TELECOM Paris

IP PARIS

PUF
Physically Unclonable Function
A Device Fingerprint to Increase Security in Digital Systems

Jean-Luc DANGER

Séminaire ICE

8 février 2023

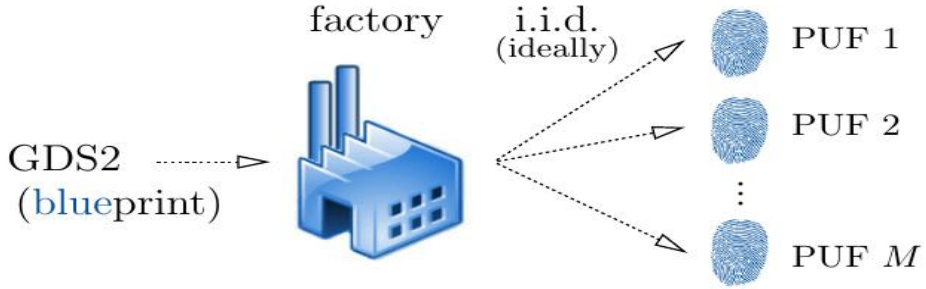
Institut Mines-Télécom

Outline

- What and Why a PUF ?
- PUF types in CMOS
- Is PUF a panacea for security ?
- How to make the PUF more reliable?
- How to make the PUF more secure ?
- Conclusions

Physically Unclonable Function: PUF

- **Function returning the fingerprint of a device**
  - Physical function,
  - which exploits material randomness, during fabrication (mismatch)
  - and is unclonable: same structure for each device

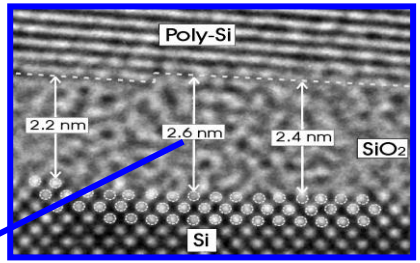


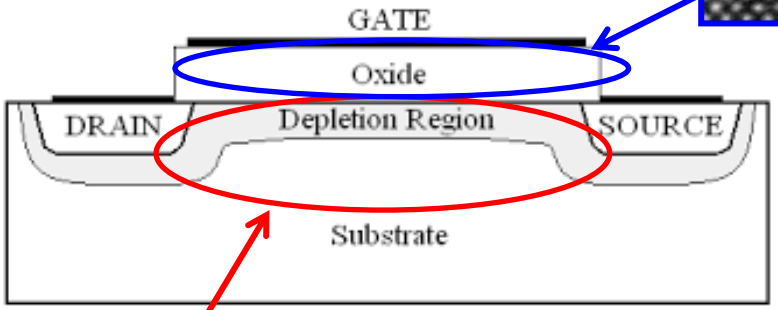
PUFs are instanciations of blueprints by a fab plant

a PUF ID is unique to each device

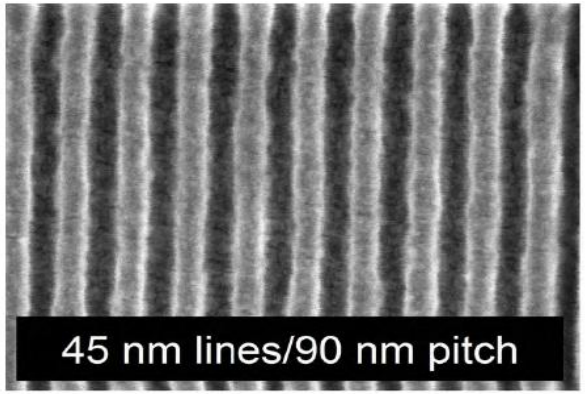
Process mismatch in CMOS technology

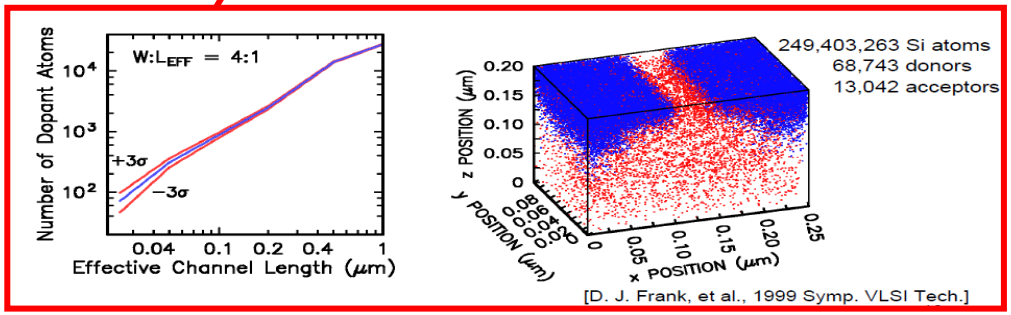
- **Examples**
  - Oxide thickness
  - Random dopant fluctuation
  - Metal line edge roughness





MOSFET transistor



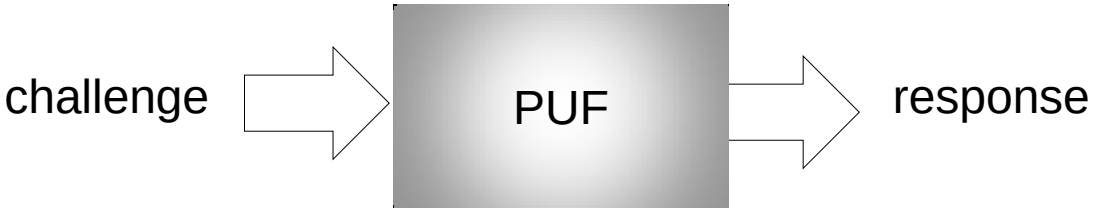


Advantages of PUF vs Non Volatile Memory "NVM"

- **PUF is self contained**
  - NVM has to be programmed with an ID, and can be tampered
- **Not clonable**
  - PUF has the same structure, NVM can be reverse engineered
- **Feasible in standard CMOS process**
  - NVM requires a specific process

