Analysis and Enhancement of Ring Oscillators Based Physical Unclonable Functions in FPGAs

Crina Costea, Florent Bernard, Viktor Fischer, Robert Fouquet

2010 International Conference on Reconfigurable Computing

Analysis and Enhancement of Ring Oscillators Based Physical Unclonable Functions in FPGAs

Crina COSTEA, Florent BERNARD, Viktor FISCHER, Robert FOUQUET
Université de Lyon
CNRS, UMR5516, Laboratoire Hubert Curien
F-42000, Saint-Etienne, France
crina.costea@etu.univ-st-etienne.fr,{florent.bernard, fischer, robert.fouquet}@univ-st-etienne.fr

***Abstract*—The paper analyzes and proposes some enhancements of Ring Oscillator based Physical Unclonable Functions (PUFs) that are used to extract a unique signature of an integrated circuit in order to be used for device authentication purposes and/or key generation. We analyze in more details the concept developed by Suh et al. in 2007. Contrary to what authors claim, we show that the designer of the Ring Oscillator PUFs implemented in FPGAs needs precise control of placement and routing in order to get unique responses and repeatable results for each individual device, especially when the rest of the reconfigurable device should remain upgradable. One main disadvantage of the original design is its high power consumption. We propose a simple improvement that reduces the consumption of the PUF published by Suh et al. by up to 96.6%. Last but not least, we point out that ring oscillators significantly influence one another and can even be locked. This questions the reliability of the PUF and should be taken into account during the design.**

***Index Terms*—Physical unclonable functions, FPGA, cryptographic key generation, IC authentication**

I. INTRODUCTION

Security in integrated circuits (ICs) has become a very important problem due to high information security requirements. In order to assure authenticity and confidentiality, cryptographic keys are used to encrypt the information. Several solutions have been proposed for key generation, each with their upsides and downsides.

Confidential keys can be generated using True Random Number Generators (TRNGs) and stored in volatile or non volatile memories. Saving the confidential key in a non volatile memory inside the device ensures that the key will never be lost and that it will not be disclosed in case of passive attacks. On the other hand, non volatile memories are easy targets for invasive attacks [4]. Volatile memories are typical for FPGAs. Storing the confidential key in a volatile memory permits to erase the memory contents in case of invasive attack detection. This implies the use of a communication channel to transmit the key after device configuration [4]. Communication channels are usually easy to corrupt and information can be easily intercepted. The confidentiality and authenticity of designs are therefore compromised. A solution is backing-up the embedded volatile memory block with a battery. However, it has been proved that battery-backed RAMs content can be read after a long period of storage [2],[1],[13] even if the memory is not powered any more. Thus, the need of generating secret keys inside the IC has become obvious.

An alternative to TRNG for key generation is Physical Unclonable Function (PUF). PUFs are functions that extract a unique signature of an IC that can be used as device-dependent key or device identification code. The main advantage of this principle introduced by Pappu et al. in [10], [11] is the fact that the key does not need to be stored in the device and it is thus harder to disclose. Based on intrinsic physical characteristics of circuits, the extracted signature is impossible to reproduce by a different IC or by an attacker. PUFs work on challenge-response pairs. The challenge is usually a stimulus sent from outside the device, and the response is the signature of the circuit.

The quality of a PUF is determined by its uniqueness and its reproducibility. In order to quantify the quality of a PUF, two types of response variations: intra- and inter-chip variations [14] are used. The intra-chip variation refers to the responses of the same PUF (the same device) at the same challenge, regardless of environmental changes. In the ideal case, this variation should be 0. This means that even with major changes in the environmental conditions, the response of the PUF for a given challenge should always be the same. The intra-chip variation measures the reproducibility of the response. The function must be able to reproduce the same response over and over again, especially in the case of reconfigurable devices.

The inter-chip variation refers to the responses of different PUFs (different devices) at the same challenge. Ideally, this variation should be of 50%, meaning that the PUF should produce uniformly distributed random bits. If this variation is close to 50% then the uniqueness of the responses is guaranteed.

