Physically Unclonable Functions - PUFs

Paolo PRINETTO
Director
CINI Cybersecurity
National Laboratory

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Acknowledgments

- The presentation includes material from
  - Giorgio DI NATALE
  - Elena Joana VATAJELU

  whose valuable contribution is here acknowledged and highly appreciated

Goal

- Introduce *PUF - Physically Unclonable Functions*
- Present a taxonomy
- Present some practical implementations

Outline

- Introduction
- Silicon PUFs
- Definitions and properties
- PUF Applications

Outline

- Introduction
- Silicon PUFs
- Definitions and properties
- PUF Applications

The ultimate goal

- Preventing counterfeiting to a maximum extent

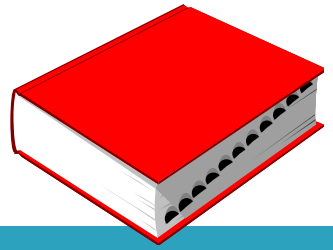
Ideal solution

- Associating each device an ID that is:
  - Unique for each device
  - Unclonable
  - Marginally impacting on circuit performances and global costs
  - Time invariant
  - Robust
  - ...

Practical solution

- Implement & exploit *PUFs* (*Physically Unclonable Functions*)

PUFs



- A PUF (Physically Unclonable Function) is a built-in mechanism that generates one stable different ID for each identically-manufactured device, without the need of
  - programming the ID value
  - storing the value

PUFs -- Advantages

- No reverse engineering can be applied
- Even if you discover an ID in one circuit, you cannot "clone" the device, since *each* manufactured device has its own unclonable ID

PUFs – Basic concept

Basic concept

- PUFs are based on the comparison of *nominally-identical* physical characteristics

Examples

- Delay of some networks
  - Ring Oscillator PUF
  - Arbiter PUF
- Content of SRAMs at power-up
- Resistance of STT-MTJs elements
- Capacitance of TSVs

Outline

- Introduction
- Silicon PUFs
- Definitions and properties
- PUF Applications

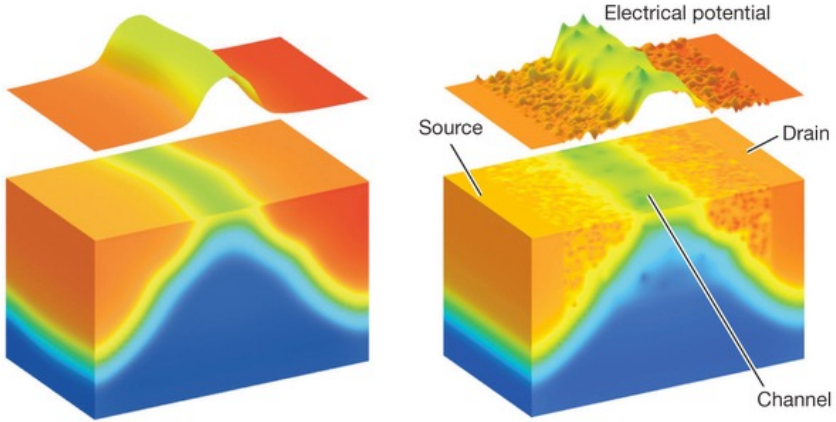
Silicon PUFs

- Silicon PUFs exploit inherent physical variations (process variations) that exist in modern integrated circuits

Process variations

- Process variations are uncontrollable and unpredictable, making PUFs suitable for IC identification and authentication

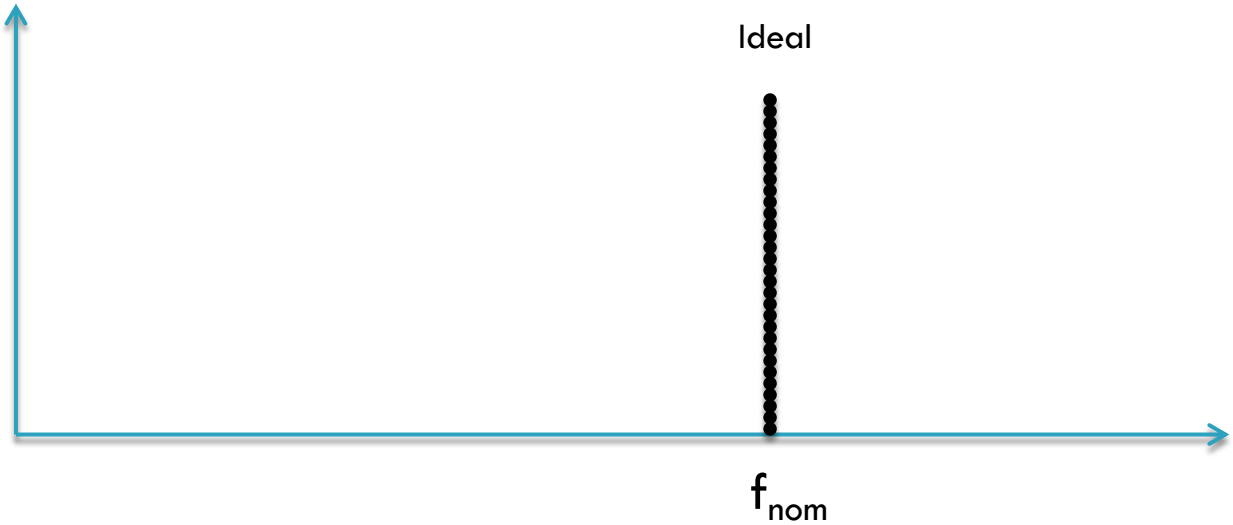
Random Dopant Fluctuation



[Igor L. Markov,
Limits on fundamental limits to computation,
Nature 2014]