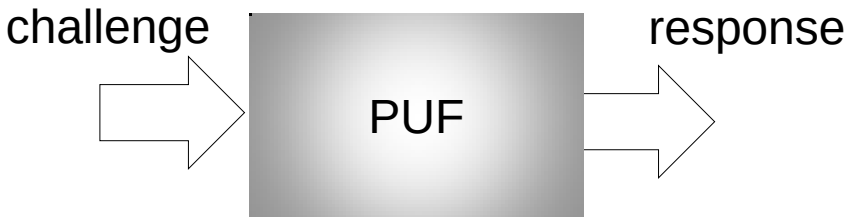
PUF interface

- **challenge ⇨ response**



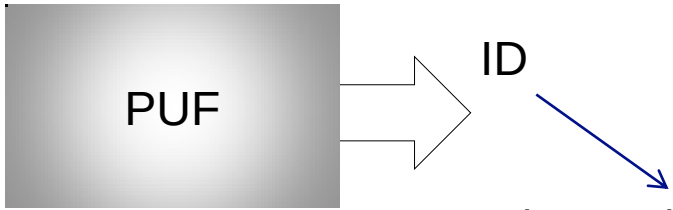
2 types of PUF identifiers

- **List of public challenges/responses pairs CRP**



"strong" PUF

- **No challenges, private identifier**

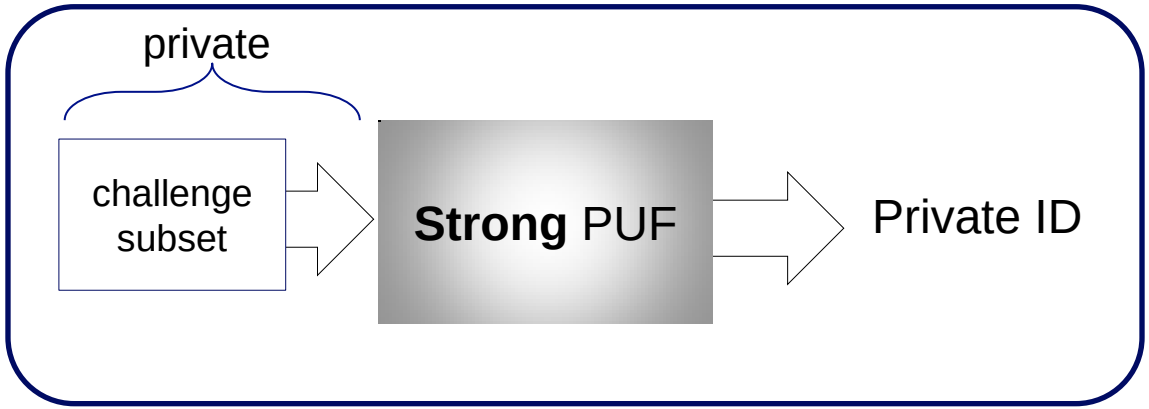


"weak" PUF

can be used as a cryptographic key

Strong vs Weak PUF

- **The terms strong and weak are misleading**
  - They just indicate the number of challenges.
- **A strong PUF can be used as weak PUF:**



weak PUF

Necessity to enroll the ID

- **Enrollment phase**
  - Carried out just after fabrication
  - The ID (cryptographic key or list of CRPs) is stored in a trusted server
  - The direct access to the PUF (necessary for the weak PUF) is locked at the end of enrollment
- **Reconstruction phase**
  - Corresponds to the PUF usage
  - The ID is self reconstructed by accessing the PUF (hence no storage)
  - A security protocol with the trusted server can take place

PUF main use-cases

- **Authentication**
  - strong PUF:
    - the authentication is performed by a challenge-response pair (CRP) protocol
    - Well suited for low-cost devices as there is no need of cryptography
  - weak PUF:
    - a cryptographic protocol is used with the cryptographic key output from the PUF
- **Confidentiality**
  - weak PUF only:
    - Encryption/decryption is used with the cryptographic key output from the PUF

Outline

- What and Why a PUF ?
- PUF types in CMOS
- Is PUF a panacea for security ?
- How to make the PUF more reliable?
- How to make the PUF more secure ?
- Conclusions

PUF types in CMOS

PUF type	physical source	design type	members
Delay-PUF	difference of delays in delay chains	standard	Arbiter-PUF, XOR-arbiter PUF, interpose-PUF, RO-PUF, RO-sum PUF, Loop-PUF
Memory-PUF	difference of threshold voltage of two looped inverting gates	standard or no design	SRAM-PUF, DFF-PUF, latch-PUF, buskeeper-PUF, MECCA-PUF, TERO-PUF
Metal-PUF	conductivity of wires and vias	custom	Contact-PUF, Litho-PUF
Oxide-breakdown-PUF	gate oxide rupture when stressed	custom	SOFT-BD-PUF
Emerging NVM -PUF	cell resistivity after initialisation	custom, hybrid technology	RRAM-PUF, MRAM-PUF

strong PUF

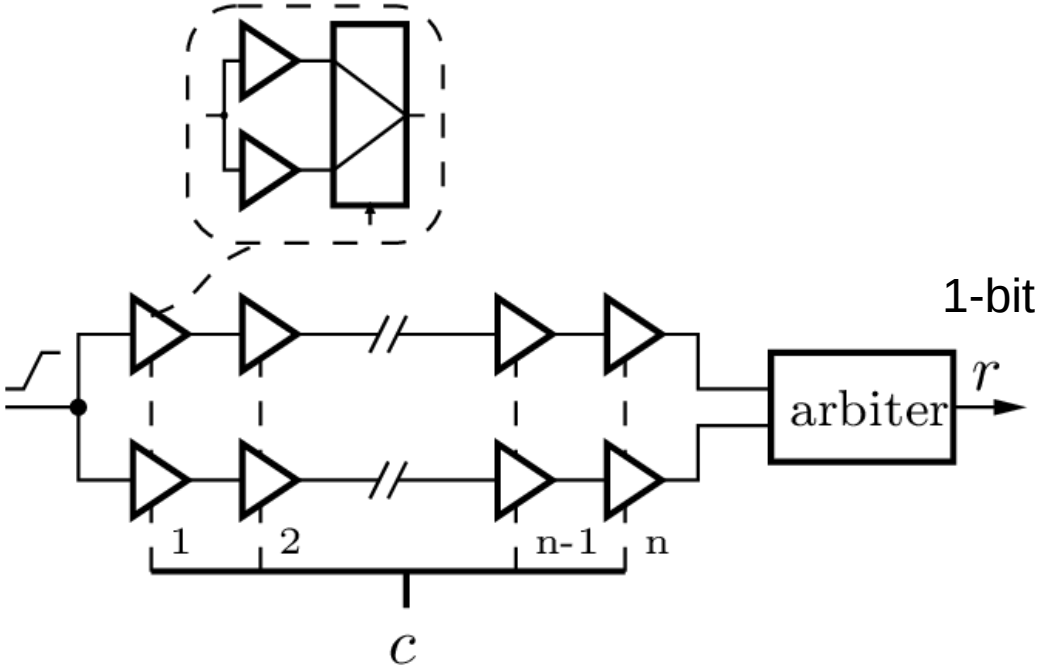
weak PUF

PUF types in CMOS

PUF type	physical source	design type	members
Delay-PUF	difference of delays in delay chains	standard	Arbiter-PUF, XOR-arbiter PUF, interpose-PUF, RO-PUF, RO-sum PUF, Loop-PUF
Memory-PUF	difference of threshold voltage of two looped inverting gates	standard or no design	SRAM-PUF, DFF-PUF, latch-PUF, buskeeper-PUF, MECCA-PUF, TERO-PUF
Metal-PUF	conductivity of wires and vias	custom	Contact-PUF, Litho-PUF
Oxide-breakdown-PUF	gate oxide rupture when stressed	custom	SOFT-BD-PUF
Emerging NVM -PUF	cell resistivity after initialisation	custom, hybrid technology	RRAM-PUF, MRAM-PUF