Several concepts have already been introduced until now. PUFs based on SRAM cells [7], on unstable states of flip-flops [8], functions exploiting transistors threshold, differences in the silicon layers of the device [14],[6],[9].

In this paper, we focus on the "reconfigurable" property of a reconfigurable device. Indeed, if the PUF changes when the device is reconfigured, the uniqueness and reproducibility of a PUF are strongly questionable. We analyze and propose some enhancements of the concept introduced in 2007 by Suh et al. [14]. We selected this principle for our experiments, because it is one of the most suitable for implementation in FPGAs, independently from the technology. The PUF uses a relatively

978-0-7695-4314-7/10 $26.00 © 2010 IEEE
DOI 10.1109/ReConFig.2010.63

high number of ring oscillators (ROs) in order to emphasize the intrinsic characteristics of ICs and extract the signature. The principle is based on the fact that the frequency of ROs depends on gate and routing delays determined partially in an uncontrolled way by the manufacturing process.

In the first part of our work, we had to deal with implementation issues related to the mapping of the PUF to various FPGA technologies. We found out that, contrary to what original authors stated, the placement and routing constraints play a very important role in the design of the function, especially if one wants to obtain sufficient inter-chip variability. The precise control of the initial phase of ROs and careful design of frequency comparators is another important issue that determines the precision of the function and thus reduces intra-chip variations. This was not discussed before. The main disadvantage of the original design is the high power consumption. We propose a simple modification enabling significant power economy. During our experiments we observed a very important phenomenon that has a significant impact on the generated results and that was completely neglected in the original design: the existence of a mutual dependence between the ROs can lead sometimes to their mutual locking in FPGAs. It is essential to take into account this unavoidable behavior of ROs in the PUF design.

The paper is organized as follows: Section II deals with the PUF design issues and with the first problem stated: the need of manual placement and routing of the design. Section III presents results of implementation of the RO PUF in main FPGA technologies and analyzes the quality of the PUF in relationship to the selected technology and the quality of the evaluation board. It also evaluates the impact of the mutual dependence of rings on the reliability of the PUF. Section IV proposes some important enhancement of the function and finally, Section V concludes the paper.

II. PUF Design Issues

A. Principle of the PUF and its implementation in FPGA

In the principle of the PUF published in [14] that was selected for our experiments, N identically laid-out ROs are placed on the IC. Slight differences between their frequencies will appear because of the unavoidable differences in the silicon layers of the semiconductor device caused by the manufacturing process. Pairs of oscillators are chosen one after another and their frequencies compared. The response of the PUF is equal to 1 if the first RO is faster and 0 otherwise.

The RO PUF in Fig. 1, as not many details were given by the authors in [14], is realized using 32 ROs controlled by an enable signal for all selected technologies (ALTERA, XILINX and ACTEL). Two counters are employed for frequency measurements. They count the rising edges of the clock signals generated by the ROs. When a preselected value has been reached for one of the counters, the oscillations and counting stop for both signals and the results are evaluated by a comparator. If the first RO of the pair is faster, the result of the comparison is '1', else '0'. The comparator and the counters are asynchronous to the global clock. The counters

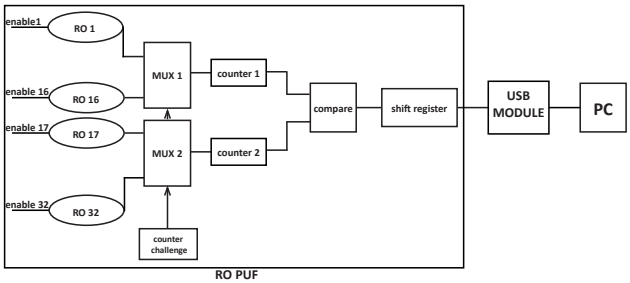

Fig. 1. RO PUF Scheme

are controlled by the output clock of ROs and the comparator contains combinatorial logic and an asynchronous latch in order to react immediately at the moment the counting should stop. The combinatorial logic is required in order to obtain accurate evaluations of the frequencies. Once the comparison output was obtained, the oscillations and counters can restart using the same control signal (enable).