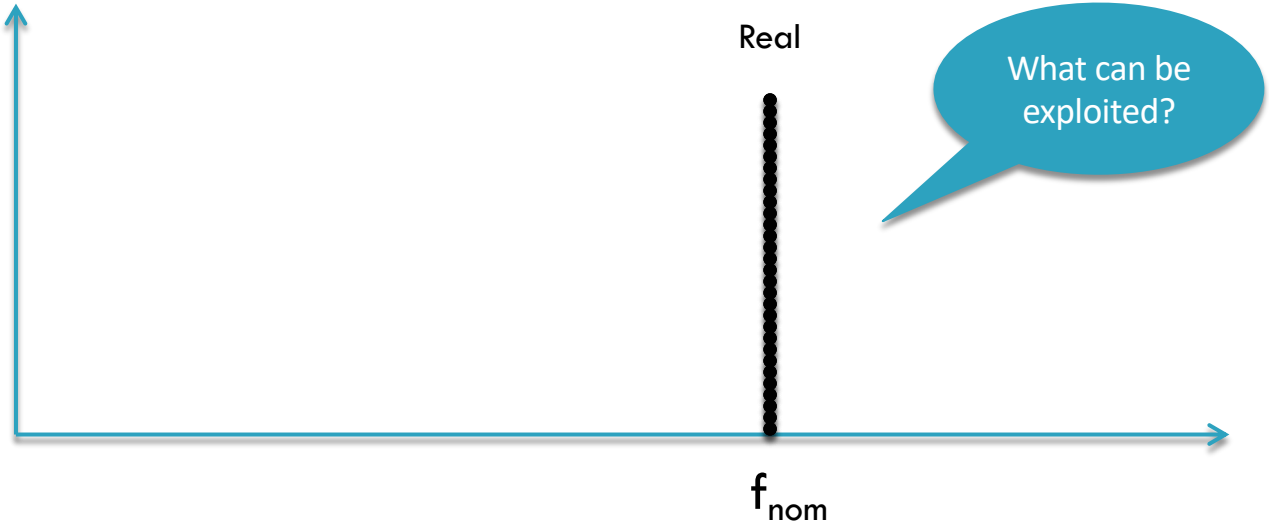
Process Variability

- Example: frequency of **identical** ring oscillators



Process Variability

- Example: frequency of **identical** ring oscillators

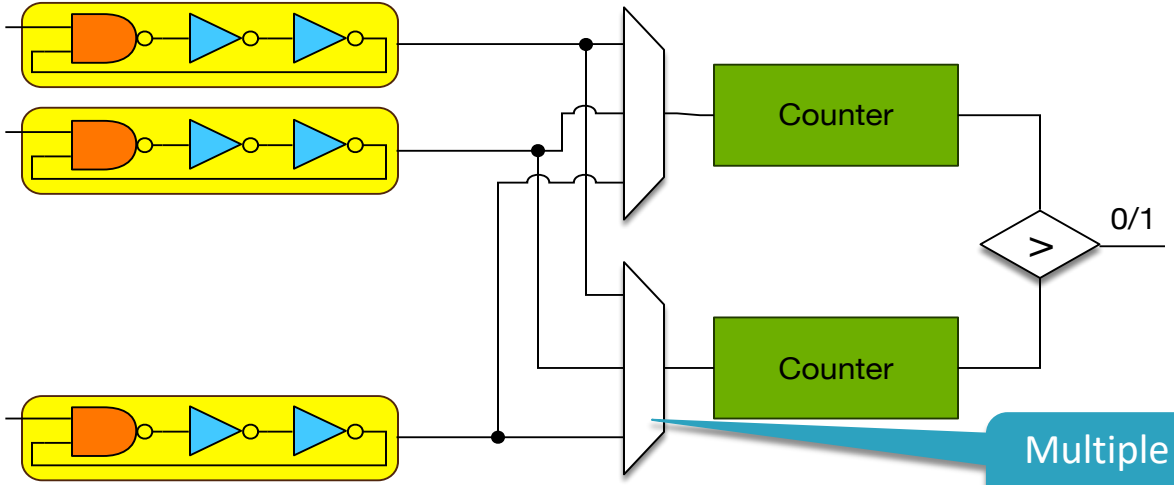


Silicon PUFs -- Implementations

- The most widely used Silicon PUFs implementations include:
  - *Ring Oscillator PUFs*
  - *Arbiter PUFs*
  - *SRAM-based PUFs*

Ring Oscillator PUF

- Ideal: Frequencies of all Ring Oscillators identical
- Reality: because of process variations, all different!



