KU LEUVEN

COSIC

The need for hardware roots of trust

Ingrid Verbauwhede
KU Leuven, ESAT - COSIC

erc

Šibenik June 21, 2019

Slides credit: Miloš Grujić, Jeroen Delvaux, Kent Chuang, Adriaan Peetermans, Roel Maes and other PhD students

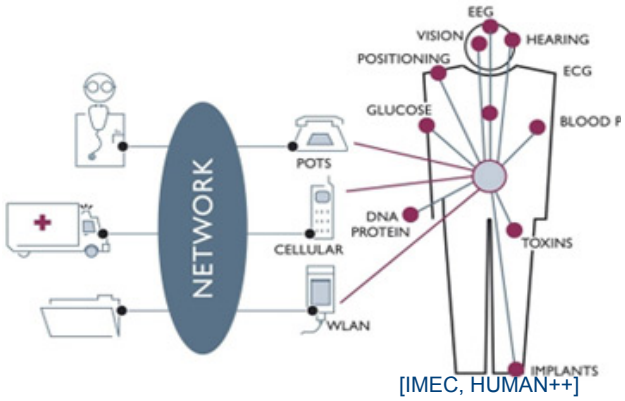
Outline

- Implementation Challenges
- Hardware roots of trust
- PUFs
- TRNG
- Conclusions

KU LEUVEN

Internet of Everything – IOT – Industry4.0 E-…

- Internet of things
- E-health, e-commerce
- E-voting, e-…
- Smart grid
- Big data



[IMEC, HUMAN++]

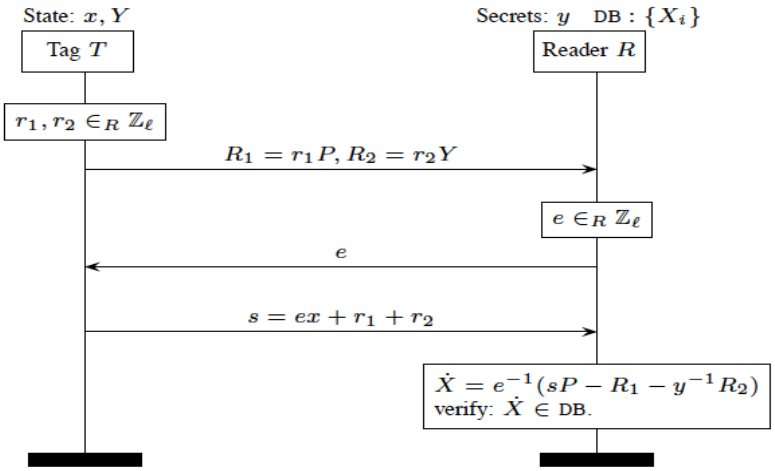
Anything E- or Smart needs security

3

KU LEUVEN

How the crypto protocol paper sees it:

Some calculations are **on** the arrows?

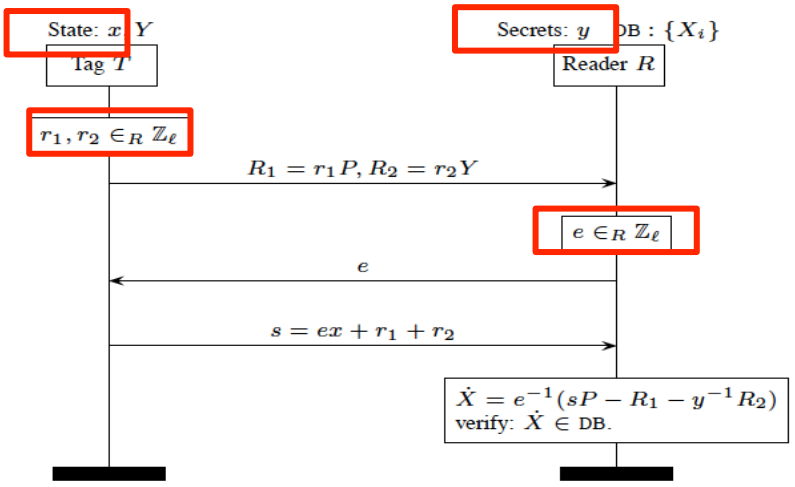


Source: J.Hermans, et al., "Proper RFID Privacy: Model and Protocols," IEEE Trans on Mobile computing, 2014

4

KU LEUVEN

Protocol relies on secrets and random numbers



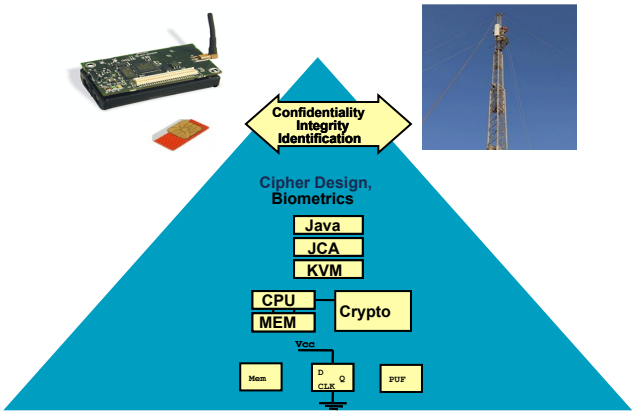
Source: J.Hermans, et al., "Proper RFID Privacy: Model and Protocols," IEEE Trans on Mobile computing, 2014

5

KU LEUVEN

Root of Trust

DESIGN METHODS: DECOMPOSE IN COMPONENTS



- Application: secure communication
- Algorithms: public key, secret key, relies on secret key, post-quantum
- Architecture: Hardware/Software platform, Sancus
- Micro-architecture: crypto co-processors, instruction set extension,
- Logic circuits and (secure) memory
- TRNGs and PUFs

[DATE2007]

*"A root of trust is a component at a **lower** abstraction layer, upon which the system relies for its security."*

6

KU LEUVEN

How to store a secret?

Permanently: e.g. for a master key

- Fuses: large, visible, limited numbers
- Non-volatile memory: extra processing
- Battery-backed SRAM, cumbersome, battery can die
- PUFs: physically unclonable functions

= a cost-efficient replacement technology for secure non-volatile memory (NVM)

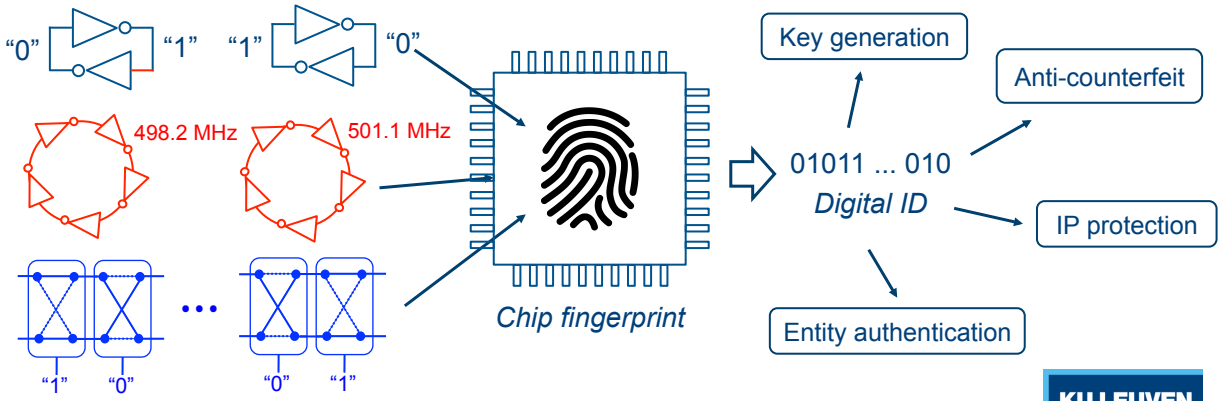
[PhD Jeroen Delvaux]

KU LEUVEN

7

Silicon PUF: An unique fingerprint of a chip

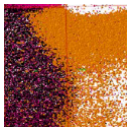
- PUF can be viewed as an *unique* fingerprint of a chip
- Comes from *random process variations*
- Various implementations and applications



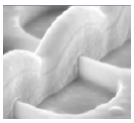
KU LEUVEN

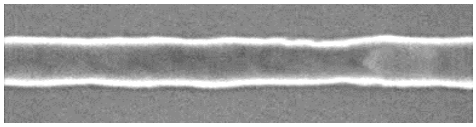
8

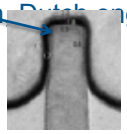
Silicon PUFs - Variability

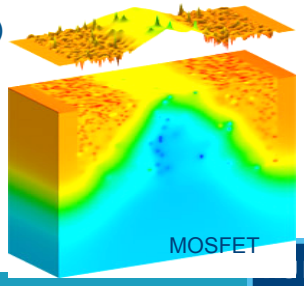


- Silicon Biometrics
- Variability in transistors and interconnect
- In general undesired, except for PUFs
- Random dopant fluctuation
- Tox
- Line edge/width roughness
- Crucial design challenge with CMOS down scaling (Moore's law)
  Pelgrom's law: $\sigma^2 \sim 1/WL$ (Marcel Pelgrom, Dutch engineer)