The output of the PUF presented in Fig. 1 is 1 bit wide. In order to obtain a wider response, we use a shift register with a 16 bit output. To get more responses at once, the challenge generator is included in the design. It is a simple 8-bit counter incremented after each comparison of the frequencies. For each value of this counter, two different oscillators are chosen for comparison. They are separated in two groups of 16 (group A and group B). Every oscillator in group A is compared to all oscillators in group B. This way, we obtain 16x16=256 different challenges thus 256 responses of 1 bit for each device. For simplicity, we consider that each IC delivers 256-bit responses. The generated bit-streams are sent to the PC using a USB interface. For this reason, a small USB module featuring a Cypress EZ-USB device was connected to the evaluation board containing FPGA. A 16-bit communication interface with this module was implemented inside the FPGA. A Visual C++ application running on the PC reads the USB peripheral and writes data into a text file. For both ALTERA and XILINX technology, delay elements of the ROs are implemented using Look Up Tables (LUTs). Finally, one NAND gate that is necessary to obtain oscillations, closes the loop and provide the structure with an enable signal.This configuration allows the use of either an odd or even number of delay elements. Thus, the ROs used in the design are made of 7 delay elements and one NAND gate in order to fit the ring into one LAB. In ACTEL technology, the oscillators employ 7 AND2 gates as delay elements and a NAND gate as a control gate.

B. Implementation results and tests

As resources in FPGAs have increased in volume and performances, integrated development environments (IDE) are charged with automatic placement and routing. This is very convenient in common applications. The design can be translated into a significant number of logic elements, thus with automatic placement and routing, the user gains time and the surface of the IC is used at its maximum capacity. The

compiler used by these environments calculates the optimal disposition of logic cells.

The RO PUF presented in [14] exploits, as any other PUF, the intrinsic characteristics of an IC. By definition, the position of the PUF on the IC determines the set of challenge-response pairs. RO placed in LAB A will most probably not oscillate at the same frequency as RO placed in LAB B. In order to use this PUF for the authentication process or as a secret key generator, one must be sure that the response of the PUF will be the same under any environmental conditions and even more important, after each reconfiguration of the device.

The first tests were conducted on ALTERA DE1 boards including Cyclone II EP2C20F484C7N FPGA. We used the PUF to authenticate the devices available in the laboratory. It allowed us to identify a given IC between the 13 available.

As expected, we were able to perform this operation on all the devices: both inter- and intra-chip variations were in normal ranges. However, authors insisted that the design needed no further placement and routing constraints in [14], page 3: "[...] there is no need for careful layout and routing. For example, the paths from oscillator outputs to counters do not need to be symmetric". If it is clear that ROs must be identically laid-out (which is achieved thanks to a macro in [14]), it is questionable that extra logic arround the ROs needs no placement and routing constraints.
We therefore moved on to evaluating the impact of major changes of the extra logic. Thus, we added a counter to the design. This counter was unrelated to the function, therefore, it should not have influenced the response of the PUF. Contrary to all expectations, instead of obtaining almost the same response of the PUF (i. e. obtaining small intra-chip variation), the device gave completely different response so that presumably low intra-chip variation after addition of the additional logic was almost as high as an ideal inter-chip variation: 48,8% of the response bits changed. We went even further with our experiments in order to see if minor changes in the architecture had any influence on the response. Thus, we kept the initial PUF and sent its output, as before, towards the USB module and additionally, towards the 7 segments display available on ALTERA DE1 board. We obtained once again the intra-chip variation (that should be close to zero) comparable in size to an inter-chip variation (close to 50%). Table I presents the responses of the same PUF (same placement of the ROs) in the three experimental conditions described above.

