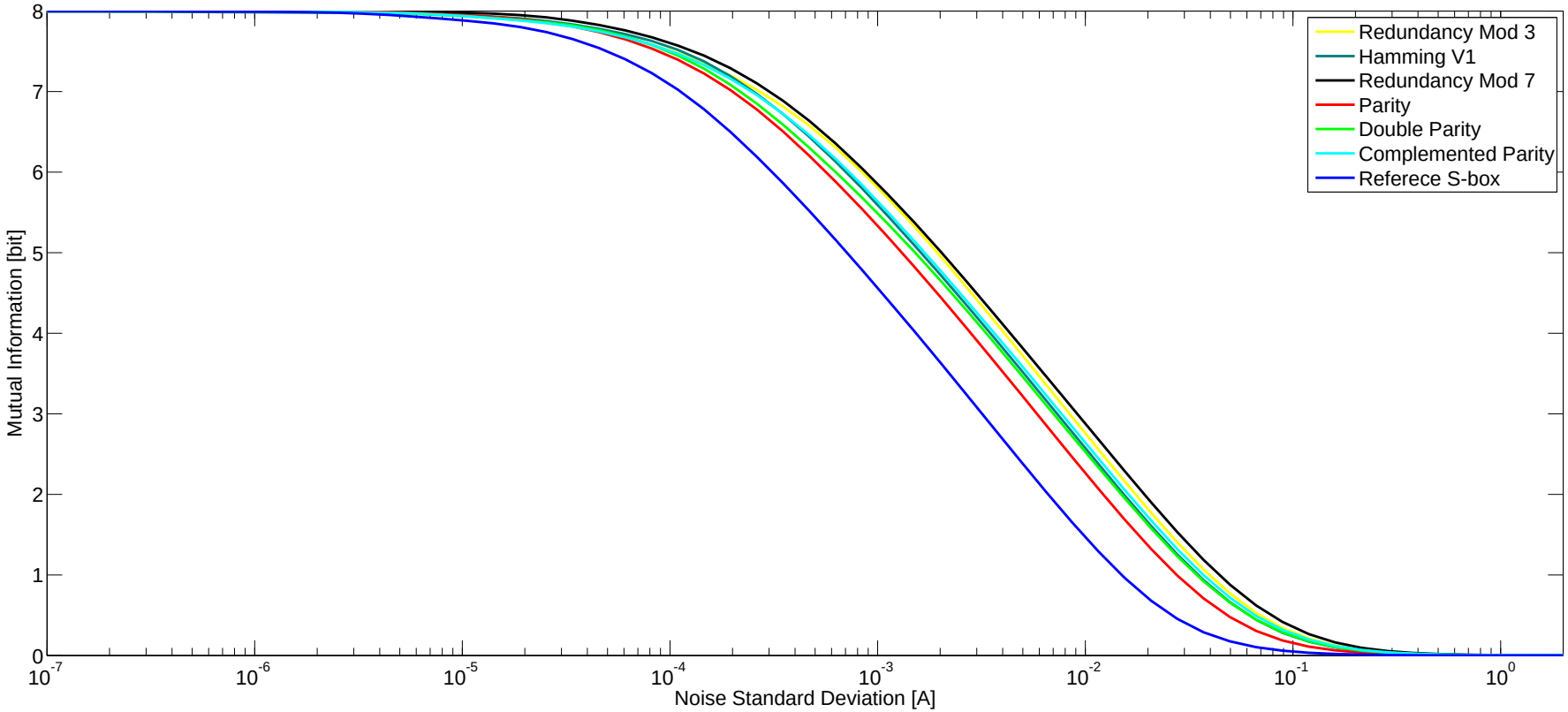


- Reference
- Parity
- Complemented Parity
- Double Parity
- Residue Modulo 3
- Residue Modulo 7
- Hamming Code

Error Correcting Code



Error Correcting Code



I am helping the DPA attacker!