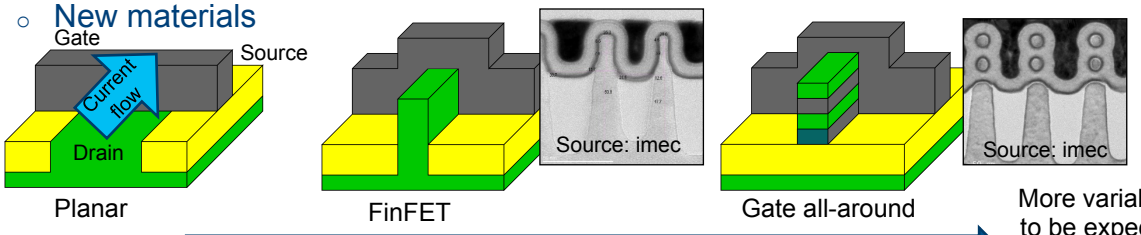


9

LEUVEN

More opportunities brought by scaling

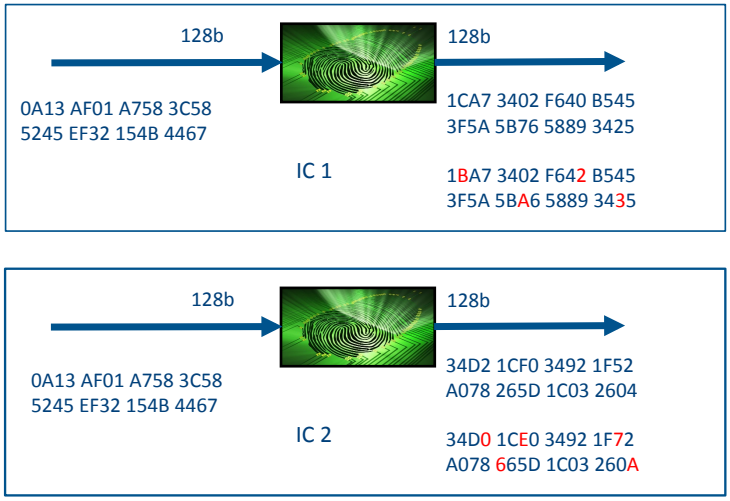
- Even more challenging to manufacture identical devices in scaled technologies
  - Moore's Law
  - 40nm → 28nm →16nm → 7nm → ...
- More variability comes from:
  - More processing steps
  - Decreased size (e.g. 2nm difference → **5%** in 40nm and **30%** in 7nm)
  - New materials



10

KU LEUVEN

The ideal PUF?



Chip-dependent binary function with noisy output

Evaluation 1

≈ 1-15% noise

Evaluation 2

Evaluation 1

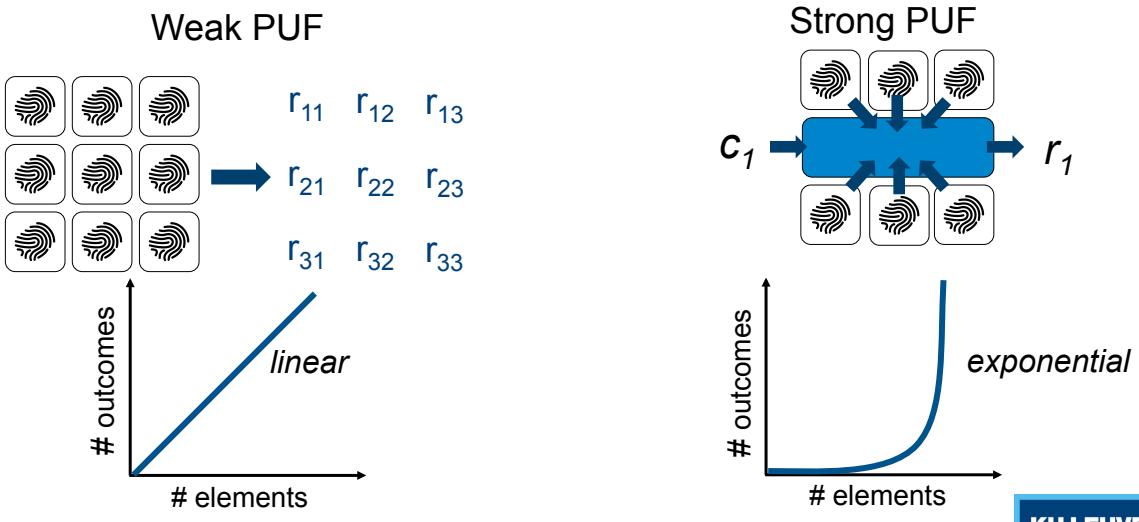
Evaluation 2 ≈ 1-15% noise

IDEAL PUF is without noise

KU LEUVEN

Two design methodologies

Dream 1: IDEAL PUFS don't exist..

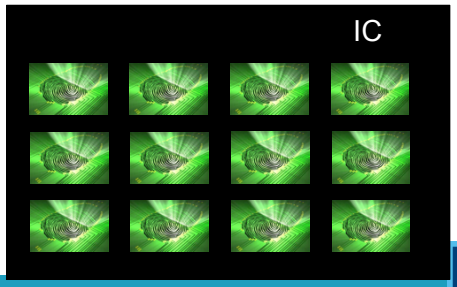


KU LEUVEN

Weak PUF

- An array of identically designed circuit elements
- Each producing 1 (or a few) response bit(s)
- High-quality response bits, i.e., high entropy
- Limited number of bits, e.g., a few 1000s
- Weak because of limited response size, but the best in reality
- E.g., SRAM PUF, spot-break-down PUF
- Typical application: key generation

  E.g. 128-bit AES

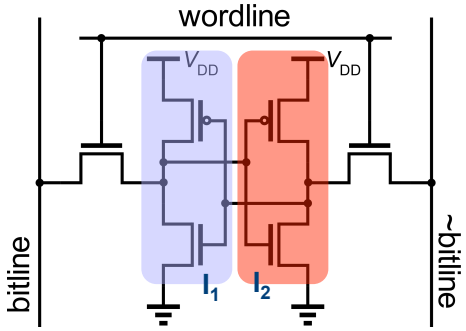


13

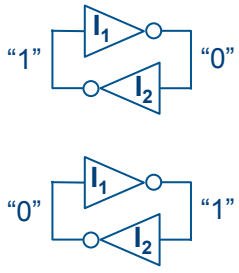
KU LEUVEN

SRAM PUF – a classic weak PUF

- 2D array of 1-bit memory cells
- Variability: ***mismatch*** between the cross-coupled inverters
- Volatile: data is cleared after power-off