Response n° 1 (only the PUF)	031f031f031f0005573f031f031f471f 573f031f011fd73ff7bf431f0117031f
Response n° 2 (PUF and counter)	010000004df20000000045a04de245e0 45a0fff70000000045e0fff745e045e0
Response n° 3 (only the PUF and an extra output)	0007600f7eef7eef200fffff724f704f 600f0007600f7eef0005000772ef0007

TABLE I
RO PUF WITHOUT SPECIFIC PLACEMENT & ROUTING CONSTRAINTS

C. Imposing placement & routing solution for response stabilization

The results previously presented prove that the optimization performed by the compiler implies different placement and routing for the frequency comparator after each recompilation. Data in Tab.I proves that without placement and routing constraints for this part of the function, the RO PUF concept cannot be used in reconfigurable devices and is unquestionably unreliable.
As we need to have the minimal intra-chip variation, this is of extreme importance. Different placement implies different PUF and different intrinsic characteristics to exploit. In both targeted processes (authentication and key generation) this is not acceptable. Therefore, imposing constraints on both placement and routing is mandatory in order to obtain a PUF independent on architecture modifications in a reconfigurable device. We note that while ALTERA and XILINX technologies allow the user to impose placement and routing constraints, ACTEL technology permits to constrain only the placement. Once we dealt with this problem, the PUF provided excellent and expected responses. In Table II, there is an example of response with specific placement and routing constraints with few differences printed in bold.

Response n° 1 (only the PUF)	ea09ebf9ea09ebfbea09e**b**59ebfbeb79 ea09ebfbfffffbffeb79ea49ebfb0001
Response n° 2 (PUF and counter)	ea09ebf9ea09ebfbea09ea**6**9ebfbeb79 ea09ebfbfffffbffeb79ea49ebfb0001
Response n° 3 (only the PUF and an extra output)	ea09ebf9ea09ebfbea09ea59ebfbeb79 ea09ebfbfffffbffeb79ea**1**9**f**bfb0001

TABLE II
RO PUF WITH SPECIFIC PLACEMENT & ROUTING CONSTRAINT

III. OBSERVATION OF THE PUF IN VARIOUS TECHNOLOGIES AND ENVIRONMENTAL CONDITIONS

A. Observing the PUF in ALTERA, XILINX and ACTEL technologies

In order to compare fairly different FPGA technologies, we would need a huge number of devices for all of tested families. Unfortunately, we had only cards with thirteen Altera Cyclone II and four Cyclone III devices, five Xilinx Spartan 3 and three Xilinx Virtex 5 chips and 5 Actel Fusion FPGAs at our disposal. For this reason, we used the biggest group of 13 Cyclone II FPGAs to verify the inter-chip variation. The obtained value was 48% in average, which is close to the ideal value of 50%.
The intra-chip variation of the PUF was first tested on four ALTERA Cyclone III EP3C25F256C8N ICs. These experiments were conducted under variable temperature and voltage conditions. Results have been prevailed for a temperature range from 30 to 80° Celsius (see Fig. 2) and a voltage range from 0.9 to 1.3V for the nominal voltage of 1.2V (see Fig. 3). In these two figures, the distribution of the number of bits (x-axis) that changed between two different responses from the

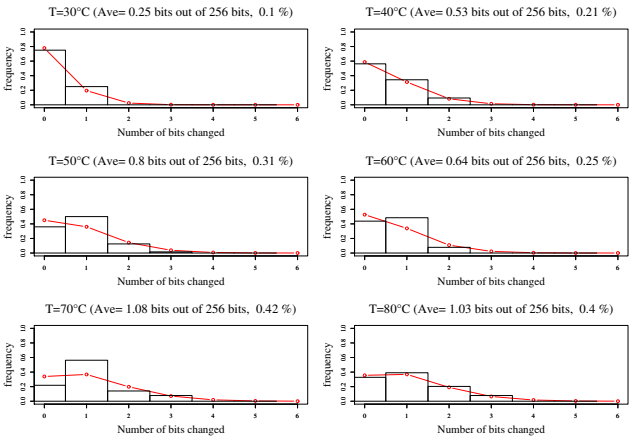

Fig. 2. Intra-chip variation on the same Cyclone III EP3C25F256C8N FPGA for various temperatures