Challenge One Summary

We know very little about interactions of countermeasures!

Challenge Two

What is the Designer Goal?

To Secure the Chip

Towards Automation

Inputs:

- Unprotected Algorithm
- Countermeasure

Output:

- Algorithm where the countermeasure is Applied

- Algorithm where the countermeasure is applied does NOT mean protected Algorithm

Do We Need Automation?

Do We Need Automation?

Manual Design ‖

Do We Need Automation?

Manual Design ‖ Simple Chips

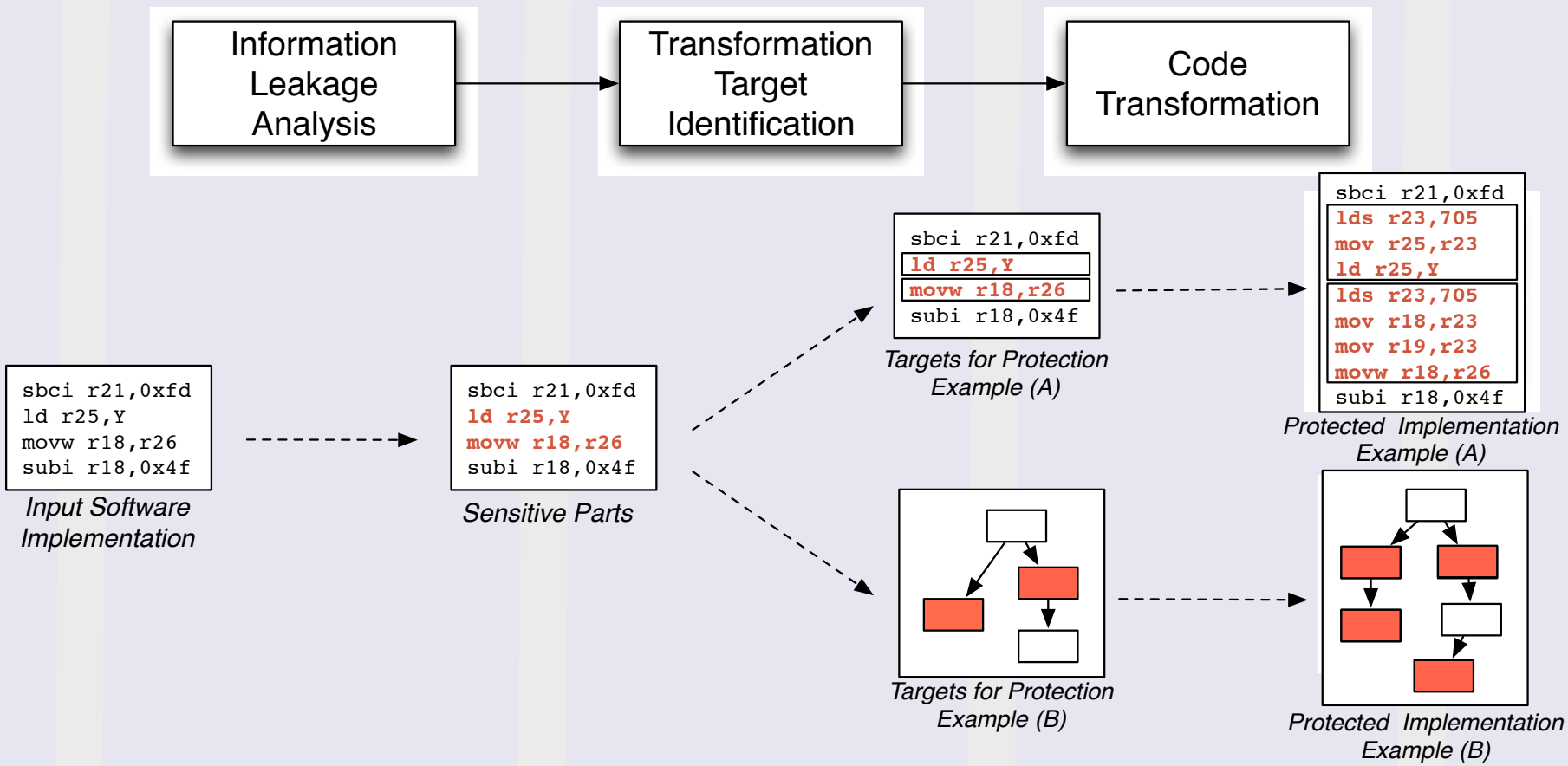
Do We Need Automation?