6T-SRAM cell



Bi-stable states

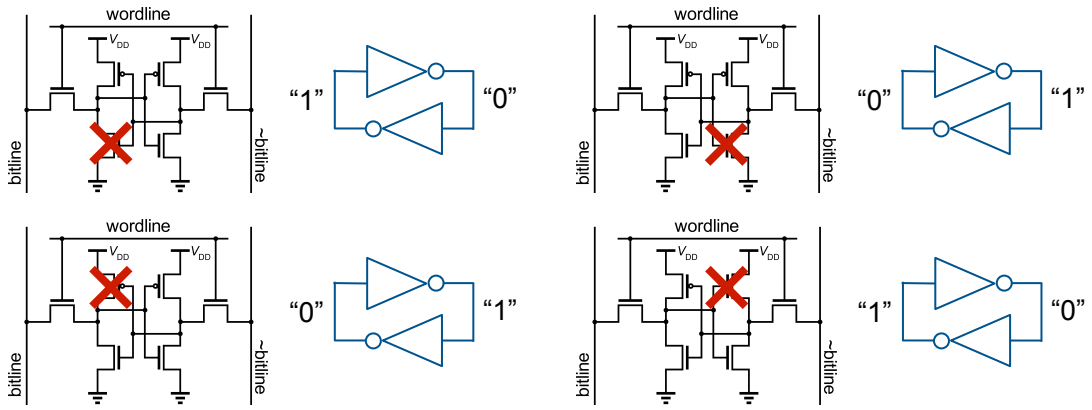
Two possible outcomes after power-up

14

KU LEUVEN

Transistor variations determines PUF bits

- Assume one of the transistors is much weaker than others
- Four extreme cases

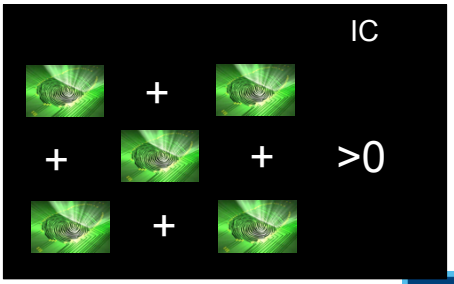


KU LEUVEN

Strong PUF

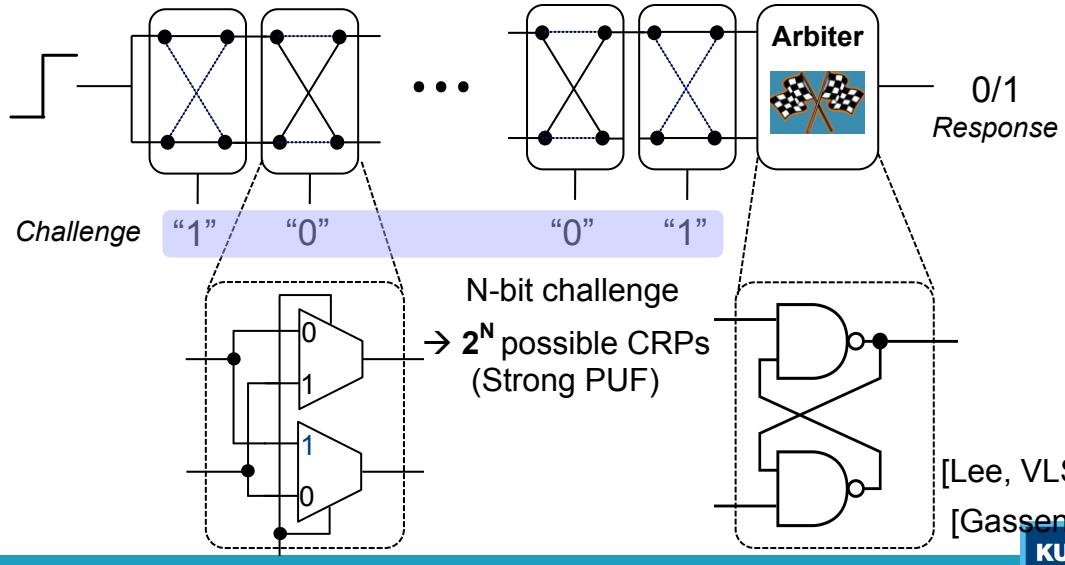
- Finite number of physical building blocks combined with mathematical operations
- E.g., sum of delays, currents, voltages etc.
- Can produce a gazillion of response bits (2^{128}) ➔ Strong
- Low-quality bits: highly correlated, low-entropy

- E.g., arbiter PUF
- Typical application: IC authentication



KU LEUVEN

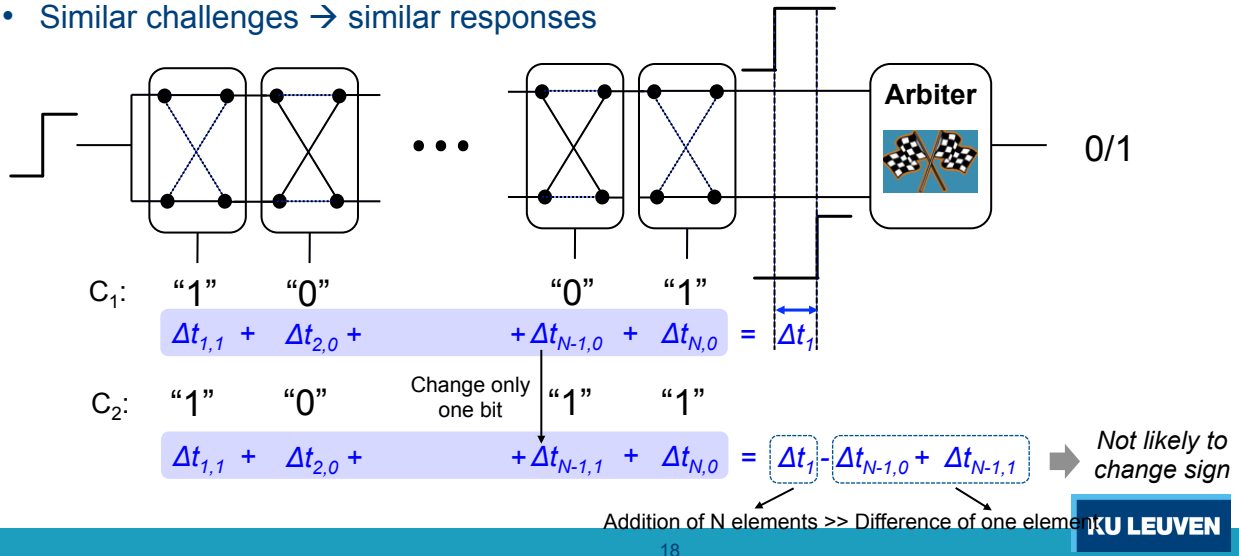
Arbiter PUF – based on timing differences



KU LEUVEN

Arbiter PUF is not an ideal strong PUF

- *Linear* additive structure: sum of delays
- Similar challenges → similar responses

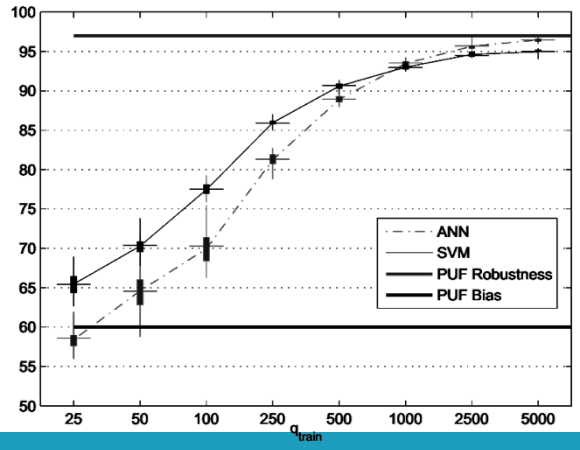


KU LEUVEN

Strong PUF problem: responses easily predicted

- CRPs are highly correlated: low entropy

→ Prone to machine learning (ML) attacks



Experimental results on 65 nm CMOS: only a few 1000 CRPs are sufficient to model the PUF with high accuracy

[Hospodar, WIFS 2012]

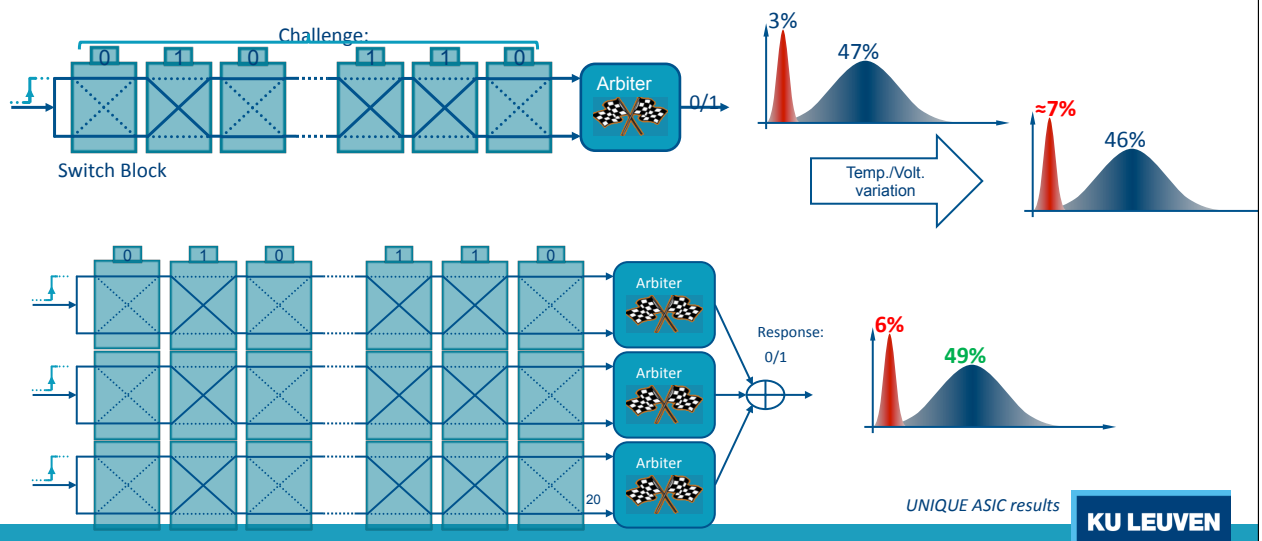
[Ruhrmair, ACM CCS 2010]