strong PUF

Delay PUF: arbiter-PUF



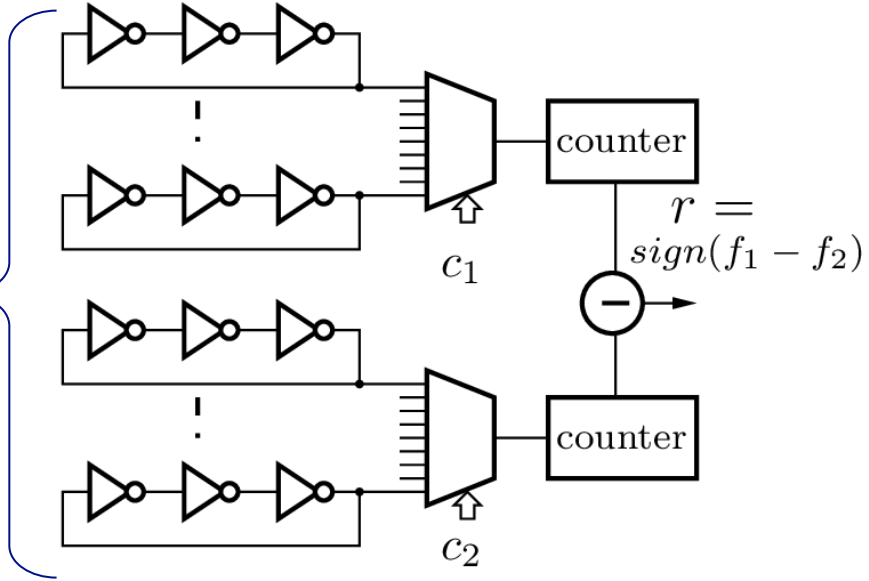
1-bit response

N-bit challenge

GASSEND B., CLARKE D. E., VAN DIJK M., DEVADAS S., “Silicon physical random functions”, ATLURI V., ed., Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002, ACM, p. 148–160, 2002.

Delay PUF: RO-PUF

Identical
Ring Oscillators

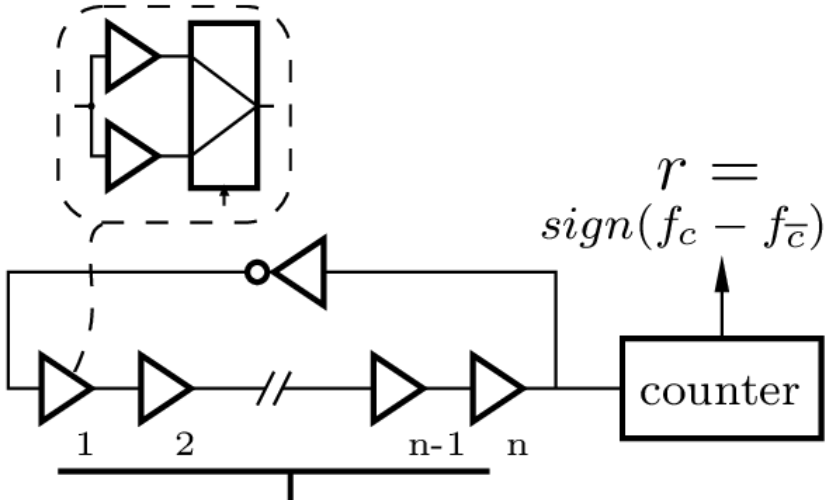


1-bit response
M-bit magnitude

challenge c1||c2

SUH G. E., DEVADAS S., “Physical Unclonable Functions for Device Authentication and Secret Key Generation”, Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, June 4-8, 2007, IEEE, p. 9–14, 2007.

Delay PUF: Loop-PUF



1-bit response
M-bit magnitude

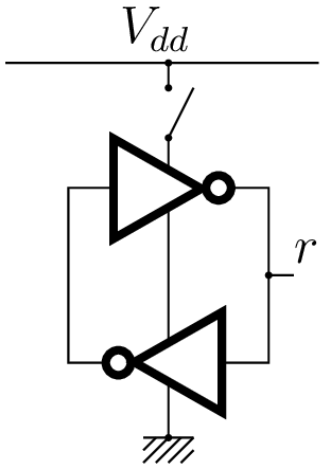
CHERIF Z., DANGER J.-L., GUILLEY S., BOSSUET L., “An Easy-to-Design PUF based on a single oscillator: the Loop PUF”, DSD, September 5-8 2012.

PUF types in CMOS

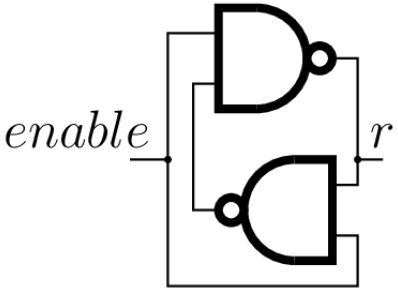
PUF type	physical source	design type	members
Delay-PUF	difference of delays in delay chains	standard	Arbiter-PUF, XOR-arbiter PUF, interpose-PUF, RO-PUF, RO-sum PUF, Loop-PUF
Memory-PUF	difference of threshold voltage of two looped inverting gates	standard or no design	SRAM-PUF, DFF-PUF, latch-PUF, buskeeper-PUF, MECCA-PUF, TERO-PUF
Metal-PUF	conductivity of wires and vias	custom	Contact-PUF, Litho-PUF
Oxide-breakdown-PUF	gate oxide rupture when stressed	custom	SOFT-BD-PUF
Emerging NVM -PUF	cell resistivity after initialisation	custom, hybrid technology	RRAM-PUF, MRAM-PUF

weak PUF

Weak PUF: Memory-PUF



SRAM PUF



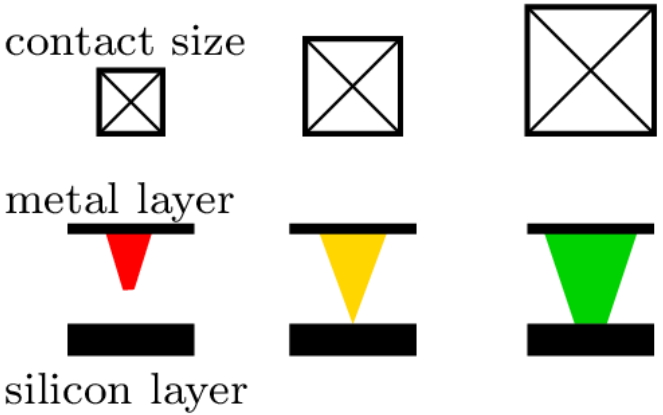
LATCH PUF

Any non-initialized SRAM is potentially a PUF !