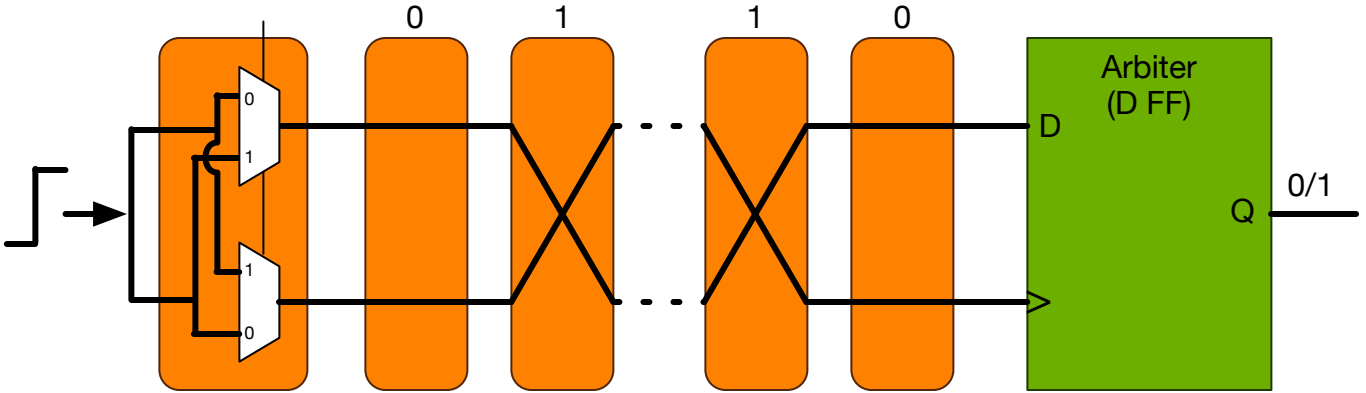
Ring Oscillator PUF

- Due to manufacturing variation, each ring oscillator oscillates with a slightly different frequency
- The counter increase its value every “loop”
- N oscillators give N(N-1) / 2 distinct pairs. E.g.,

N = 16 => 16*(16 – 1)/ 2 = 120 bits

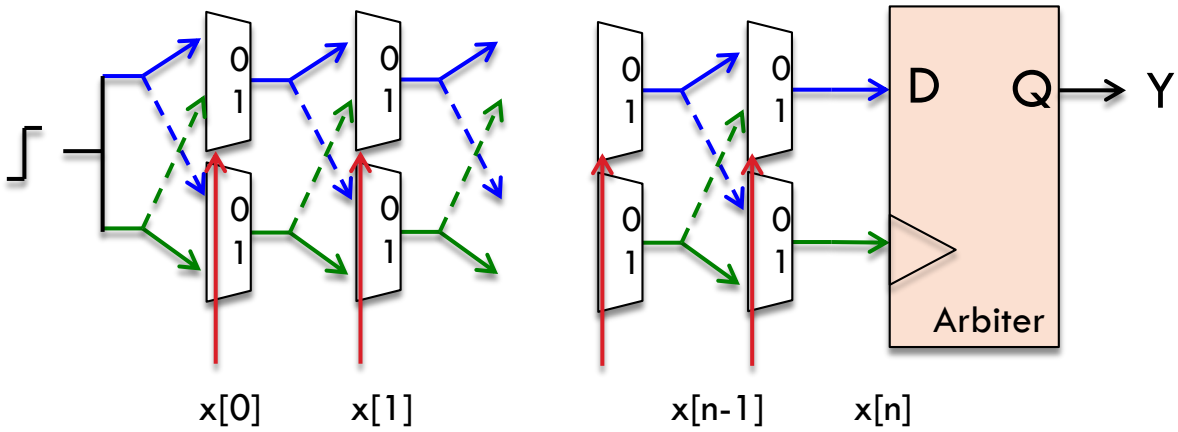
Arbiter PUF

- Ideal: Delays of all the paths from input to output identical
- Reality: because of process variations, all different!