KU LEUVEN

19

Arbiter PUFs: XOR Variant

- Arbiter PUF: original MIT work
- UNIQUE project result

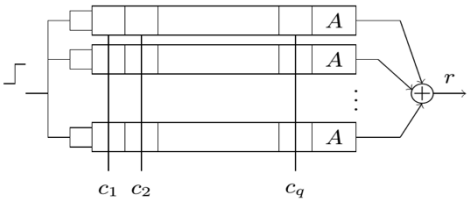


UNIQUE ASIC results

KU LEUVEN

Arbiter PUF – XOR Variant

- XOR the response of multiple chains
- More resistant against machine learning
  - # CRPs in training set ↑
  - Training time ↑
- Unfortunately, noise amplification as well
- Example: Becker et al. at CHES 2015



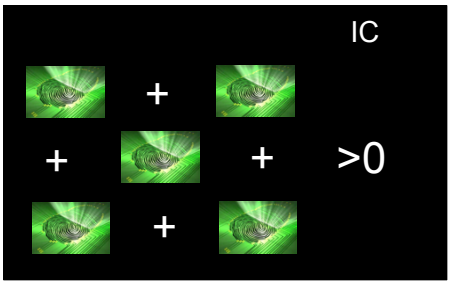
ML Method	CRP Source	Pred. Rate	No. of XORs	CRPs ($\times 10^3$)	Training Time
LR	FPGA	> 99%	3	19.5	51.5 sec
			4	39	139 sec
			5	78	39 min
LR	ASIC	> 99%	3	19.5	26 sec
			4	39	63.5 sec
			5	78	18:09 min

[Ruhrmair, IEEE TIFS 2013]

21

KU LEUVEN

Dream or future research?



Wish a strong PUF:

- Finite number of elements
- Gazillion Challenge Response Pairs
- Non-linear combination to resist modeling attacks: ideally cryptographic functions
- BUT: noise amplification makes output not useful

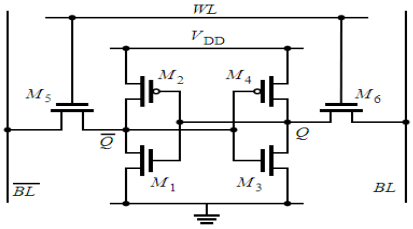
Dream: strong PUF from finite number of elements, resistant to modeling, noise tolerant

Maybe: computational security?

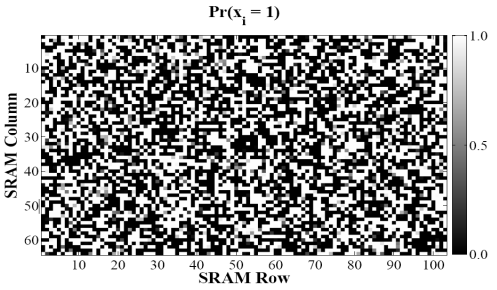
22

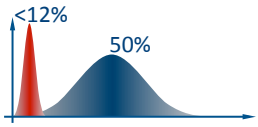
KU LEUVEN

Weak SRAM PUF: Basics

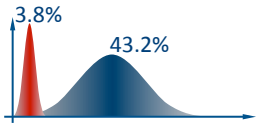


6T CMOS SRAM Cell

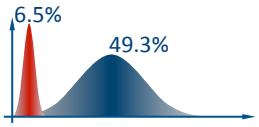




Guajardo et al. 2007, FPGA SRAM temp./volt. var.



Holcomb et al. 2009, Commercial SRAM



Holcomb et al. 2007, Embedded SRAM

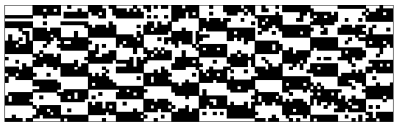
23

KU LEUVEN

PUF behavior of SRAM in commodity micro-controller

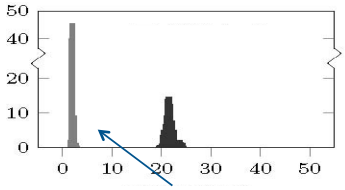
Black box approach (off the shelf micro-controllers)

- PIC16F1825

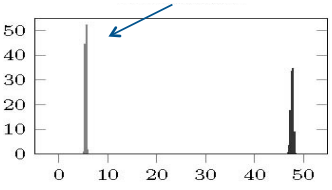


- STM32F100R8

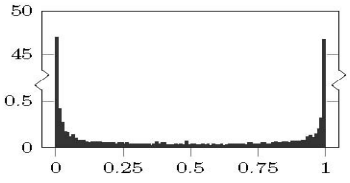
Within and between class HD (%)

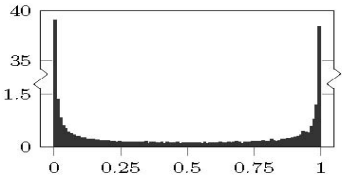


Within Class



Average bit value (%)





[PhD Anthony VH]

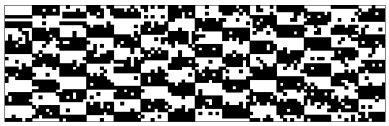
24

KU LEUVEN

PUF behavior of SRAM in commodity micro-controller

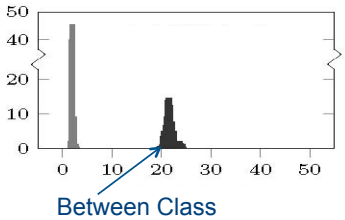
Black box approach (off the shelf micro-controllers)

- PIC16F1825

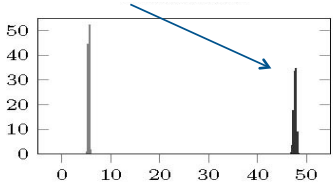


- STM32F100R8

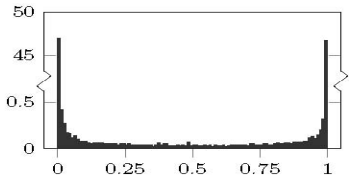
Within and between class HD (%)

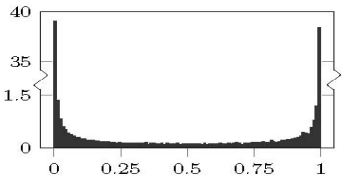


Between Class



Average bit value (%)





Needs post-processing to create key!

25

KU LEUVEN

Reliability

- PUF responses are not exactly reproducible
  - At different time
  - In different environment

PUF response r_1=

#1: 101001001010100 01...

#2: 1011010100**0**01010001...

#3: 1010011**1**0101010001...

KU LEUVEN

26

Short-term reliability (data stability)

- PUF response changed temporarily caused by:
  - Environment change (external)
  - Internal fluctuation

External:
- Temperature
- Supply voltage
- Humidity
- Radiation
- ...

Internal
- White noise
- Flicker noise
- Cross-talk
- Glitch
- ...

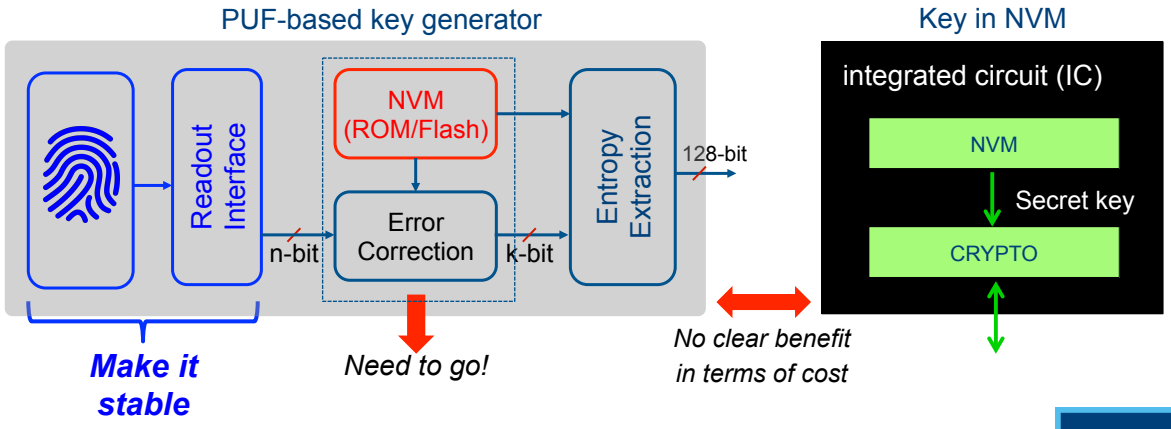
How to improve the short-term reliability?