Manual Design EDA Tools	Simple Chips

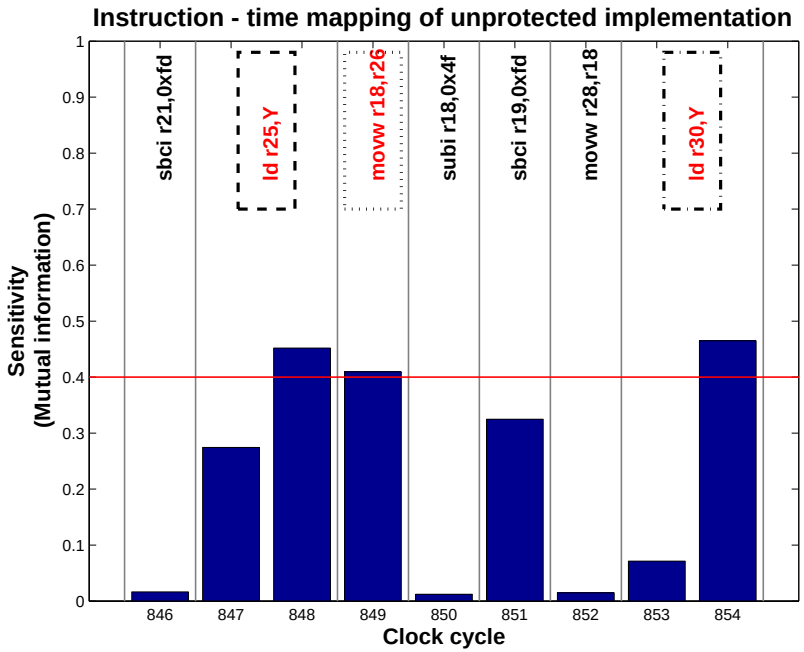
Do We Need Automation?