Arbiter PUF

- The input challenge X determine the PATH by controlling the muxes



Arbiter PUF

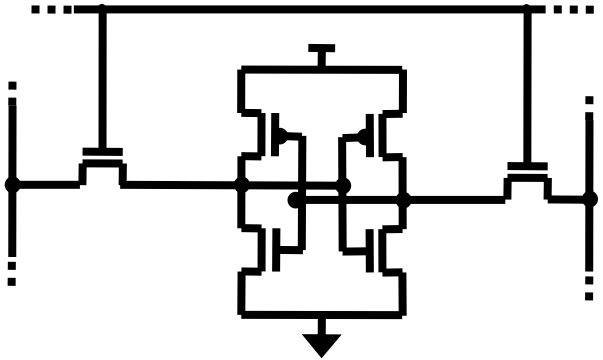
- A rising signal is given to both paths at the same time, the signals race through the two **delay paths**, and the arbiter (latch) at the end decides which signal is faster
- The output is
  - **one** if the signal to the latch data input (D) is faster
  - **zero** otherwise

SRAM-based PUFs

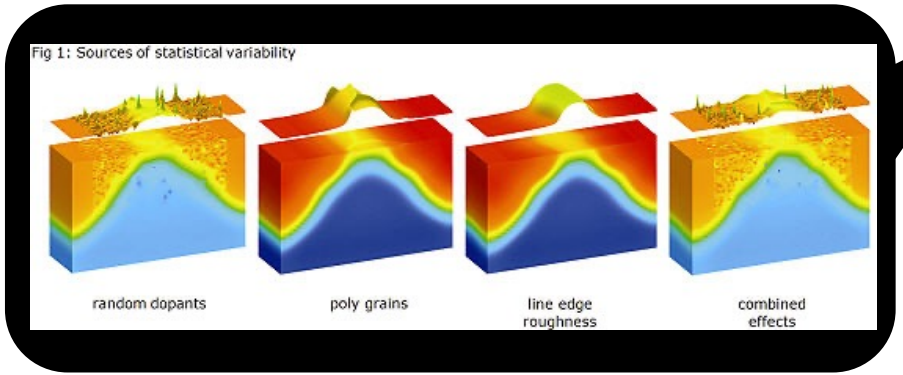
- “Strength” of all inverters: identical
- Reality: because of process variations, all different!