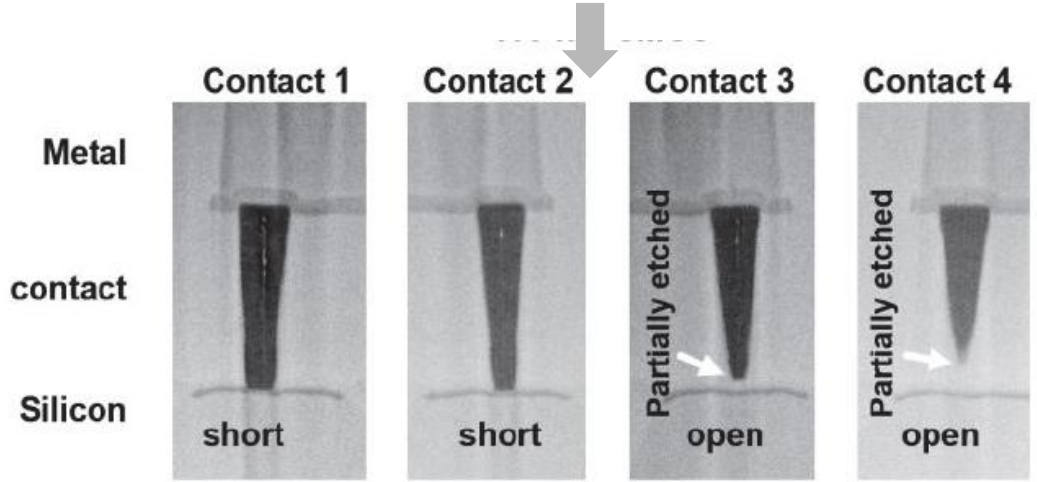
Guajardo, J., Kumar, S. S., Schrijen, G. J., & Tuyls, P. (2007). FPGA intrinsic PUFs and their use for IP protection. In *Cryptographic Hardware and Embedded Systems-CHES 2007: 9th International Workshop, Vienna, Austria, September 10-13, 2007. Proceedings 9* (pp. 63-80). Springer Berlin Heidelberg.

Kumar, S. S., Guajardo, J., Maes, R., Schrijen, G. J., & Tuyls, P. (2008, June). The butterfly PUF protecting IP on every FPGA. In *2008 IEEE International Workshop on Hardware-Oriented Security and Trust* (pp. 67-70). IEEE.

Weak PUF: Contact-PUF



Scanning electron mircroscopy in 180nmCMOS

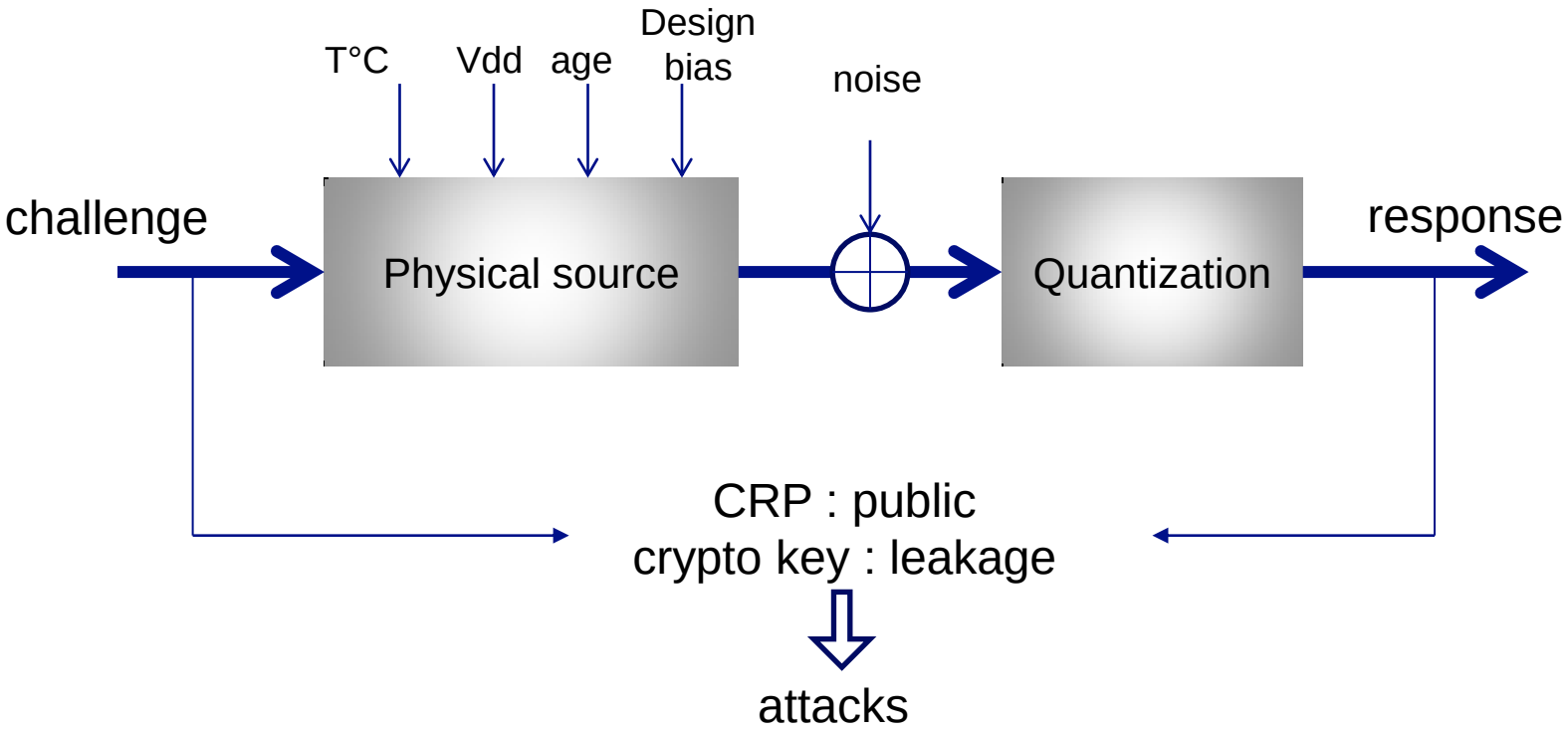


Jeon, D., Lee, D., Kim, D. K., & Choi, B. D. (2022, June). Contact PUF: Highly Stable Physical Unclonable Functions Based on Contact Failure Probability in 180 nm, 130 nm, and 28 nm CMOS Processes. In *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (pp. 85-88). IEEE.