KU LEUVEN

27

Good reliability is crucial

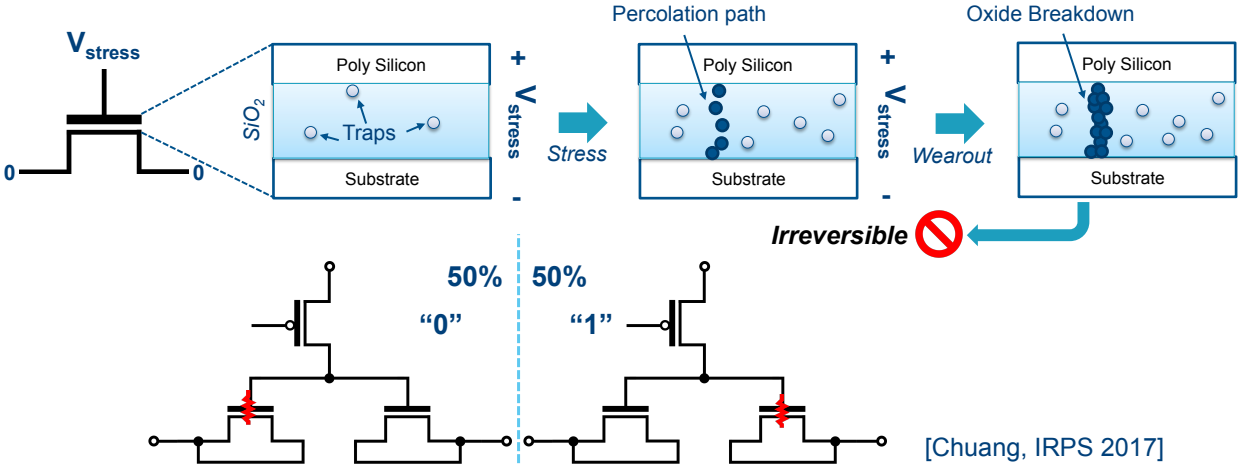
- Error correction codes need to be stored → NVM needed
- Why not just store the key in NVM?



KU LEUVEN

28

Oxide breakdown spots are natively stable



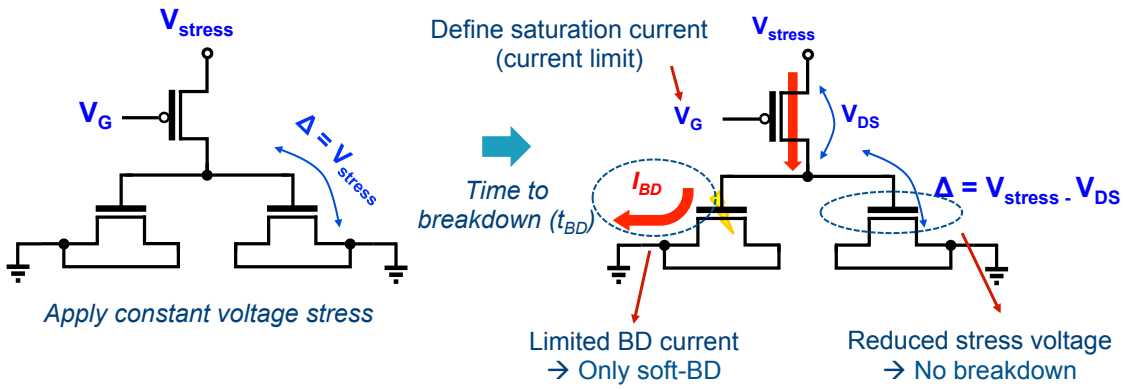
[Chuang, IRPS 2017]

Time dependent dielectric breakdown (TDDB) can be accelerated by voltage → Less time consuming

KU LEUVEN

29

Generating only one BD spot



- Current and voltage are limited by the PMOS selector
- Multiple BD spots are unlikely to happen

KU LEUVEN

30

Spot breakdown PUF circuits

imec

IMEC Breakdown PUF [patent]

Random Spatial Distribution

Si proven stable PUF

- ✔ Low cost, low power for IoT, reliable, small footprint
- ✔ Both random key generation & programmable keys

KU LEUVEN

PUF Usage

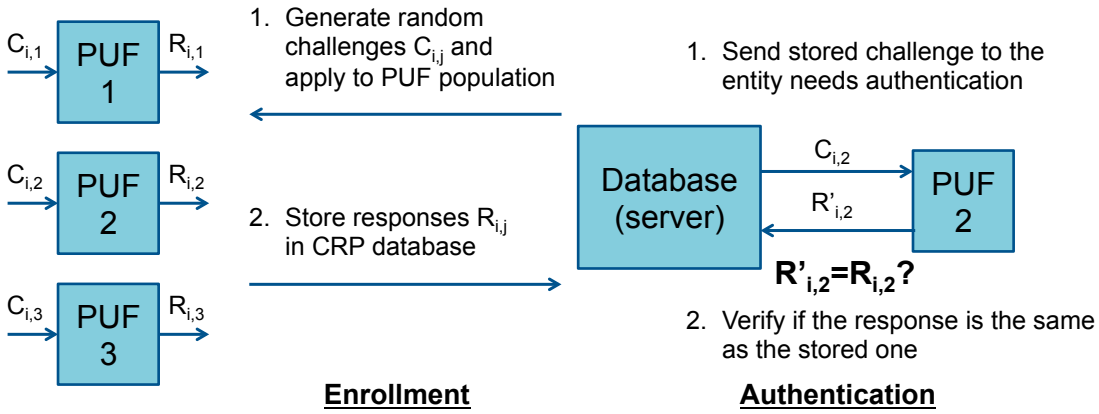
Strong PUF: Entity authentication

Weak PUF: Key Generation

KU LEUVEN

Realizing an ideal authentication scheme

- Entity authentication based on challenge and response



*Needs a huge amount of **uncorrelated** challenge-response pairs (CRPs)*

KU LEUVEN

Protocols for strong PUFs

A large-scale study of strong PUF protocols:

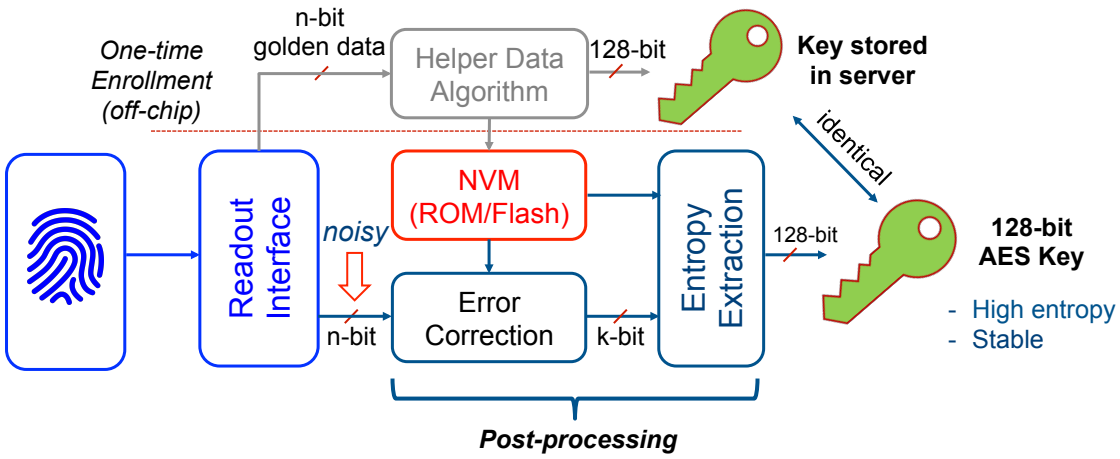
„Secure Lightweight Entity Authentication with Strong PUFs: Mission Impossible?“

[Delvaux, CHES 2014]

[Delvaux, ACM Comp. Surveys 2015]