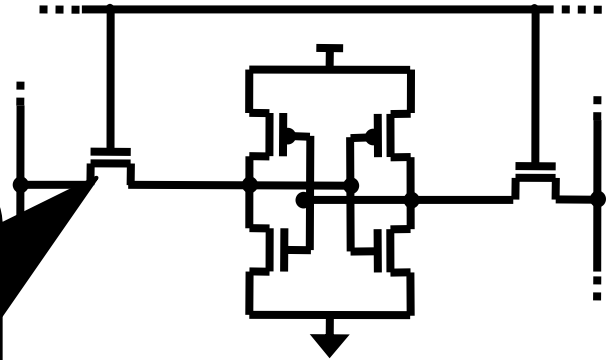
Power Up

SRAM-based PUFs

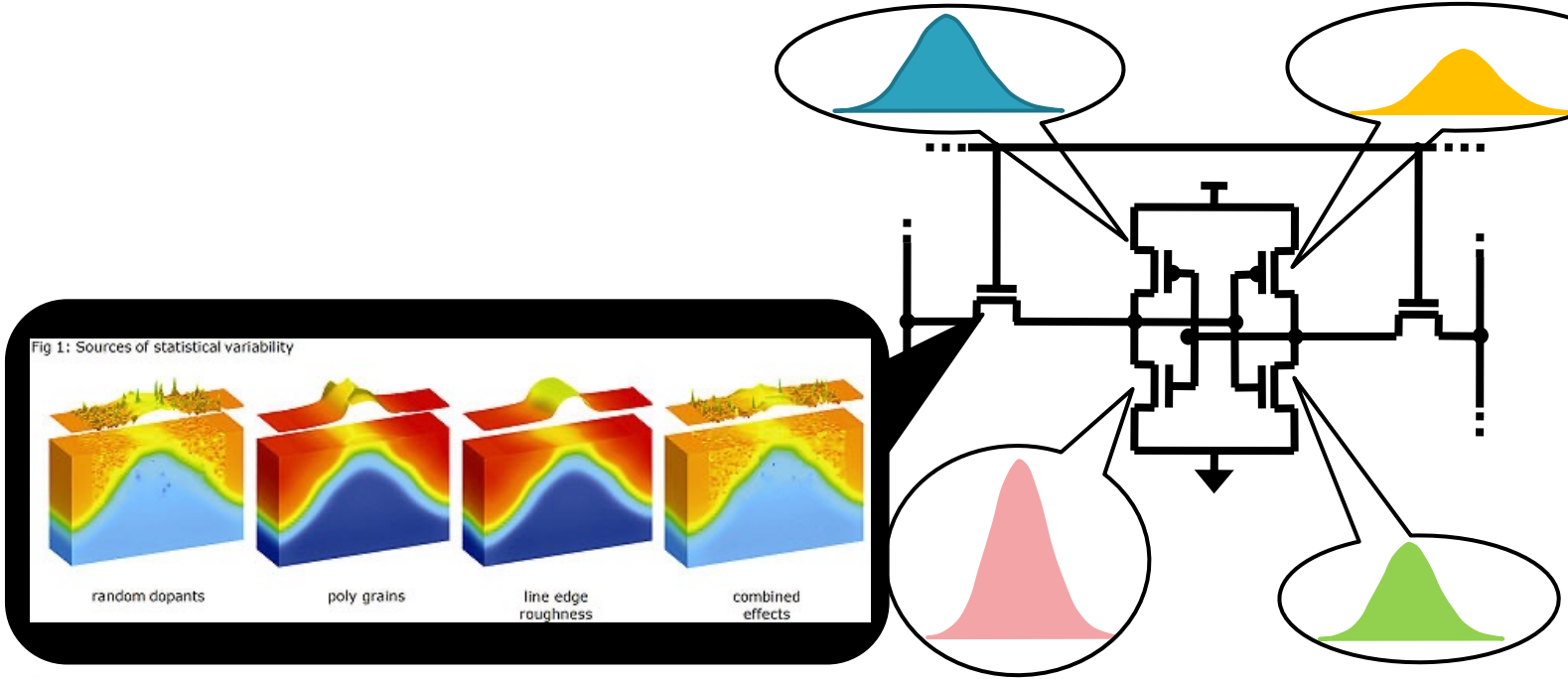


SRAM-based PUFs

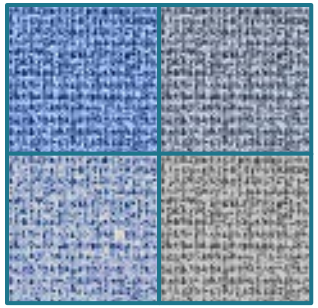


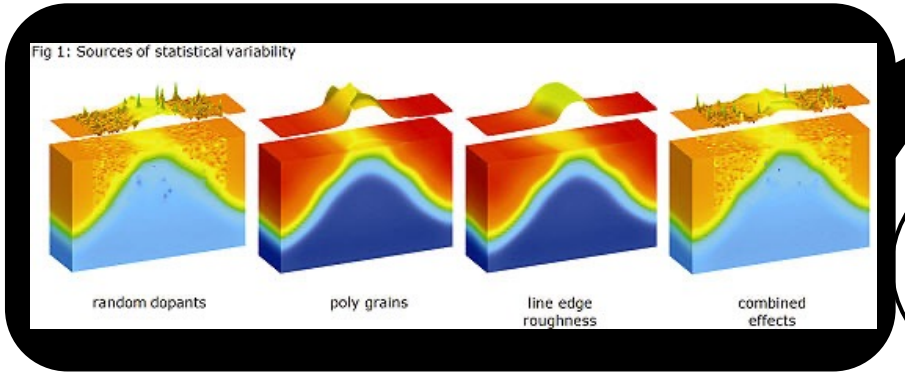


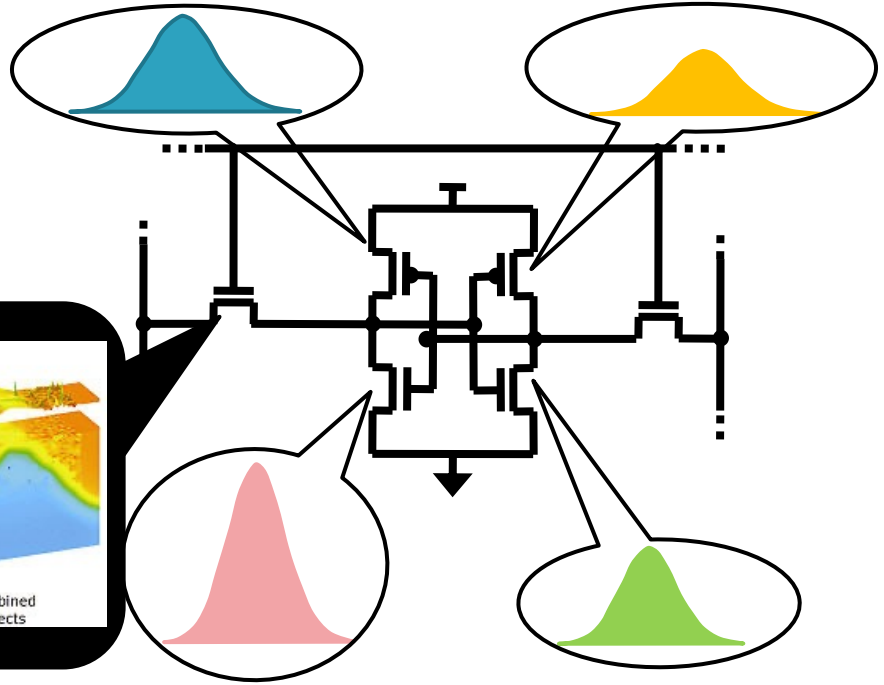
SRAM-based PUFs



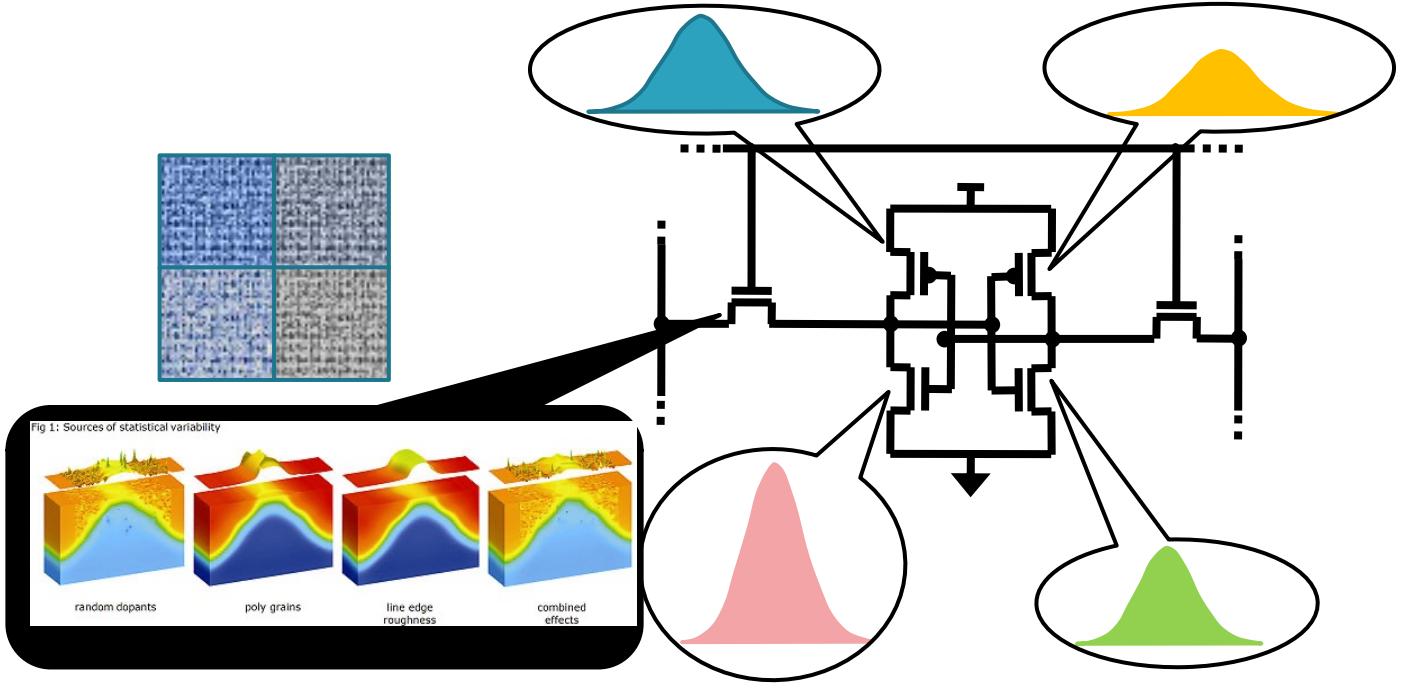
SRAM-based PUFs



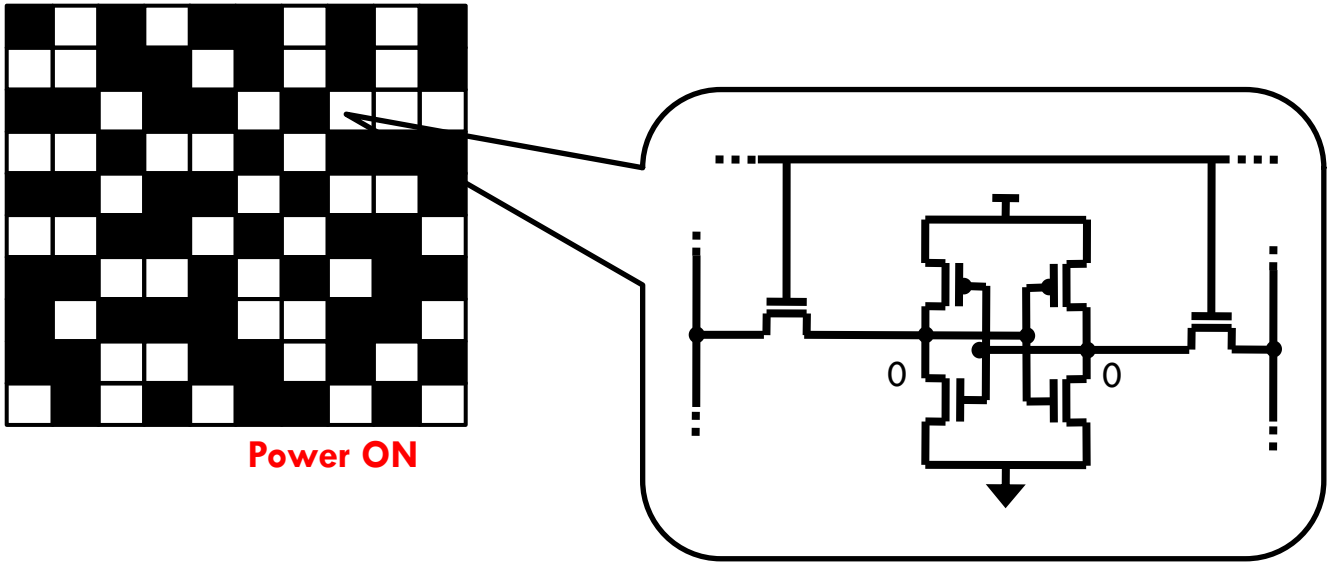




SRAM-based PUFs



SRAM-based PUFs

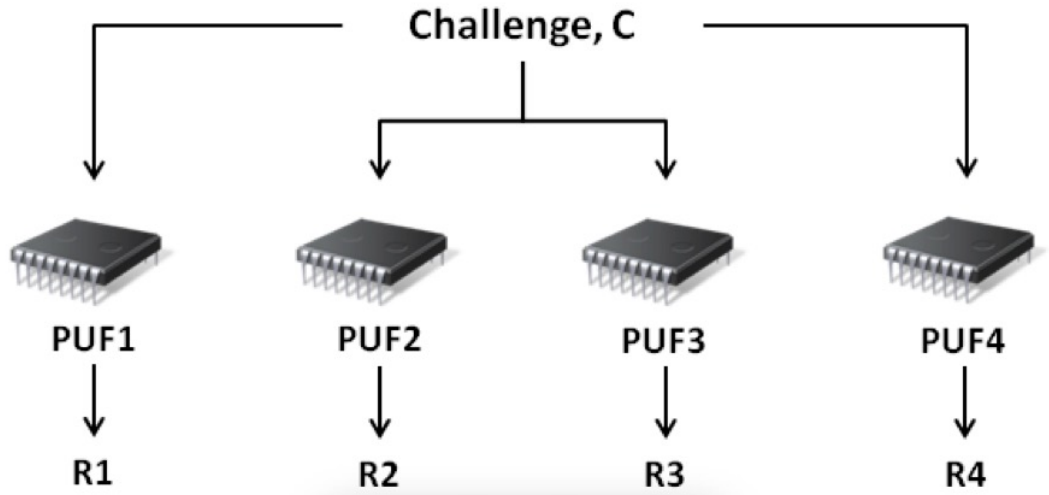


Outline

- Introduction
- Silicon PUFs
- Definitions and properties
- PUF Applications

Challenge-Response Pair (CRP)

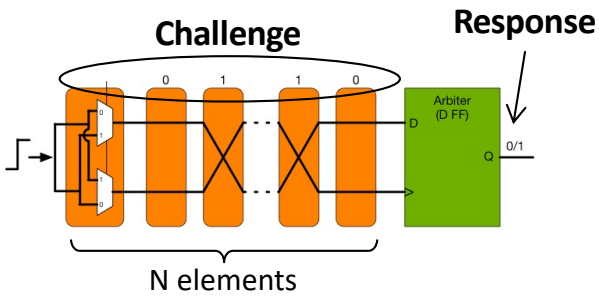