Outline

- What and Why a PUF ?
- PUF types in CMOS
- Is PUF a panacea for security ?
- How to make the PUF more reliable?
- How to make the PUF more secure ?
- Conclusions

PUF in its real environment



Is PUF a panacea for security? 1/2

- **Definitely NO concerning the raw PUF**

1. The raw PUF is **unreliable** : 2 to 15% of Bit Error Rate
   - much **noise** in silicon: thermal, flicker, coupling
   - impact of T°C, Vdd, aging

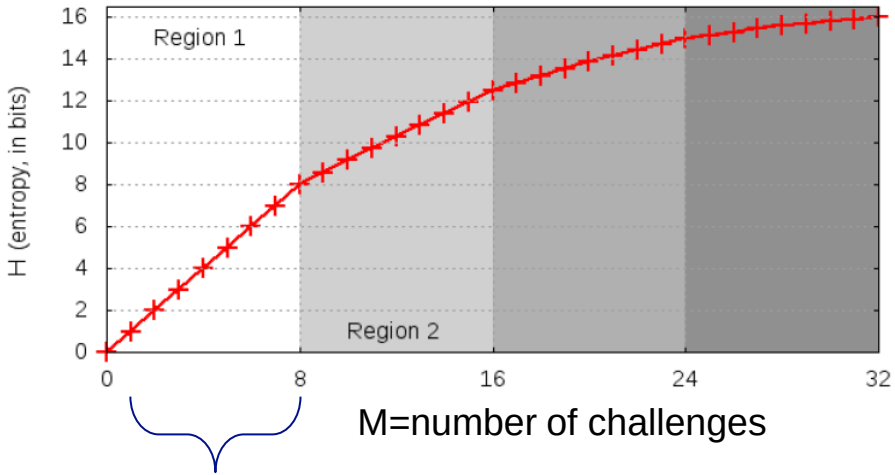
2. Its **entropy** may be insufficient (presence of bias)
   - Intra-entropy (or **randomness**):
     - entropy of the responses
   - Inter-entropy (or **uniqueness**):
     - PUFs never share the same ID.

Intra Entropy

n=number of elements

Weak PUF : H(n) < n theoretically n but design bias

Delay PUF : H(n) = n with Hadamard codes then no linear growth



8 hadamard codes n=8

Is PUF a panacea for security? 2/2

- **Definitely NO concerning the raw PUF**
  3. It is sensitive to powerful attacks
     - **Modeling** attacks
       - The PUF behavior of the strong PUF is modeled by using many CRPs
     - **Physical** attacks
       - The PUF behaviour is observed or modified

⇨ **An efficient postprocessing is required to get it reliable, robust and entropic**

Outline

- What and Why a PUF ?
- PUF types in CMOS
- Is PUF a panacea for security ?
- How to make the PUF more reliable?
- How to make the PUF more secure ?
- Conclusions

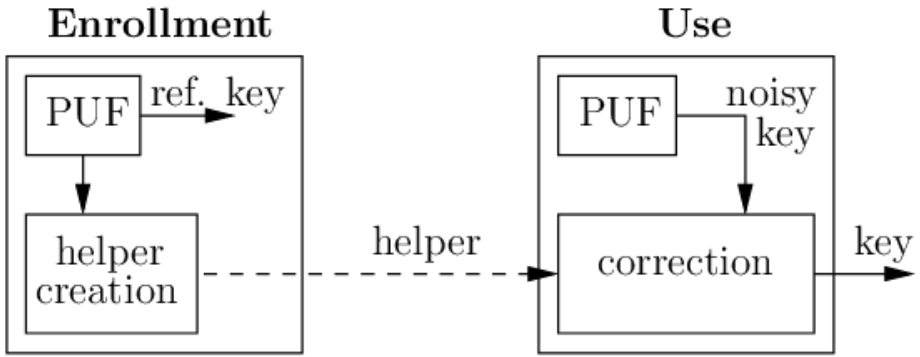
How to get a PUF more reliable

- **Use of Error Correction Codes (ECC)**
  - Needs a public word: the Helper Data
- **Filter out the unreliable bits: the dark bits**
  - Needs a public word: the Helper Data
- **Use technology which provides native steadiness**
  - Bur requires custom design and new technologies as:
    - Contact PUF
    - Oxyde breakdown PUF
    - RRAM PUF

Note: This mainly applies for **weak PUFs** as the unreliability of strong PUFs can be managed at CRP protocol

Helper Data to enhance reliability

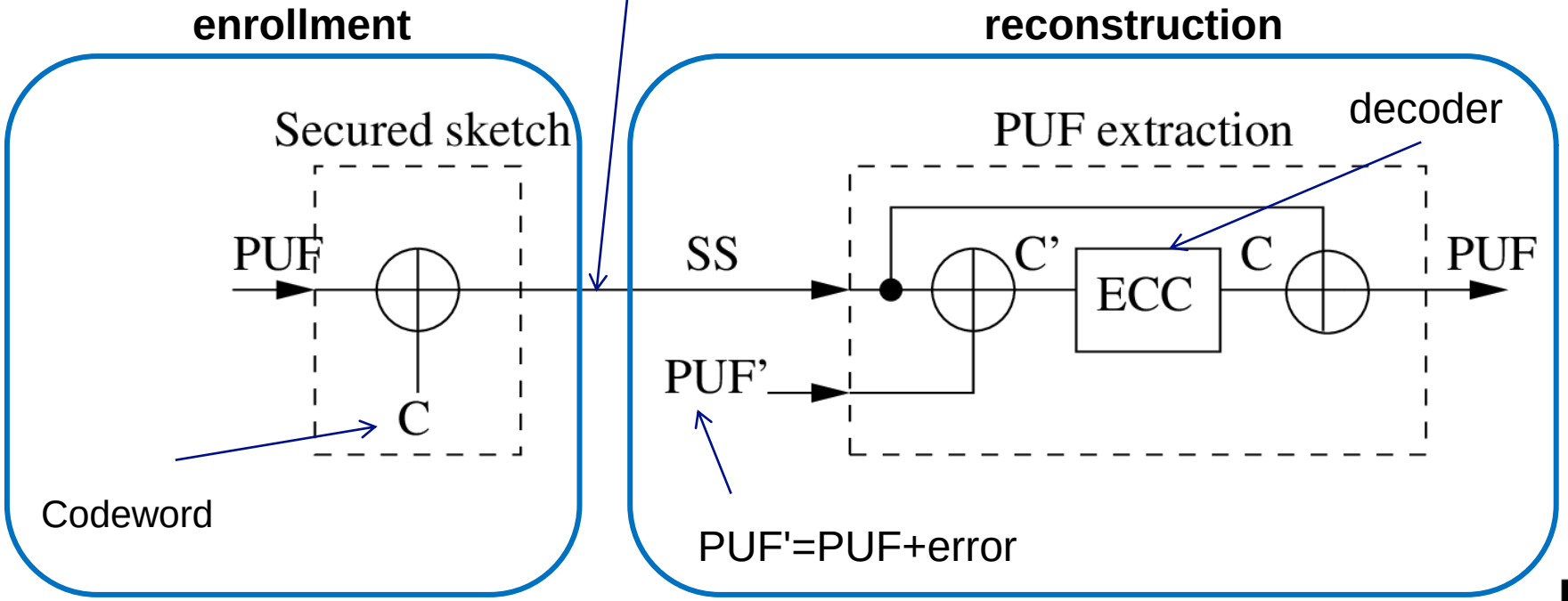
- **Helper Data Build during Enrollment**
  - Public word associated with the reference key of the PUF
- **Used During Reconstruction**
  - To help correcting errors



