Methods to Enhance the PUF Reliability of Key Generation from PUFs

J.-L.Danger, F. Lozac'h, Z. Cherif

PROOFS'14, Busan, South Korea

Presentation Outline

PUF

PUF reminder

- Device fingerprint
- Avoid Reverse engineering attack of NVM memory but
- Suffers from attacks and reliability problems

This talk:

- presents methods to enhance the PUF reliability
- how to apply them to the “Loop PUF”
- presents the results from real devices (49 PUFs in ASIC 65nm)

Loop PUF

- Set of N identical controllable delay chains of M elements forming a ring oscillator
- For each challenge of MxN bits, the time is measured
- The response is the sorting of the time obtained from the different challenges
- FPGA implementation presented by Cherif et al.[1]

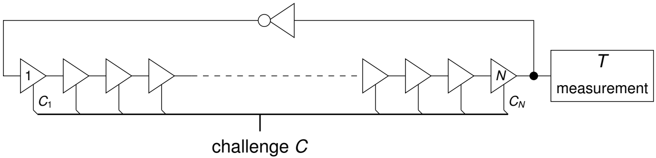


Figure: Example of LPUF composed of N delay chains of 1 element

[1] Cherif et al. [CDGB12]

Example of Key generation with the Loop PUF

1. Choose two **equivalent** challenges (same Hamming Weight)
2. Measure the Time T_1 with Challenge C_1
3. Measure the Time T_2 with Challenge C_2
4. The Key bit is given by

$$KEY\ bit = sign(T_1 - T_2) \tag{1}$$

Reliability issue

- The $\Delta_T = T_1 - T_2$ measurement is highly dependant on the noise level, thus generating potential errors.
- An helper data is very useful to help correcting the errors

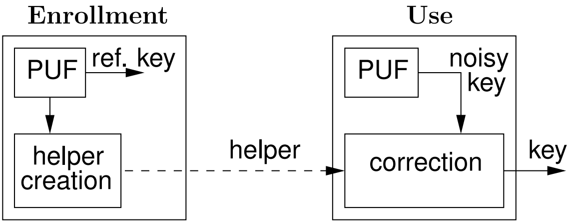


Figure: Use of helpers to correct the key

Studied Methods to improve the reliability

1. **Selecting the challenges**
2. **Enlarging the PUF measurement window**
3. **Increasing the number of measurements**
4. **Removing the most unreliable bits**
5. **Correcting the key**

Selecting the challenges

What are the best challenges to generate one key bit ?
Answer: those having the maximum Hamming Distance
Proof: as

$$\Delta_T = T_1 - T_2 = \sum_{i=1}^{N} t_{i,C1_i} - t_{i,C2_i} \tag{2}$$

Where $t_{i,C1_i}$ represents the time of the elementary delay element i controlled by the challenge bit $C1_i$.

$\Rightarrow$ The total number of elementary delays involved in Δ_T is the Hamming distance $HD(C1, C2)$ between the two challenges.

$\Rightarrow$ For one key bit, choose two **equivalent** and **complementary** challenges (HW=$N/2$, HD=N)

Selecting the challenges : all key bits

The Hamming distance between complementary challenge pair and the other pairs must be as great as possible to avoid **correlated** key bits. **references** :

- ⇒ Use of Constant Weight Codes $A(n, d, w)$, studied in [BSR, CDG$^+$13, CCD$^+$]

Table: Lower Bound of Constant Weight Codes

(n,w) \ d	n/2	n/3	n/4	n/5	n/6	n/7
(12,6)	22	132	?	-	?	-
(16,8)	30	-	1170	-	-	-
(18,9)	34	424	-	-	?	-
(20,10)	38	-	?	13452	-	-
(24,12)	46	2576	15906	-	151484	-
(28,14)	54	-	?	-	-	1535756
(30,15)	58	19210	-	?	?	-

Enlarging the PUF measurement window

- Based on an increase of the measurement time.
- Classical methods for RO-PUF [DV13].

The noise can be reduced when enlarging the measurement window (width = *mw*)

$$\Delta_T = T_1 - T_2 + n(t) \tag{3}$$
$$n(t) \sim \mathcal{N}(0, s^2/mw) \tag{4}$$

but this can increase significantly the key generation time.