		Reference II-A	Reference II-B	Basic	Controlled	Bolotnyy et al.	Öztürk et al.	Hammouri et al.	Kulseng et al.	Sadeghi et al.	Reconfiguration	Reverse FE I	Reverse FE II	Converse	Lee et al. I	Jin et al.	Slender	Xu et al.	He et al.	Jung et al.	Lee et al. II	Noise bifurcation	System of PUFs
resources (#6)	Weak PUF	✓	✓	×	×	×	×	×	×	×	×	×	✓	×	×	×	×	×	×	×	×	✓	✓
	Strong PUF[1]	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	MTP NVM	×	×	×	×	×	✓	×	×	×	×	×	✓	×	×	×	✓	×	×	×	×	×	×
	TRNG	×	✓	×	×	×	×	✓	×	✓	×	×	✓	✓	×	×	✓	×	×	×	✓	×	×
	Gen	×	×	×	×	×	×	×	×	×	×	×	✓	×	×	×	×	×	×	~	×	×	×
	Rep	✓	✓	×	×	×	×	×	×	×	×	×	✓	✓	×	×	×	×	×	×	×	×	×
	Crypto (Enc/Hash/MAC)	✓	✓	×	✓	×	×	×	×	✓	✓	×	✓	✓	✓	×	×	×	×	✓	×	×	×
	PRNG	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	$\oplus, =?$	×	✓	×	×	×	✓	✓	✓	×	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×
	$1\times$ interface	✓	✓	×	✓	×	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×
claims	Server authenticity	×	✓	×	×	×	×	×	✓	×	×	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	×
	Token authenticity	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Token privacy	×	×	×	×	✓	×	×	✓	✓	~	×	×	×	×	~	×	✓	✓	✓	✓	×	×
	Leakage resilience (#2)					×	×		×	✓	~				×	✓		×	✓	✓	✓		
	# Authentications	∞	∞	d	d	d	∞	∞	∞	∞	d	∞	∞	∞	d	∞	∞	d	∞	∞	∞	∞	d
evaluation	Noise Robust (#3)	✓	✓		✓	×	~	×	×	✓		✓	✓		×	×	✓	×		×	×	✓	~
	Modeling Robust (#4)	✓	✓	×	✓	×				✓	×	✓	✓				✓	~				✓	×
	PUF Independency (#7)	✓	✓		~					✓		×	✓				×					×	
	Server authenticity (#8)		✓							~		~	✓	×	✓	×		×	×	×			
	Token authenticity (#8)	✓			✓				×	✓		✓	✓		✓		~		×	×		✓	
	Token privacy (#8)								×	✓									×	×			
	DoS prevention (#8)	✓	✓		✓	×	×		×	✓	×	✓	✓		×	×	✓		×	×	×	✓	
	Leakage resilience (#8)								34	✓									×	×			
	Scalability[2] (#9)	✓	✓	✓	✓					×		✓	✓				✓					✓	

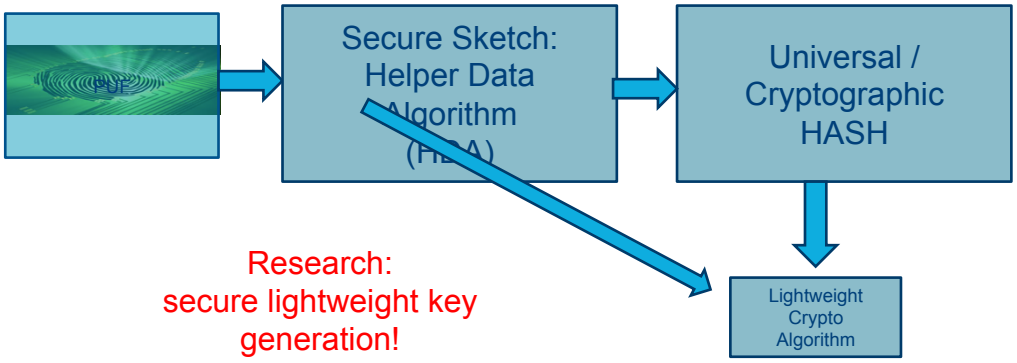
From process variation to a secret key



KU LEUVEN

Light weight solution

- PUF is promised as 'light-weight' key generation for Internet of Things, RFID tags, etc.
- Key generation is larger than lightweight algorithm??



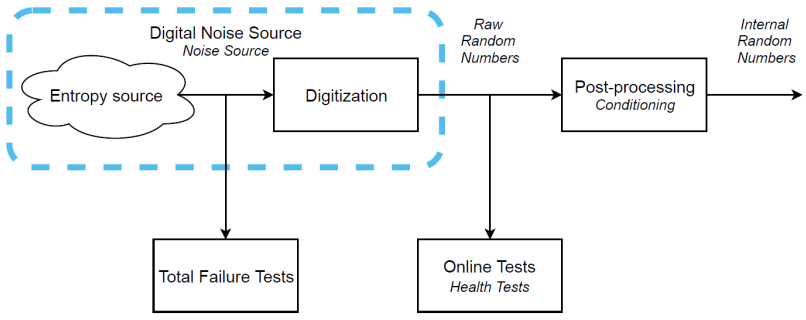
Research: secure lightweight key generation!

KU LEUVEN

True Random Number Generation

KU LEUVEN

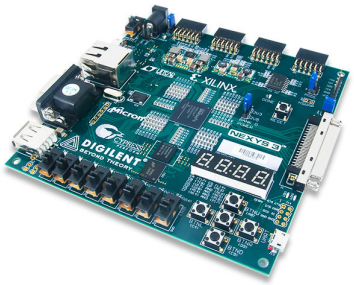
True Random Number Generators



TRNG architecture

KU LEUVEN

TRNGs on FPGAs



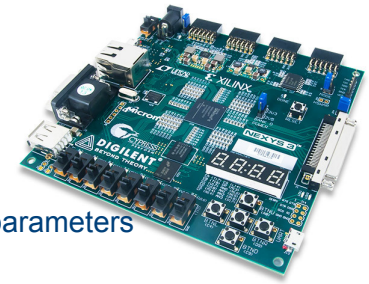
Challenges:

- FPGAs developed to behave in deterministic digital manner
- Vendors' goals: low noise & stable digital behavior → decrease variations of non-deterministic processes → limited number of available noise sources
- TRNG simulations not supported by tools and implementations require special constraints ("set_dont_touch")

KU LEUVEN

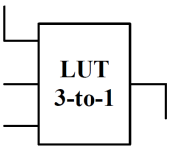
39

TRNGs on FPGAs

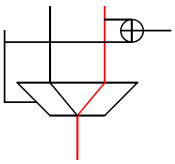


Challenges:

- Portability: different FPGAs → different structure & physical parameters
- Maintain claimed security level on different platforms
- Unrealistic noise assumptions



Fast carry structure on Cyclone IV Intel

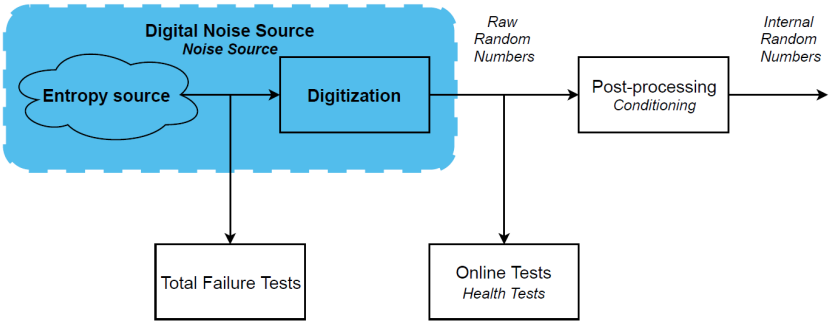


Fast carry structure on Spartan 6 Xilinx

KU LEUVEN

40

Digital noise sources on FPGAs

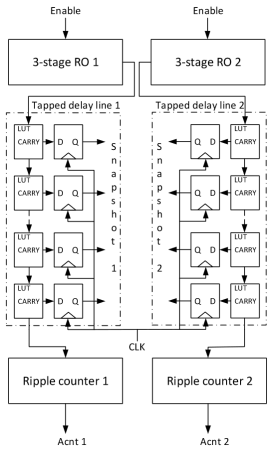


TRNG architecture

KU LEUVEN

Digital Noise Source

- Reliably measuring parameters of randomness generating process
- Sources of randomness:
  metastability & timing jitter
- Solution: on-chip differential jitter measurement
- Isolate contribution of the white noise:
  - differential setup → reduce global noise
  - short measurement time → reduce flicker noise
- Lower bound on jitter strength → conservative entropy estimate



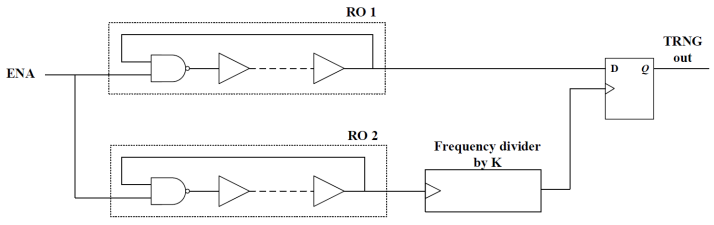
Intel Cyclone IV

[Yang et al., AsianHOST'17]

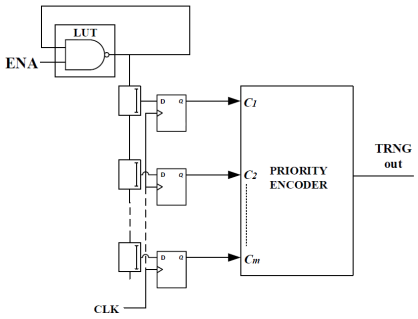
KU LEUVEN

Architectures suitable for FPGAs

- Entropy source: timing jitter
- Time deviation from a periodic signal due to random noise - accumulates *very slowly*
- Relatively easy to model
- Susceptible to temperature and voltage variations



Elementary oscillator based TRNG [BLMT11]