Helper Data with ECC to correct PUF

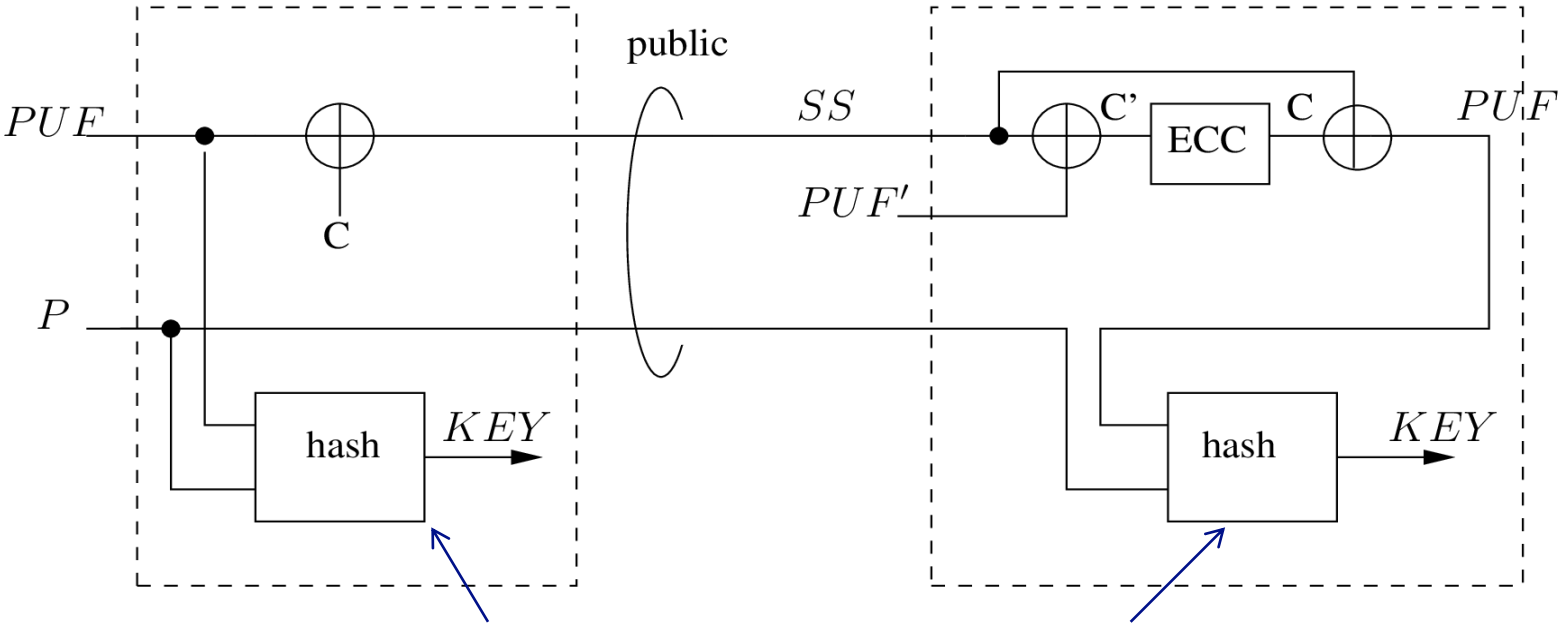
Code offset construction

Helper Data = "Secure Sketch"



DODIS Y., REYZIN L., SMITH A., “Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data”, EUROCRYPT, p. 523-540, 2004

Fuzzy extraction for Key generation



The Key changes with Hash(P||PUF)

DODIS Y., REYZIN L., SMITH A., “Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data”, EUROCRYPT, p. 523-540, 2004

Helper Data to indicate unreliable bits

- **The response bits are declared unreliable under a certain threshold of BER**
- **How to determine this threshold:**
  - Native weak PUF
    - carry out multiple tries as the response is 1-bit
  - for strong PUFs used as weak PUFs (RO-PUF, Loop-PUF)
    - Directly depends on the M-bit magnitude responses
    - A stochastic model can be derived according to SNR

Delvaux, J., Gu, D., Schellekens, D., & Verbauwhede, I. (2014). Helper data algorithms for PUF-based key generation: Overview and analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *34*(6), 889-902.

Helper Data to indicate unreliable bits for strong PUFs

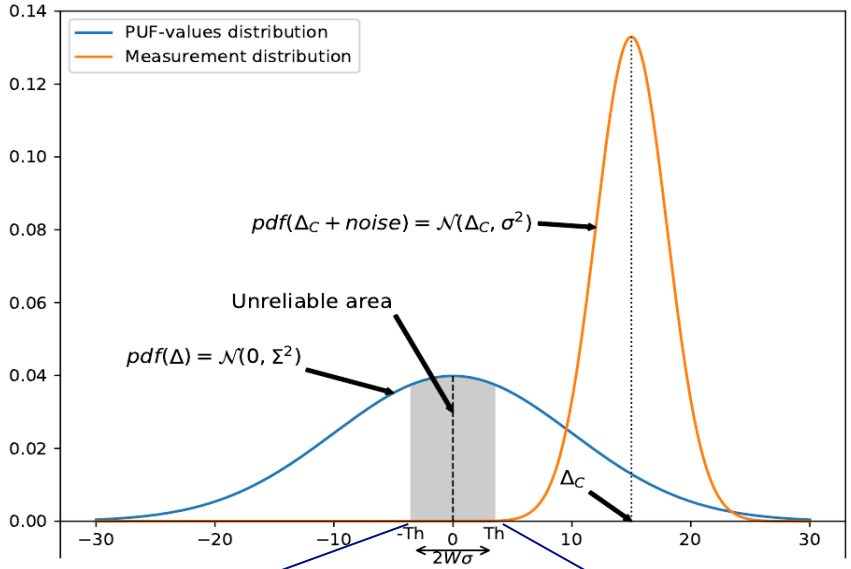
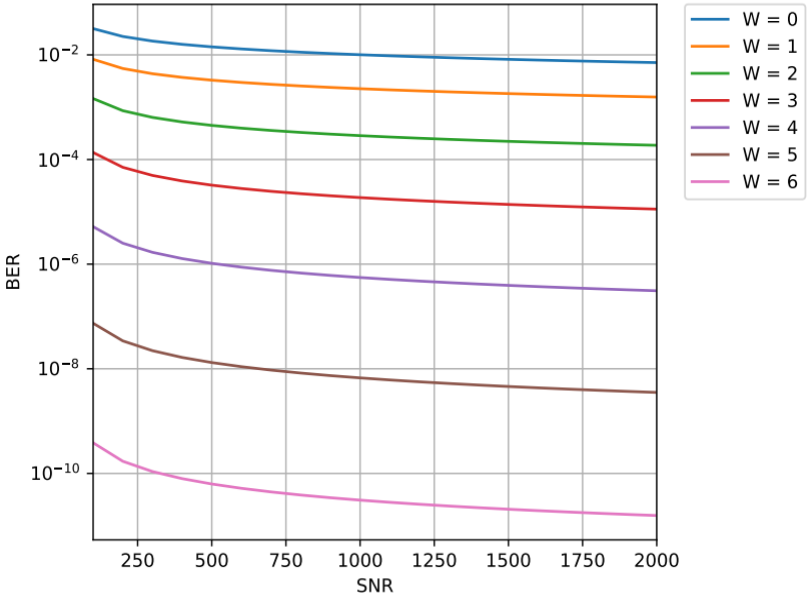