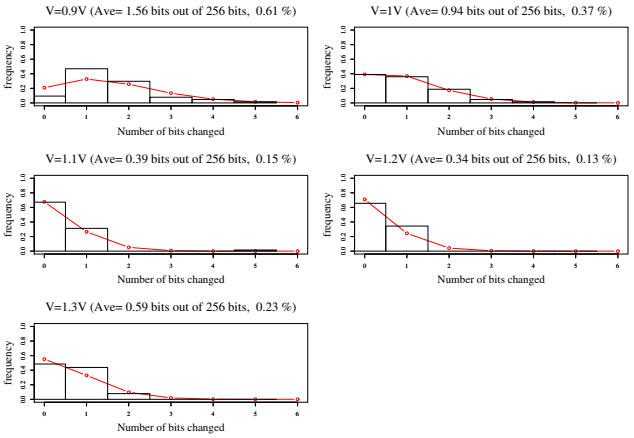

Fig. 3. Intra-chip variation on the same Cyclone III EP3C25F256C8N FPGA for various voltages

same PUF is shown as a histogram. The doted line presents the binomial distribution $\mathcal{B}(n,p)$, where $n = 256$ is the number of bits of the response and p is the number of bits that changed between two responses in average.

Experiments show that intra-chip variation increases when temperature increases. Furthermore, the behavior of the PUF drifts from the binomial distribution. This is probably caused by the influence of thermal noise which is more important as temperature increases and superposes a normal distribution on the binomial distribution. The PUF was tested also on XILINX Spartan 3 XC3S700AN and on XILINX Virtex 5 XC5VLX30T devices. Experimental results confirm the fact that placement constraints are mandatory. The intra-chip variation was lower for Spartan 3 than the one obtained in Cyclone III (0.05% and even 0% in certain cases).

For ACTEL technology, the tests were performed on ACTEL Fusion M7AFS600 FPGA. The intra-chip variation reaches 13.5%! This technology presents the highest intra-chip variation which is unexploitable for IC authentication. One of the reasons for which we think the intra-chip variation is

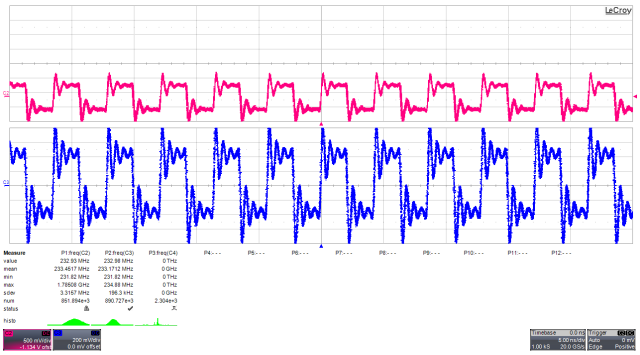

Fig. 4. Locked ring oscillators. Trigger on top signal.

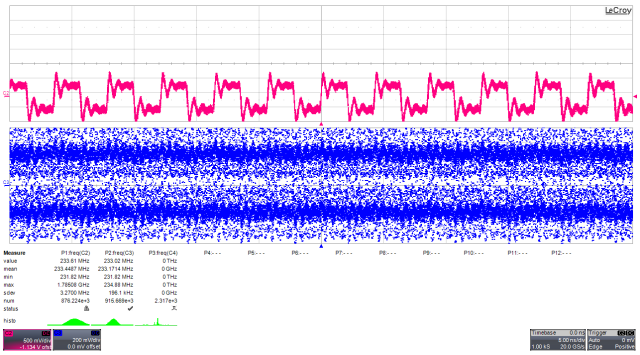

Fig. 5. Unlocked ring oscillators. Trigger on top signal.

higher for these boards is the fact that they present more noise than the other ones. We observed a peak at 20MHz in the core voltage spectrum, caused probably by some internal oscillator embedded in ACTEL FPGA. Similar peak was not detected in other technologies. These results show that the quality of this PUF is strongly related to the quality of the device and the board. In this precise case, the intrinsic characteristics of the IC are overwhelmed by the noise and the results are far from being ideal.