Manual Design	Simple Chips
EDA Tools	Complex Chips

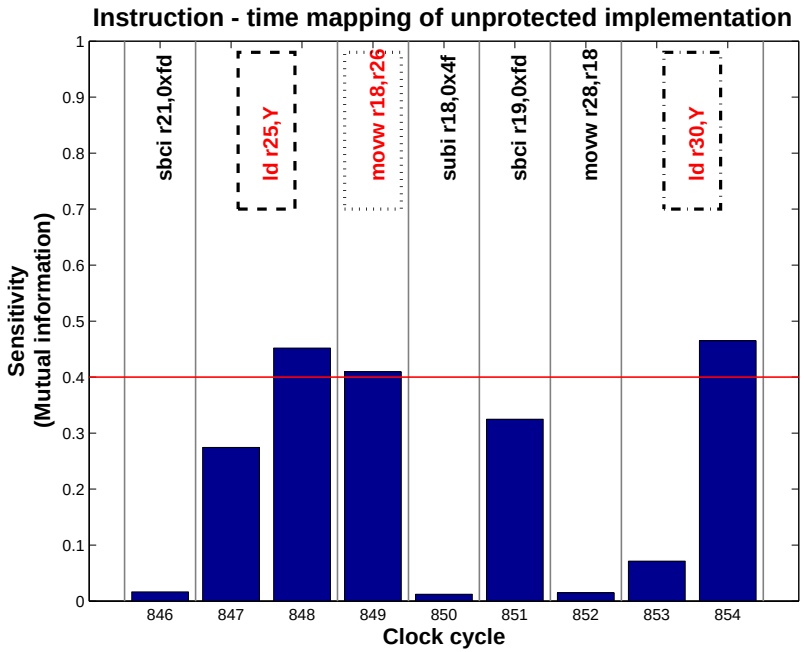
Example, Software Automation

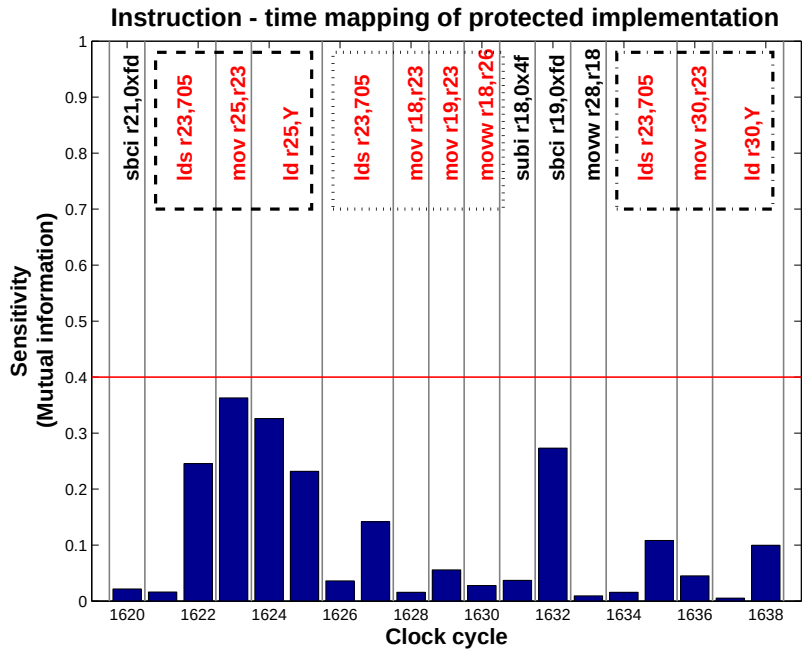


Example, Software Automation



Example, Software Automation





What is the Designer Goal?

To Secure the Chip

Towards Automation

Inputs:

- Algorithm where the countermeasure is Applied
- Countermeasure

Output:

- Assertion of the Correct Application of the Countermeasure

- Assertion of the correct application of the countermeasure does NOT mean protected Algorithm

Do We Need Verification?

```
void maskedARK() {
  unsigned char i;
  for (i=0;i<16;i++){
    st[i] = pt[i] ^
      (key[i] ^ mask[i]);
  }
}
```

avr-gcc-4.5.3 -O3

```
        .text
.global ARK
        .type   ARK, @function
ARK:
/* prologue: function */
/* frame size = 0 */
/* stack size = 0 */
.L__stack_usage = 0
        lds r24,key
        lds r25,pt
        eor r24,r25
        lds r25,mask
        eor r24,r25
        sts st,r24
        lds r24,key+1
        lds r25,pt+1
        eor r24,r25
        ...
```

What can be Done for Verification

- Represent the program as a graph
- Use satisfiability queries to detect the dependencies and sensitivity