Increasing the Number of Measurement

- The principle is to repeat the measurement of Δ_T R times.
- Method very similar to the Time Majority voting presented in [AMS$^+$10].

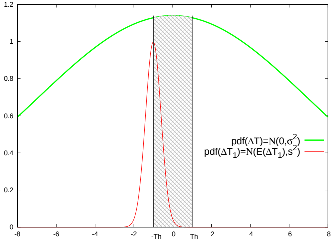
$$n(t) \sim \mathcal{N}(0, s^2/R) \tag{5}$$

- The difference with enlarging *mw* is that the repetition R of the measurement can be controlled dynamically.
- If Δ_T is not above a fixed threshold *Th*, There is a new measurement

Removing the most unreliable key bits

- A helper data is needed in order to indicate the most unreliable bits [HB10].
- the error probability depends on the probability of having $|\Delta_T|$ less than the Threshold $|Th|$.

$$Pr(|\Delta_T| < |Th|) = \text{erf}\left(\frac{|Th|}{\sigma\sqrt{2}}\right) \tag{6}$$



TELECOM ParisTech

Correcting the key

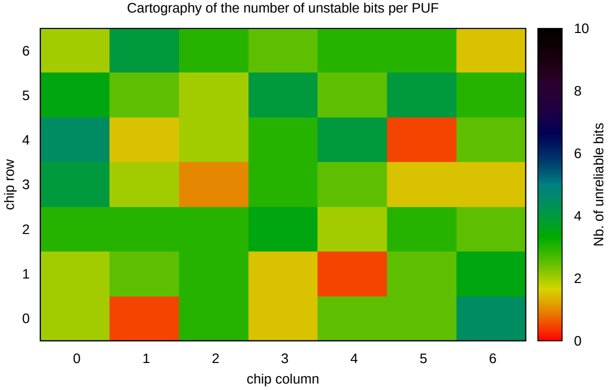
- Well known method explained in many papers [GCvDD02], [MTV09]
- based on error-correction codes (ECC) to correct errors
- The helper indicates the code
- The method can take advantage of the less reliable bits knowledge (case of the Loop PUF). For instance:
  - combine a low-cost Hamming codes
  - and the Chase algorithm [Cha72]

Setup and parameters

- Methods tested on ASIC prototype embedding 49 Loop PUFs.
- 3 result types:
  1. **The error rate.** shows the performance of the key generation procedure in terms of reliability.
  2. **The Key length.** depends on both the number of challenge pairs and the number of ignored unreliable bits *mnib*.
  3. **The key generation time consumption.** influenced by both the measurement window *mw* and the number of unreliable bits *mnib* .

Unstable bits

Cartography of the 49 PUFs:



Key Generation Time Consumption

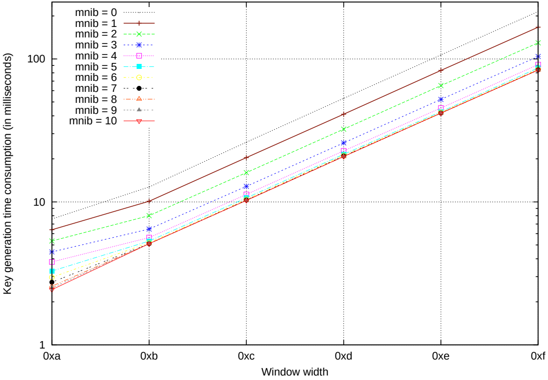


Figure: Impact of *mnib* and the *mw* on the key generation time.

Error Rate Evaluation Without Correction Scheme

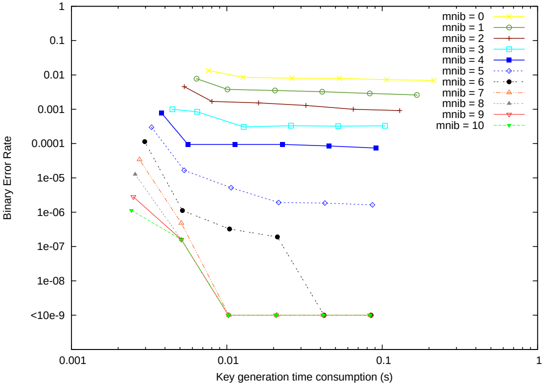


Figure: BER evolution without correction schemes when varying the *mnib* parameter.