- Is the pair *input-combination/generated-output*



PUFs Classification

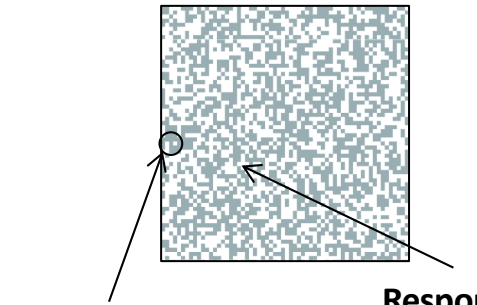
- PUFs can be classified in:
  - *Strong PUF*: the single device can generate a huge number of CRPs ($>2^{80}$)
  - *Weak PUF*: the single device can generate few CRPs

Strong PUF



2^N CRPs

Weak PUF



Challenge
(the address of
the selected bit)

Response
(the content of
the SRAM cell
after power-up)

From few CRPs up to
16 Mbit ➔ 2^{24} CRPs

PUFs Properties

- *Unpredictability*: It is not possible to predict a CRP even knowing other CRPs
- *Unclonability*: It is not possible to manufacture 2 (or more) devices with the same CRPs set
- *Uniformity*: The number of 0 and 1 are equally distributed in the whole set of CRPs
- *Repeatability*: when the challenge is applied, the same device has always the same response

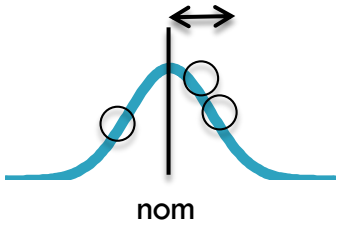
PUF Properties – Uniformity

- The number of 0 and 1 are equally distributed in a response.
- The ideal condition is RU = 50%

$$RU(c) = \frac{1}{N} \sum_{t=1}^{N} bit(\pi(c), t) * 100\%$$

PUF Properties – Repeatability

- The sets of CRPs should not change in time
  - Also called the “reliability” of the PUF
- Nevertheless, PUFs are affected by:
  - Aging
  - Environmental conditions
    (Temperature , Voltage, Noise, …)