Figure 7: Unreliable area vs Distributions of Δ and noise

Bit **unreliable** ⇔ |response| < Th



$$\mathrm{SNR} = \frac{\mathbb{E}[\Delta_C^2]}{\mathbb{E}[Z^2]} = \frac{\Sigma^2}{\sigma^2}$$

SCHAUB A., DANGER J.-L., GUILLEY S., RIOUL O., “An Improved Analysis of Reliability and Entropy for Delay PUFs”, NOVOTNY M., KONOFAOS N., SKAVHAUG A., eds., 21st Euromicro Conference on Digital System Design, DSD 2018, Prague, Czech Republic, August 29-31, 2018, IEEE Computer Society, p. 553–560, 2018.

Outline

- What and Why a PUF ?
- PUF types in CMOS
- Is PUF a panacea ?
- How to make the PUF more reliable?
- How to make the PUF more secure ?
- Conclusions

Modeling Attacks

- **Based on Machine Learning algorithms**
  - To get the model of the Challenge-Response function
  - Applies only to strong PUFs with CRP protocol
- **Example : Arbiter-PUF**

$$B_i = \text{sign}(c_i \cdot X)$$

$$c_i \cdot X = \sum_{j=1}^{n} c_{i,j} X_j$$

Challenge i Delay difference Elementary delay difference

ML Method	No. of Stages	Prediction Rate	CRPs	Training Time
LR	64	95%	640	0.01 sec
		99%	2,555	0.13 sec
		99.9%	18,050	0.60 sec
LR	128	95%	1,350	0.06 sec
		99%	5,570	0.51 sec
		99.9%	39,200	2.10 sec

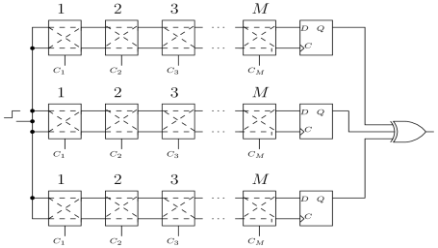
PUF

Very easy to attack by ML !

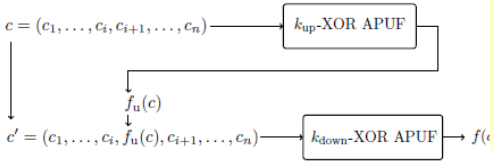
RUHRMAIR U., SEHNKE F., SOLTER J., DROR G., DEVADAS S., SCHMIDHUBER J., “Modeling attacks on physical unclonable functions”, Proceedings of the 17th ACM conference on Computer and communications security, p. 237–249, 2010.

Protections against ML attacks

- **PUF combination**
  - Examples:
    - XOR-arbiter PUF
    - Interpose PUF
  - Attack still works
    - with millions of CRPs



SAHOO D. P., SAHA S., MUKHOPADHYAY D., CHAKRABORTY R. S., KAPOOR H., “Composite PUF: A new design paradigm for Physically Unclonable Functions on FPGA”, 2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014, Arlington, VA, USA, May 6-7, 2014, IEEE Computer Society, p. 50–55, 2014



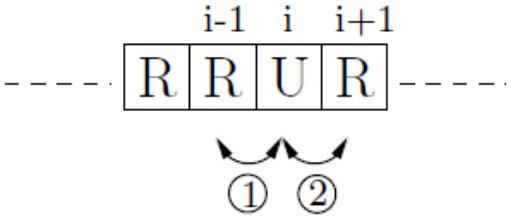
WISIOL N., MUHL C., PIRNAY N., NGUYEN P. H., MARGRAF M., SEIFERT J., VAN DIJK M., RUHRMAIR U., “Splitting the Interpose PUF: A Novel Modeling Attack Strategy”, IACR Trans. Cryptogr. Hardw. Embed. Syst., vol. 2020, no. 3, p. 97–120, 2020.

- **Challenge obfuscation**
  - By cryptographic block ?
    - ⇨ The use of CRP protocol is questionnable
- **Specific protocol**
  - As slender PUF, but proven insecure

Hence no ideal protection against ML !

Attack exploiting Helper Data

- **Public word => Helper Data manipulation to find dependency between HD and response, for instance:**
  - Related-key attacks by changing the HD ic code-offset
  - Swapping the reliability bits in bit selection HD



Abbreviations:
R: Reliable
U: Unreliable

- **Potential Protections**
  - Zero Leakage Helper Data
  - Lightweight Helper Data

STRIEDER E., FRISCH C., PEHL M., “Machine learning of physical unclonable functions using helper data: Revealing a pitfall in the fuzzy commitment scheme”, IACR Transactions on Cryptographic Hardware and Embedded Systems, p. 1–36, 2021.

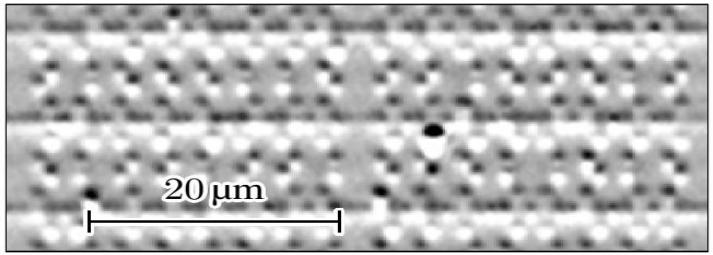
Side-Channel Attacks

- **Many possibilities:**
  - Observation of raw oscillating frequency
    - Applies to RO-PUF and Loop PUF
  - Attack on the Fuzzy extractor
    - Template attacks on ECC
  - Machine Learning attacks supported by SCA
    - Use of noise distribution of the arbiter PUF
    - Observation of internal variables in PUF combinations
- **Countermeasures still required**

1. Merli, D., Schuster, D., Stumpf, F., & Sigl, G. (2011, October). Semi-invasive EM attack on FPGA RO PUFs and countermeasures. In *Proceedings of the Workshop on Embedded Systems Security* (p. 2). ACM.
2. Delvaux, J., & Verbauwhede, I. (2013, June). Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise. In *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on* (pp. 137-142). IEEE.
3. Becker, G. T., & Kumar, R. (2014). Active and Passive Side-Channel Attacks on Delay Based PUF Designs. *IACR Cryptology ePrint Archive*, *2014*, 287.