Error Rate Evaluation With Correction Scheme

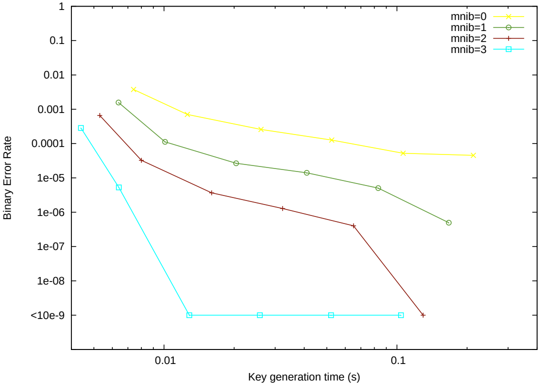


Figure: BER evolution when varying the key length using a correction scheme.

Hardware Implementation Complexity

Table: Hardware complexity of the error correction algorithm: number of occupied slices in Xilinx Virtex 5 technology.

Loop PUF complexity	20	
adaptive key quantification	97	
Key correction complexity	0	235
Total complexity	117	352
BER at 10 ms	10^{-9}	10^{-5}
BER at 100 ms	10^{-9}	10^{-9}
key length	≥ 56	≥ 61

Conclusions

- Five methods are presented to enhance the Loop PUF reliability
- Most of them portable to other PUFs
- Validated theoretically and by experience
- On a 65nm ASIC embedding 49 PUFs
- Interest to eliminate unstable bits for a low-cost and efficient PUF
- In a reasonnable time

Références

[AMS+10] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Berk Sunar, and Pim Tuyls.
Memory leakage-resilient encryption based on physically unclonable functions.
In *Towards Hardware-Intrinsic Security - Foundations and Practice*, pages 135–164. 2010.

[BSR] A. Brouwer, N. Sloane, and E.M. Rains.
Constant weight codes.
http://www.win.tue.nl/~aeb/codes/Andw.html.

[CCD+] Yeow Meng Chee, Zouha Cherif, Jean-Luc Danger, Sylvain Guilley, Han Mao Kiah, Jon-Lark Kim, Patrick Solé, and Xiande Zhang.
Multiply constant-weight codes and the reliability of loop physically unclonable functions.
IEEE Transactions on Information Theory.
To appear (accepted July 2014), DOI: 10.1109/TIT.2014.2359207.

[CDG+13] Zouha Cherif, Jean-Luc Danger, Sylvain Guilley, Jon-Lark Kim, and Patrick Solé.
Multiply constant weight codes.
In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 306–310, 2013.

[CDGB12] Zouha Cherif, Jean-Luc Danger, Sylvain Guilley, and Lilian Bossuet.
An Easy-to-Design PUF based on a single oscillator: the Loop PUF.
In *DSD*, September 5-8 2012.
Çeşme, Izmir, Turkey; (Online PDF).

[Cha72] D. Chase.
Class of algorithms for decoding block codes with channel measurement information.
Information Theory, IEEE Transactions on, 18(1):170–182, 1972.

[DV13] Jeroen Delvaux and Ingrid Verbauwhede.
Fault Injection Modeling Attacks on 65nm Arbiter and RO Sum PUFs via Environmental Changes.
Cryptology ePrint Archive, Report 2013/619, 2013.
`http://eprint.iacr.org/2013/619`.

[GCvDD02] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas.
Controlled physical random functions.
In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 149 – 160, 2002.

[HB10] Maximilian Hofer and Christoph Böhm.
An alternative to error correction for sram-like pufs.
In Stefan Mangard and François-Xavier Standaert, editors, *CHES 2010, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *LNCS*, pages 335–350. Springer, 2010.

[MTV09] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede.
Low-overhead implementation of a soft decision helper data algorithm for sram pufs.
In *CHES 2009, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 332–347. Springer, 2009.