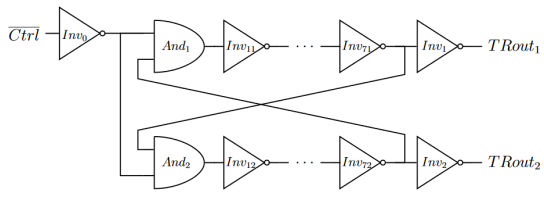


Delay chain TRNG [Rožić et al., DAC'15]

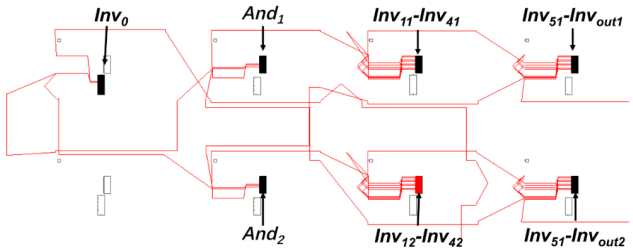
KU LEUVEN

Architectures suitable for FPGAs

- Transition Effect Ring Oscillator (TERO) TRNG
- entropy source: oscillator metastability
- digitizer: asynchronous counter → LSB used as random bit
- manual placement and routing of LUTs
- high throughput (1.3 Mb/s)



[VD10]

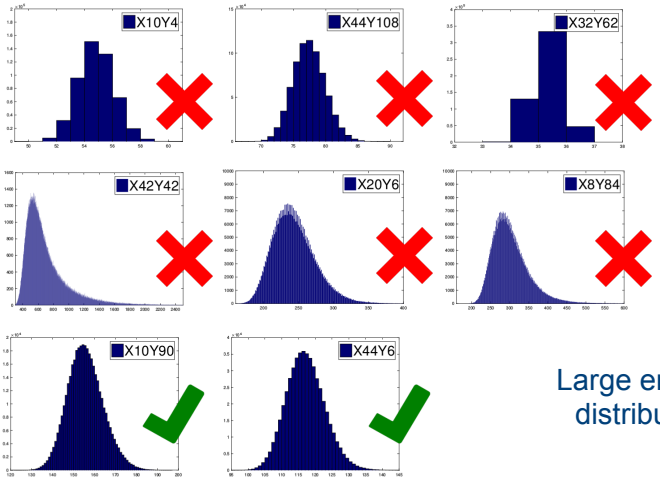


[Cao et al., MWSCAS'16]

KU LEUVEN

TERO TRNG on FPGAs

- Influence of FPGA location on oscillation occurrence



Very few transitions

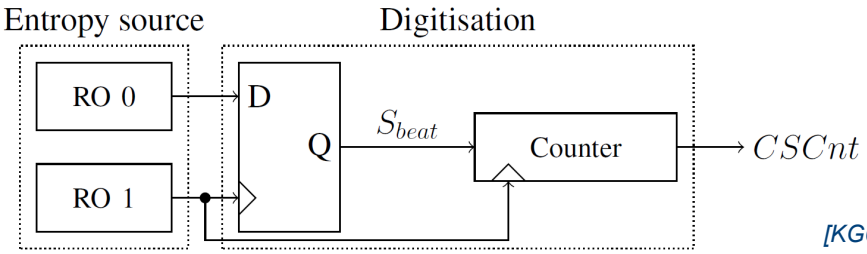
Statistical defects

Large enough number of transitions (> 100), distribution close to Gaussian, pass NIST SP800-22

KU LEUVEN

Architectures suitable for FPGAs

- ➢ Coherent Sampling (COSO) TRNG
- entropy source: timing jitter
- low frequency beat signal S_{beat}
- count period *CSCnt* → LSB is random due to jitter in ROs



[KG04]

✓ High throughput (> 1 Mb/s)

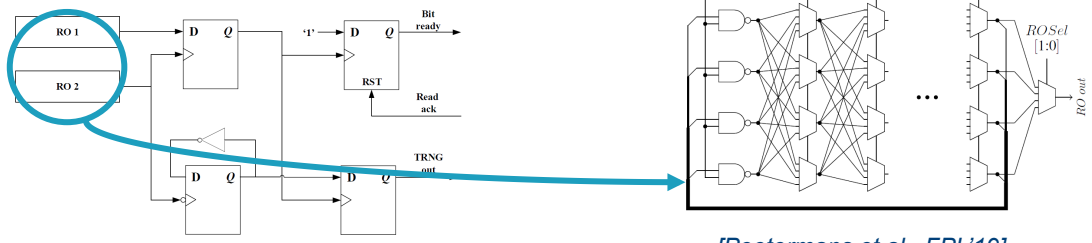
✓ Low area

✗ Extremely large design effort to match frequencies of the ROs

KU LEUVEN

Configurable COSO TRNG

- Reconfigurable ROs
- Controller with dynamic calibration

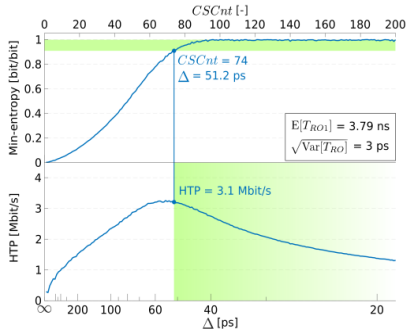


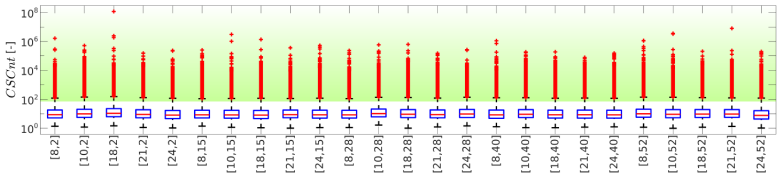
[Peetermans et al., FPL'19]

- ✓ 'Plug & Play' solution → no manual P&R
- ✓ Low design effort and high portability
- ✓ High throughput: 3.3 Mb/s on *Spartan 6* & 1.5 Mb/s on *SmartFusion 2*
- – Price paid: increased latency

KU LEUVEN

Configurable COSO TRNG - COCOSO





FPGA placement [x,y]

- Find optimal point to obtain both high throughput and high entropy – HTP (min-entropy throughput product)
- Sufficient matching of RO frequencies obtained for all FPGA locations

KU LEUVEN

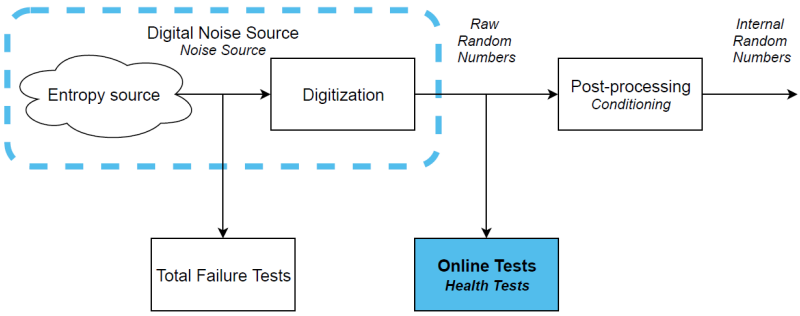
Configurable COSO TRNG – Implementation results

Architecture	FPGA family	Area [DFF/LUTs]	Throughput [Mb/s]	Statistical tests	Design effort
Configurable COSO	*Spartan 6*	39/108	3.3	AIS-31 T6-T8	-
	SmartFusion 2	38/111	1.47	AIS-31 T6-T8	-
Original COSO	*Spartan 6*	3/18	0.54	AIS-31 T8	MP
	SmartFusion 2	3/23	0.328	AIS-31 T8	MP
TERO TRNG	*Spartan 6*	12/39	0.625	AIS-31 T8	MP & MR
	SmartFusion 2	12/46	1	AIS-31 T8	MP & MR
STRNG	*Spartan 6*	256/346	154	AIS-31 T8	MP & MR
	SmartFusion 2	256/350	188	AIS-31 T8	MP & MR