- There is no way to avoid aging and variations of environmental conditions ☹

PUF Properties – Uniqueness

- It is the strength evaluation between two devices.
- Uniqueness U among K different PUF devices is:

$$U = \left(\frac{2}{K(K-1)}\right) * \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD\left(\pi_i(c), \pi_j(c)\right)}{N} * 100\%$$

  - Where HD() is the Hemming Distance between 2 responses.
  - Ideally HD between 2 different responses is 50% (U = 100%)

Outline

- Introduction
- Silicon PUFs
- Definitions and properties
- PUF Applications

PUF Applications

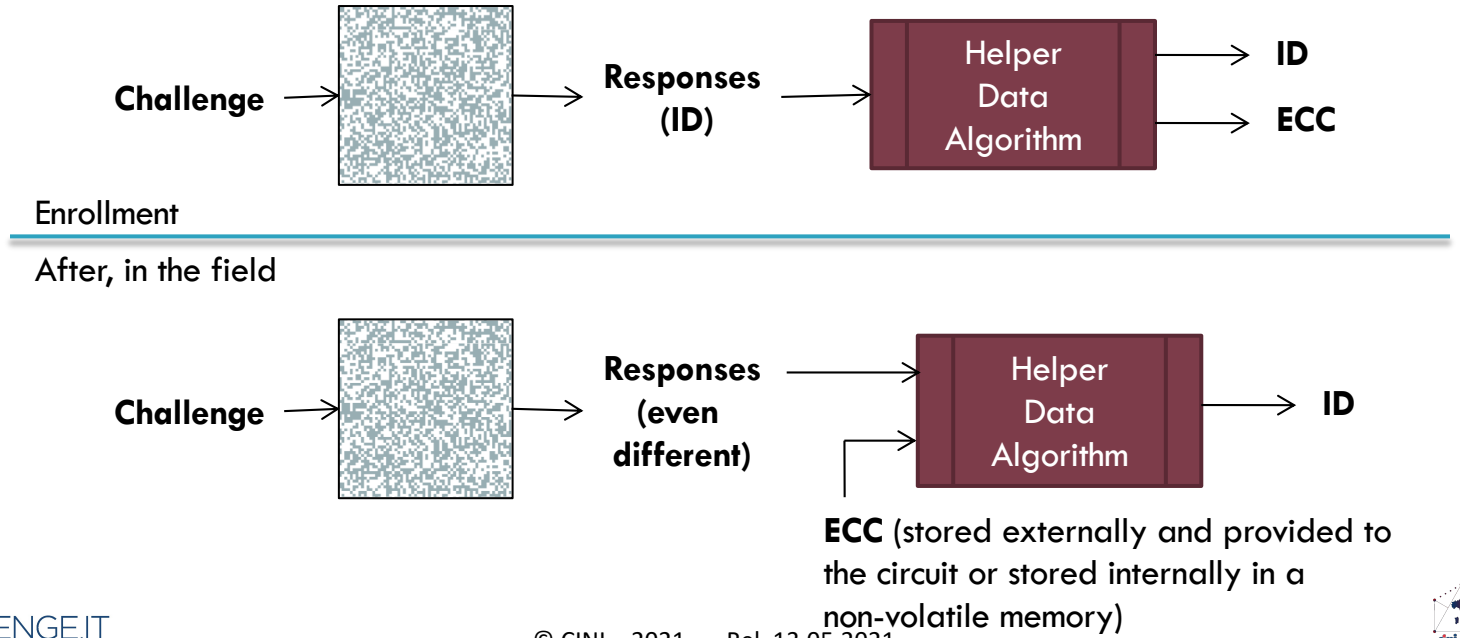
- *Weak PUF*
  - Used for device identification or to generate secret keys
  - SRAM-based
- *Strong PUF*
  - Used to perform device authentication
  - Arbiter, Ring-Oscillator

Weak PUFs: Device Identification / keys

- The content of the SRAM (after power-up) can be used as an ID or an internal secret key
- Major issue: the reliability of the PUF
  - If the value provided by the PUF differs in time, the device cannot be identified

Weak PUFs: helper data

- Solution: Helper Data (=Error Correction Codes, ECC)



Weak PUFs – (Random) Secret keys

- Responses can be used to generate internal secret keys
- To be used for:
  - Encryption of data stored in the circuit
  - Encryption of secret keys stored in non-volatile memories

Strong PUFs - IC Authentication

- After manufacturing, each device is challenged by *several random* challenges
- Responses are stored in an external secure database
- To authenticate the device, some of the challenges are used during mission mode
- Even if some CRPs do not correspond, the authentication can be performed

Conclusions

- PUFs are an excellent solution for increasing security and trust
- Many works in literature, to cover:
  - Quality and Reliability
  - Emerging technologies
  - Attacks

Paolo PRINETTO

Director
CINI Cybersecurity
National Laboratory

Paolo.Prinetto@polito.it

Mob. +39 335 227529

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it