PUF invasive attack

- **Applies on SRAM PUF**
  - Laser stimulation techniques exploiting the Seebeck effect
    - the off-transistor becomes to conduct under laser shot
    - Provides a current increase



SRAM content read out

Nedospasov, D., Seifert, J. P., Helfmeier, C., & Boit, C. (2013, August). Invasive PUF analysis. In *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on* (pp. 30-38). IEEE.

Outline

- What and Why a PUF ?
- PUF types in CMOS
- Is PUF a panacea ?
- How to make the PUF more reliable?
- How to make the PUF more secure ?
- Conclusions

Conclusions

- **A specific fingerprint for each IC**
- **Strong PUF (with CRPs, no crypto) and weak PUFs (for key generation)**
- **Used for authentication and confidentiality**
- **Two phases: enrollment (with helper data) + reconstruction**
- **Main advantages**
  - Self-generated by the device
  - No reverse engineering and limited tampering
- **Main limitations**
  - Lack of reliability
    - Mandatory post-processing
  - Can be attacked physically and mathematically
    - Protections required
- **ISO Standard for PUF validation**

Standard tests and/or stochastic model

Active discussion at ISO sub-committee 27: (**ISO 20897**)

ISO/IEC JTC 1/SC 27/WG 3 N1233

REPLACES:

ISO/IEC JTC 1/SC 27/WG 3

Information technology - Security techniques - Security evaluation, testing and specification

Convenorship: AENOR, Spain, Vice-convenorship: JISC, Japan

DOC TYPE: working draft

TITLE: **Text for ISO/IEC 1st WD 20897 — Information technology — Security requirements and test methods for physically unclonable functions for generating non-stored security parameters**

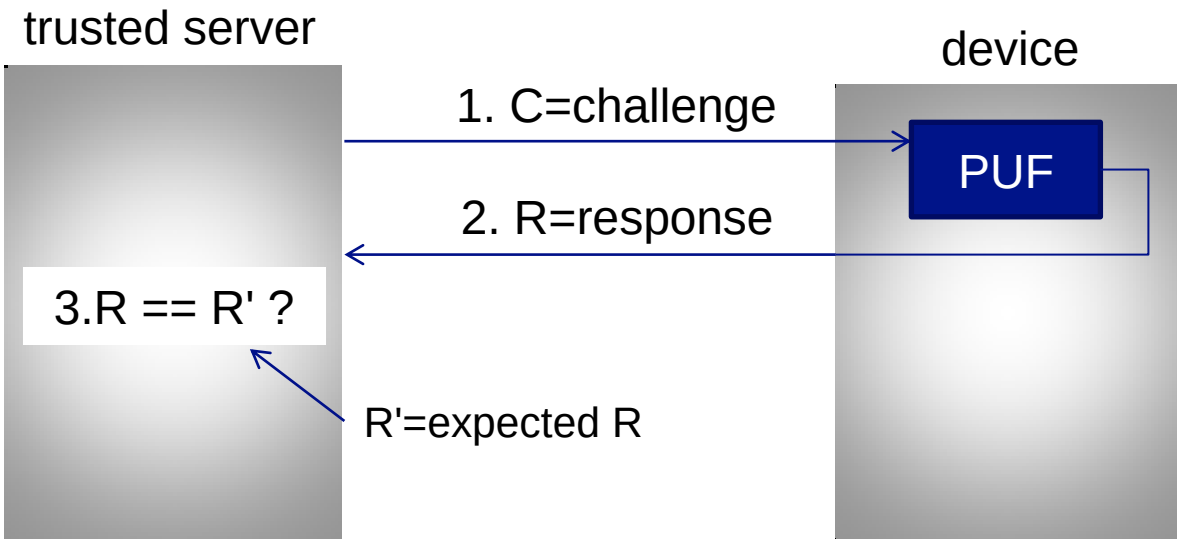
Thank you for listening

Appendix

Authentication

- **Example of CRP protocol (strong PUF only)**

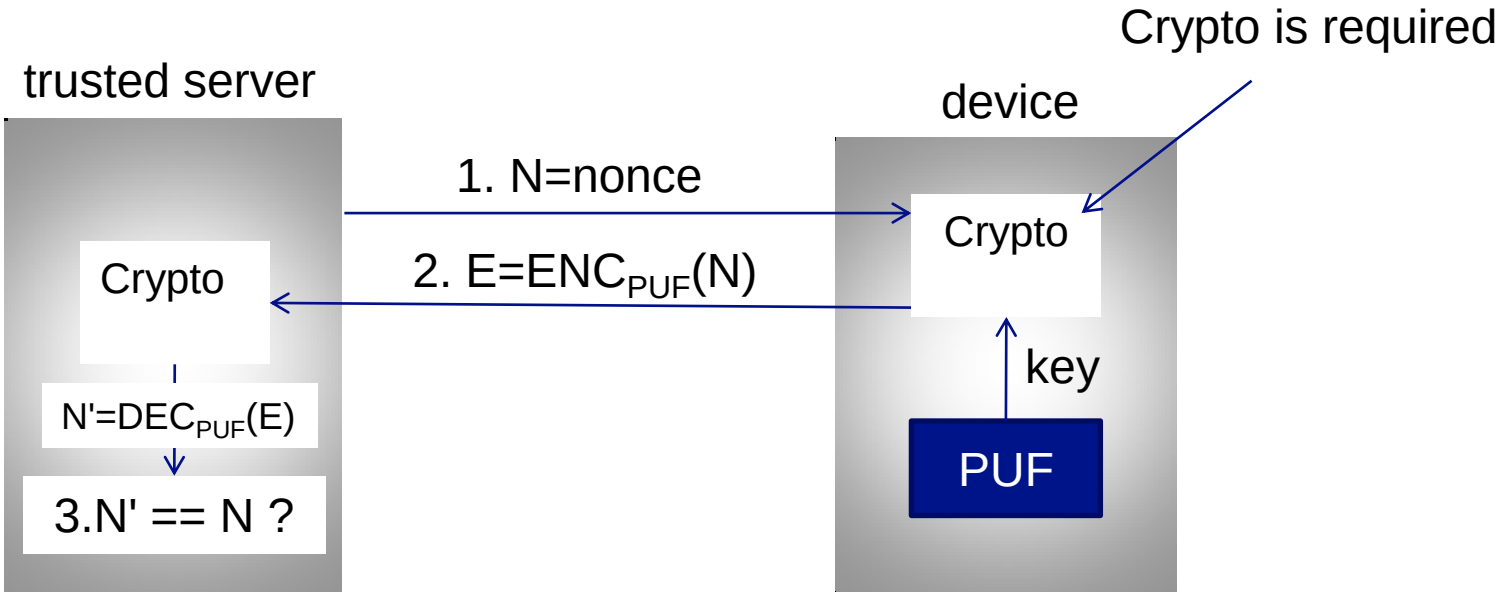
No Crypto is required !



The challenge is never sent twice to avoid replay attacks

Authentication

- **Example of cryptographic protocol**



The nonce is never sent twice to avoid replay attacks