KU LEUVEN

49

Online tests

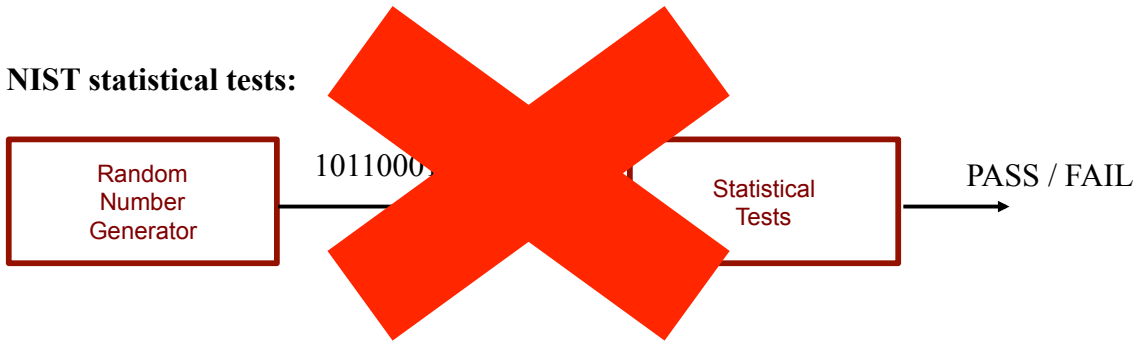


TRNG architecture

KU LEUVEN

50

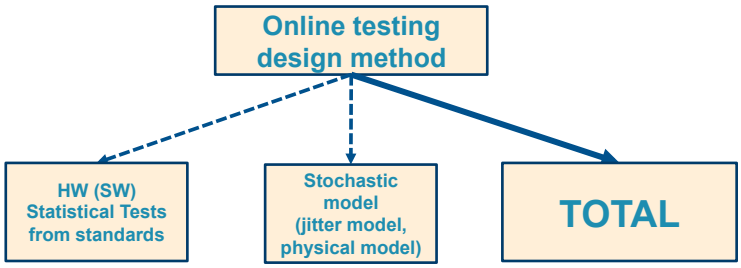
TRNG: the old school



51

KU LEUVEN

Online tests: TOTAL methodology



[Yang et al., DATE'16]

TRNG On-the-fly Testing for Attack Detection using Lightweight Hardware

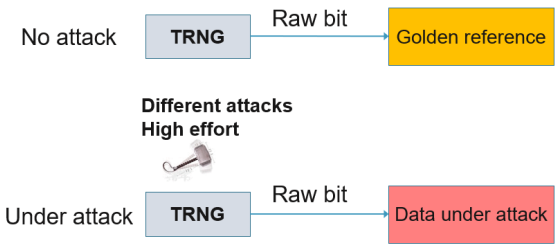
- Experiment oriented design methodology
- Tests tailored for entropy source and attacks

52

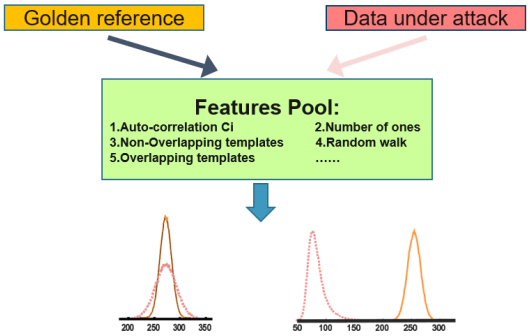
KU LEUVEN

Online tests: TOTAL methodology

- Step 1: Data collection



- Step 2: Selection of useful features



KU LEUVEN

Online tests: TOTAL methodology

- Step 3: Feature verification

➢ Collect more data to verify the bounds
➢ Test the robustness of the selected features

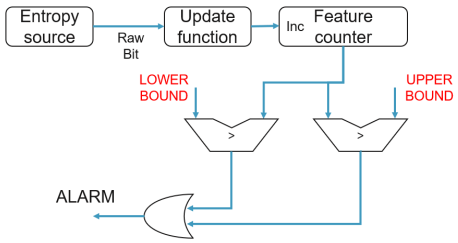
- Step 4: Attack impact analysis

➢ Check the usefulness under different attack efforts

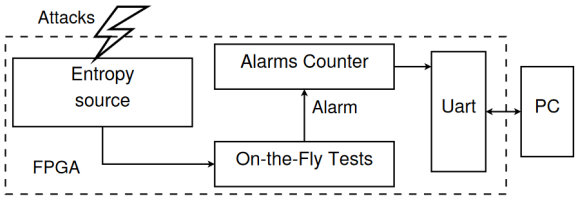
KU LEUVEN

Online tests: TOTAL methodology

- Step 5: HW implementation



- Step 6: HW verification

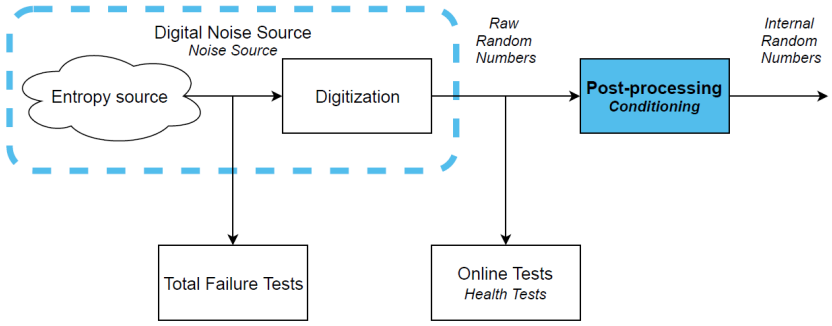


	Spartan 6 FPGA		
	Slices	LUTs	FFs
Sensitive test	9	28	25
Robust test	10	26	22
Combined test	14	42	35

KU LEUVEN

55

Post-processing



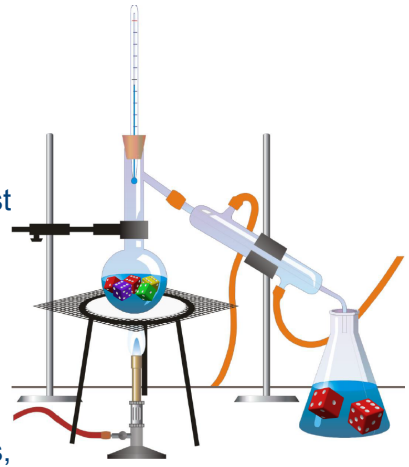
TRNG architecture

KU LEUVEN

56

Post-processing

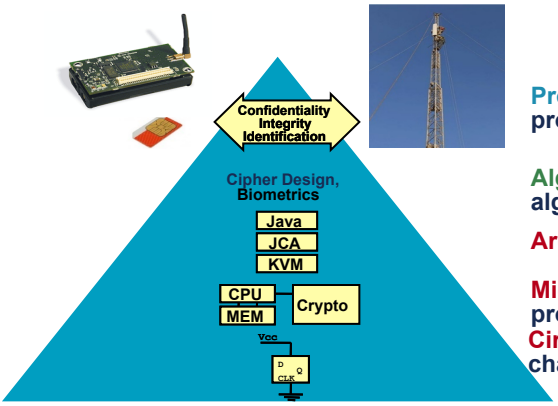
- Cryptographic post-processing
  - High implementation costs i.r.t. TRNG
  - Block ciphers & hash functions
  - Required by AIS-31 standard for the highest security level
- Arithmetic post-processing
  - Low implementation costs
  - Low latency
  - XOR, Von-Neumann debiasing, linear codes, strong blenders



KU LEUVEN

57

Conclusion: Root of trust: one level down!



Protocol: low power authentication protocol design

Algorithm: public key, secret key, hash algorithms, post-quantum

Architecture: Co-design, HW/SW, SOC

Micro-Architecture: arithmetic, co-processor design

Circuit: Circuit techniques to combat side channel analysis, TRNG's PUFs,

At the bottom: PUFs and TRNGs

58

KU LEUVEN

References

[YRG+17] B. Yang, V. Rožić, M. Grujić, N. Mentens, and I. Verbauwhede, “On-chip Jitter Measurement for True Random Number Generators, ” *AsianHOST,* 2017.

[BLMT11] M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux, “On the security of oscillator-based random number generators,” *Journal of Cryptology*, 2011.

[CFAF13] A. Cherkaoui, V. Fischer, A. Aubert, and L. Fesquet, “A self-timed ring based true random number generator,” *ASYNC*, 2013.

[FD02] V. Fischer and M. Drutarovsky, “True random number generator embedded in reconfigurable hardware,” *CHES*, 2002.

[PRV19] A. Peetermans, V. Rožić, and I. Verbauwhede, “A Highly-Portable True Random Number Generator based on Coherent Sampling,” *FPL,* 2019.

[VHKK08] I. Vasyltsov, E. Hambardzumyan, Y.S. Kim, B. Karpinskyy, “Fast Digital TRNG Based on Metastable Ring Oscillator,” *CHES*, 2008.

[CRY+16] Y. Cao, V. Rožić, B. Yang, J. Balasch, and I. Verbauwhede, "Exploring Active Manipulation Attacks on the TERO Random Number Generator," *MWSCAS*, 2016.

[VD10] M. Varchola and M. Drutarovsky, “New high entropy element for FPGA based true random number generators,” *CHES*, 2010.

[KG04] P. Kohlbrenner and K. Gaj, “An embedded true random number generator for FPGAs,” *12th International Symposium on Field Programmable Gate Arrays*, 2004.

[YRM+16] B. Yang, V. Rožić, N. Mentens, W. Dehaene, and I. Verbauwhede, “TOTAL: TRNG on-the-fly testing for attack detection using Lightweight hardware,” *DATE*, 2016.

KU LEUVEN

59