B. PUF and mutual relationship between rings

While studying properties of ROs, we observed that ROs influence one another sometimes to an unexpected extent. If the ROs are identically laid-out, their oscillating frequencies are almost the same. The differences are caused by the intrinsic characteristics of the IC as well as by the noise. If the frequencies are so close that the current peaks caused by rising and falling edges overlap, the ROs can lock and oscillate at the same frequency, either in phase or with a phase shift.

Figure 4 shows output waveforms of two ROs that are locked (both waveforms are visible) and Fig. 5 shows ROs that are not locked (the second waveform is not observable). Note that the oscilloscope was synchronized on the first waveform.

One can argue that the mutual dependence of rings could be caused by the FPGA input/output circuitry. In order to avoid influencing the results by outputting the signals from FPGA, we used simple circuitry permitting to detect the locking. The

signals delivered by the two ROs were fed into the D flip-flop: one of them to the data input and the other to the clock input. If the output of the flip-flop is constant ('1' or '0') then the oscillators are locked.

The observation of numerous rings confirmed the fact that the mutual dependence of oscillators is big enough for them to lock and oscillate at the same frequency. We could also observe that independent oscillators at moment t_0 can become locked at moment t_1 if external conditions (temperature, voltage) present even slight changes.

If the challenge sent to the PUF selects a pair of oscillators that are locked, then the response is no longer based on intrinsic characteristics of the IC. Frequencies are identical, therefore the bit should not be valid. This depends however on the method employed for frequency measurement. In our design, if the ROs are locked with a phase shift, the rising edge of the RO with an advance will always be counted before the rising edge of the second RO. Thus, the result of the evaluation will always show that this RO has a greater oscillating frequency. If the oscillators are locked without a phase shift, the two counters will finish at the same time and the bit will be declared not valid.

This rises an important question on the quality of the response delivered by the PUF. If the oscillators are locked at the moment we compare their frequencies the response is deterministic and no longer based on intrinsic characteristics of the device.

Identically laid-out oscillators request manual placement of the delay elements, as argued earlier. This means that the user will impose the position of the ROs on the device. Experimental results on the PUF showed that in certain configurations the distribution of '1's and '0's in the response was not uniform at all. In other configurations, the response presented a better distribution of values. Thus, we studied the locking phenomenon for ROs in certain configurations occupying the smallest area possible. These configurations were chosen because the surface of the PUF should be relatively small comparing to the rest of the logic implemented in the device. Moreover, the PUF needs to be implemented in an isolated zone so that additional logic has only minimum influence on the response.

We tested two particular configurations, ROs grouped in a compact block and ROs placed on two columns, face to face. Experimental results show that in the first configuration, there are more chances to have locked ROs. The most probable explanation for this phenomenon is that ROs placed close to each others are powered by the same wires. This fact has a great influence on the behavior of the oscillators. In Table III we present the influence of the voltage on the locking of the ROs on Cyclone III IC. Considering these experimental results, we cannot determine with precision the conditions under which ROs lock. We only observed that pairs of ROs can lock or unlock if environmental conditions change. Thus, questions rise on the reliability of the PUF and as manual placement is required, the configuration of the oscillators must be carefully studied.

Voltage (V)	Number of RO locked (over 16 ROs)
0.95-1.15	0
1.20-1.25	2
1.30	4
1.35-1.40	8

TABLE III
LOCKING OF ROS DEPENDING ON VOLTAGE

IV. FURTHER ENHANCEMENTS OF THE PUF

Next, we propose some modifications of the PUF in order to enhance its characteristics.