What can be Done for Verification

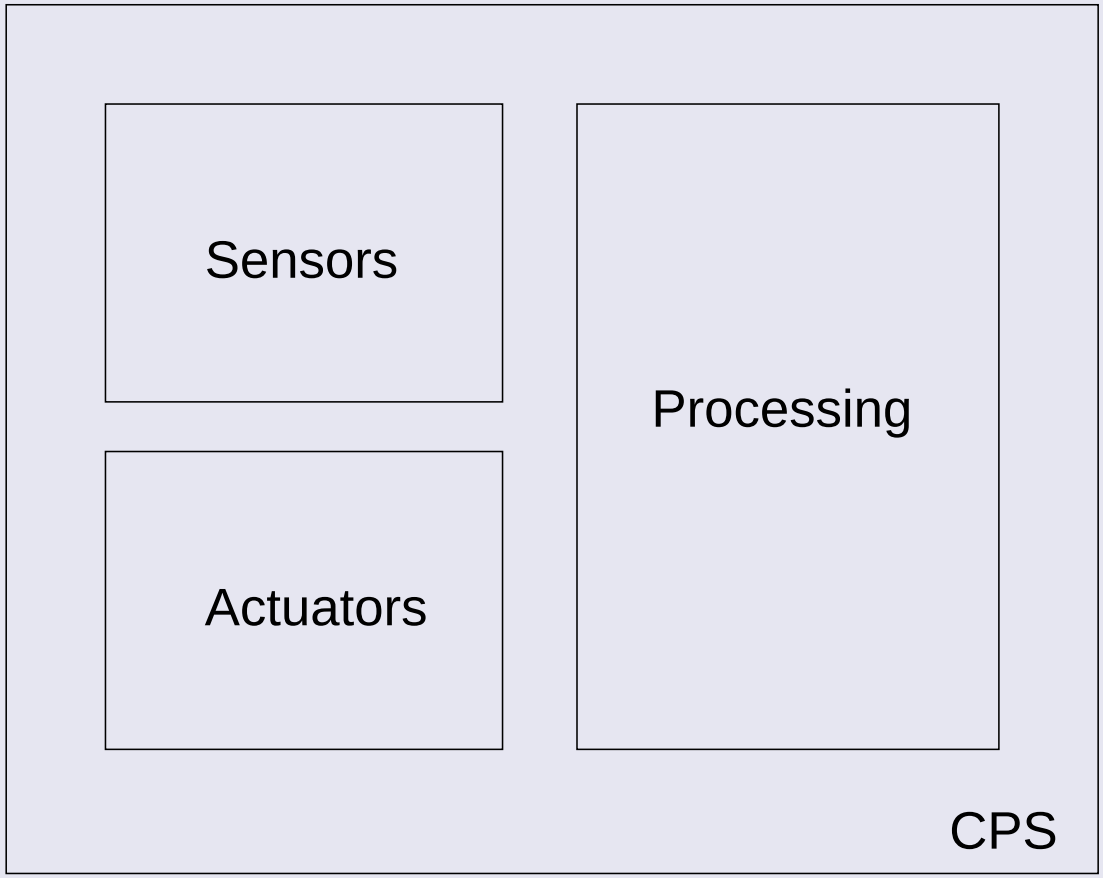
- Compiler related problems
- Programmer related problems
- Countermeasure related problem

Challenge Two Summary

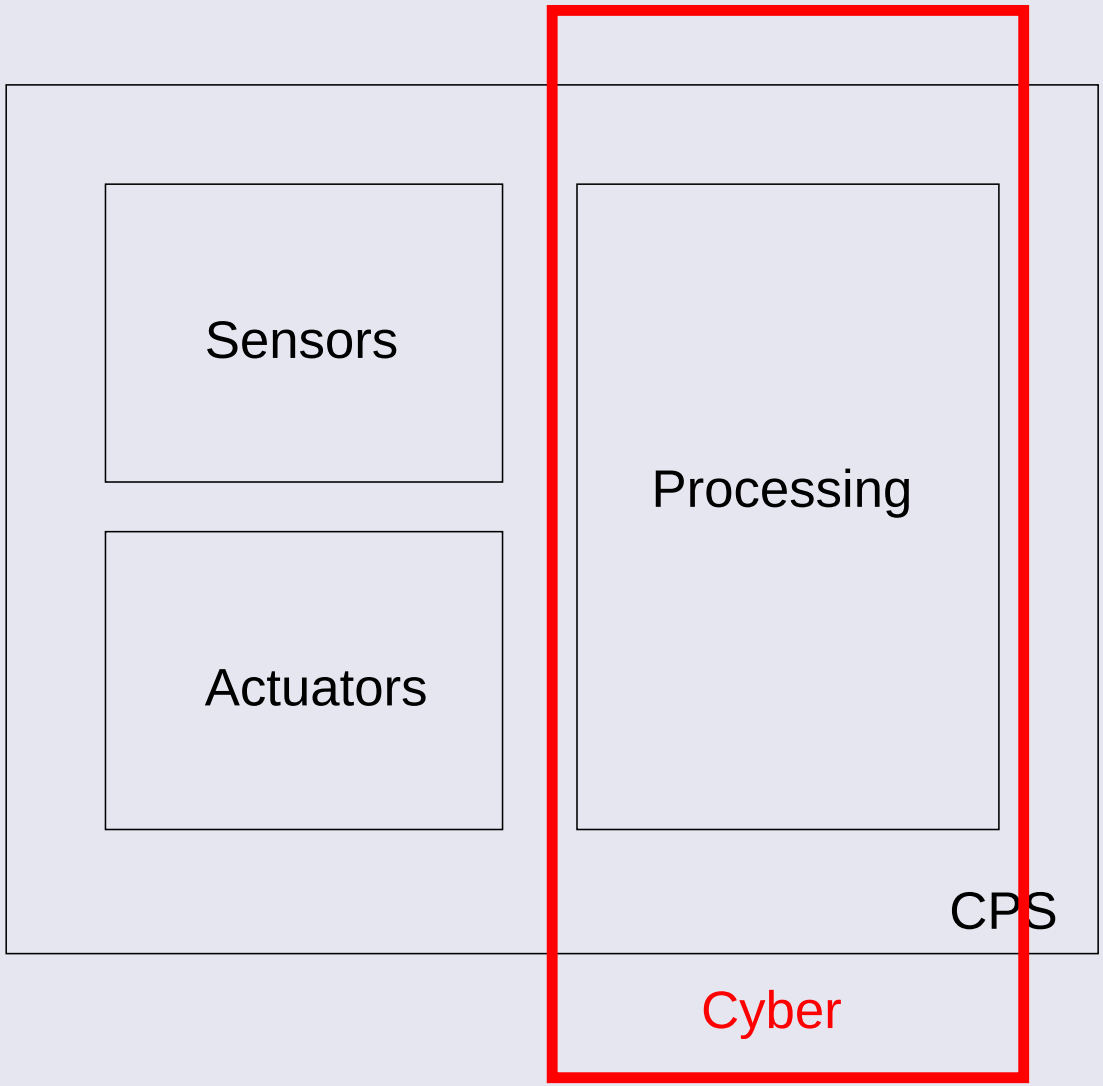
We need an automated infrastructure supporting security

Challenge Three

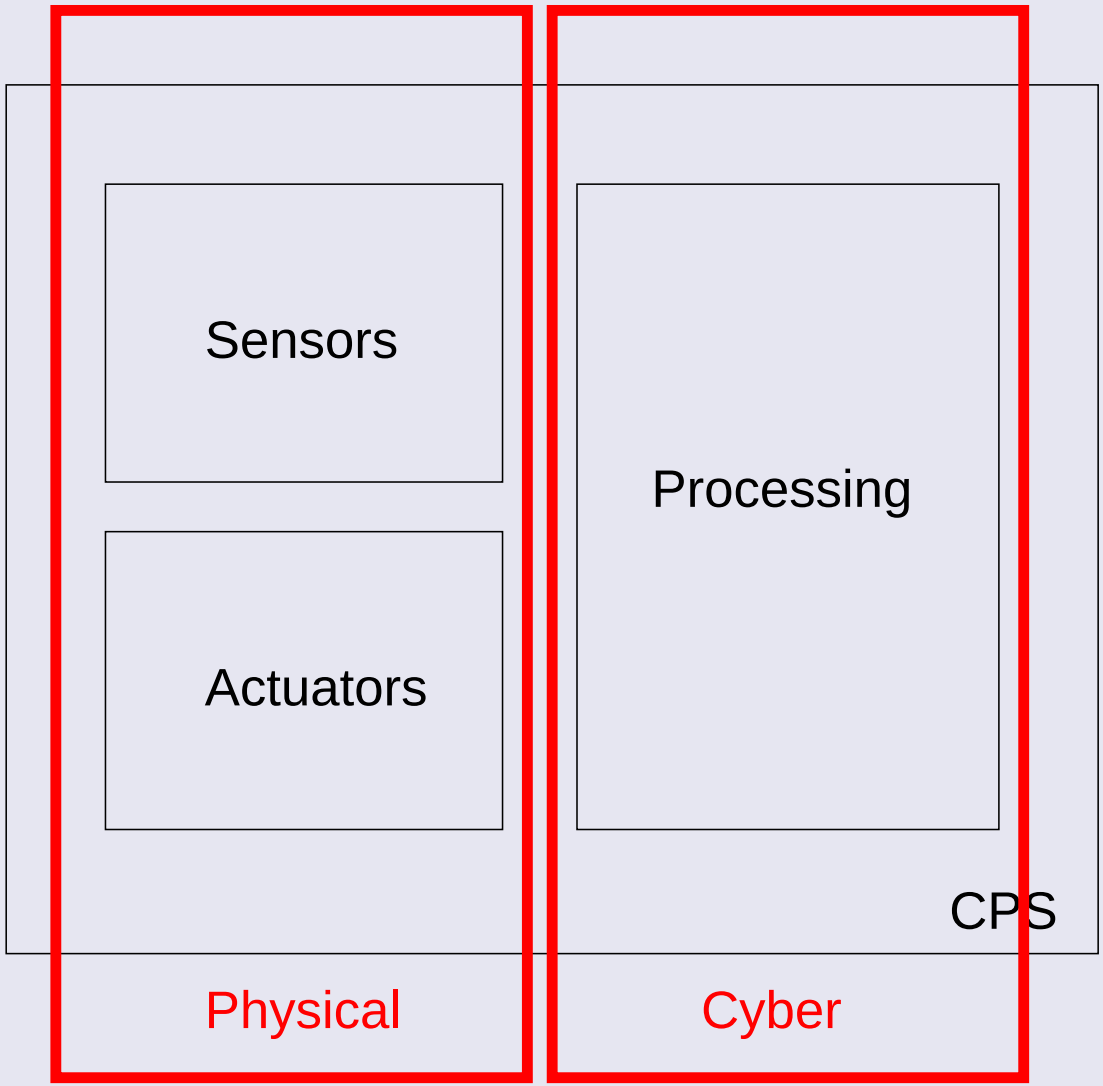
Cyber-Physical Systems



Cyber-Physical Systems



Cyber-Physical Systems



Same Challenges?

Physical Leakage from Physical Components

Same Challenges?

Car Opening

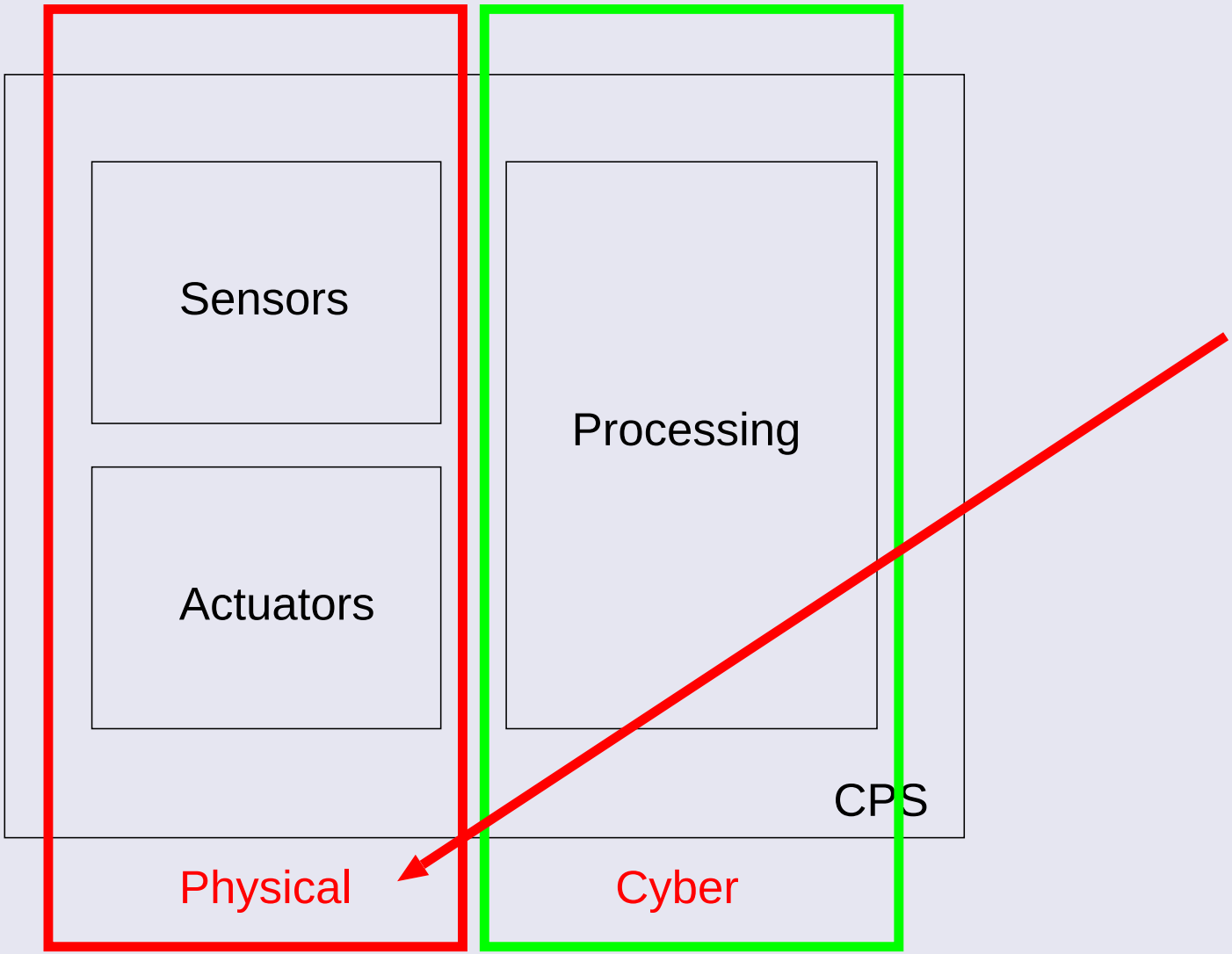
Tampering with the Keys of the car

Same Challenges?

Car Opening

Tampering directly with the Sensors of the Car

Cyber-Physical Systems



Challenge Three Summary

We need also to “ Secure the Physics”

Conclusions

- Interaction Between Robustness and Countermeasures
- Support Automation for Security
- Secure the Physics

Questions?

- “Would you tell me, please, which way I ought to go from here?”
- “That depends a good deal on where you want to get to,” said the Cat
- “I don’t much care where” said Alice.
- “Then it doesn’t matter which way you go,” said the Cat.
- “so long as I get SOMEWHERE,” Alice added as an explanation.
- “Oh, you’re sure to do that,” said the Cat, “if you only walk long enough.”

Alice In the Wonderland

Thank you for your attention!

mail: regazzoni@alari.ch