A. Reduction of intra-chip variations

As we observed in Session III, changing environmental conditions (namely voltage and temperature) increase the intra-chip variation. This is due to the fact that identically laid-out ROs have very close oscillation frequencies. Since all ROs do not have exactly the same dependence on environmental conditions, some ROs can be more affected than others, and differences in frequencies can invert. While for laboratory temperatures the intra-chip variation did not exceeded 4 bits, for temperature ranges from 30 to 80° Celsius, up to 15 bits out of 256 were unstable. For the authentication process this is not a problem because this process is more tolerant to errors. For key generation this fact is not acceptable. Therefore, as proposed by Suh and Devadas, an error correcting code can be used to correct errors due to the intra-chip variation. As usual, the response should not be used directly as a key even after correcting the errors. On one hand, there are weak and periodic patterns in the response. On the other hand, after this process, the "key" is one codeword among the 2^{64} possible codewords. Thus the "key" only has 64 bits of entropy which is far from the expected 256 bits of entropy. A hash function (e.g. Whirlpool [3] based on a modified AES) can be used to remove weak patterns but unfortunately, the entropy of the key is still the cardinal of the codeword space.

B. Reduction of the power consumption

When dealing with the power consumption of the PUF, we used small dedicated modules made in our laboratory featuring ALTERA Cyclone III EP3C25F256C8N FPGA. The static current consumption of the module is 4 mA. We measured the consumption of the PUF using the 32 ROs and a PLL delivering the clock signal. The module consumed 24.7 mA, which is indeed considerable for a background function such as PUF. However, the PUF employs each time only two out of N ROs in order to obtain one bit of the response. Thus, we propose to stop all the $N-2$ oscillators (30 in our case) that are currently not used for the response bit. The ROs are enabled and stopped using the enable input of the structure from Fig. 1. In the improved version of the PUF we obtained a 13.4 mA current consumption. This represents the sum of the static consumption (S), the consumption of the logic which is independent of the number of ROs (i.e. PLL, counters, comparators) (L) and the consumption of the logic which

depends almost linearly on the number of ROs (multiplexers and ROs) denoted by $R(N) = \lambda \times N$ where N is the number of ROs and λ a constant float. We can make a simple calculus and show that the improved model reduces considerably the consumption of the board.

We have $S+L+\lambda\times 2 = 13.4$ mA and $S+L+\lambda\times 32 = 24.7$ mA. Then $\lambda = 0.376$ mA. In the experimental conditions that we employed (32 ROs with 8 elements each) the consumption was reduced by approximately $1 - 13.4/24.7 = 51\%$.

In [14], Suh and Devadas used 1024 ROs in their design. Then, if we shutdown unused ROs for each comparison, we should obtain a consumption of approximately $S+L+\lambda\times 2 = 13.4$ mA instead of $S + L + \lambda \times 1024 = 397.6$ mA. With our improved PUF control, we obtain a current consumption reduction of $1 - 13.4/397.6 = 96.6\%$. The PUF's power consumption thus becomes much more suitable for practical implementations.

Such an idea for reducing self-heating has been proposed in [12] but was not considered by Suh and Devadas in their design. However, in this article only one ring is selected at a time. This idea cannot be used in our proposal to save more power consumption. If we want to use only one RO, we have to count its number of raising edges during one enable time, record this number and repeat this measurement by selecting another RO during the same enable time. The main problem in such a case is the influence of the global deterministic part of the jitter on the frequency of one ring oscillator [5]. This influence will not be the same from one measurement to another. Thus the comparison between the number of raising edges of two ROs will be fair only if they are influenced by the same global deterministic part of the jitter in the same time.

V. Conclusions

The concept introduced in [14] is very simple, with a differential structure that presents an excellent behavior as long as the IC is not reconfigured. As this structure (the PUF) is useless if implemented alone in an IC, we analyze the influence of additional logic upon the response of the PUF. Our work proves that placement and routing constraints are required in order to maintain the quality of the PUF in FPGAs. Without any constraints, additional logic creates a completely different PUF and implicitly a completely different response. Instead of a small and acceptable intra-chip variation after the IC reconfiguration, we obtained the variation 48.4% that was comparable in size to an ideal inter-chip variation (50%).

The huge current consumption obtained by Suh et al was also of our concern. We improved the design in order to considerably reduce the consumption. For a small PUF, (e.g. the one described in our experimental conditions with 32 ROs) the consumption was reduced by 51%. For a greater PUF, our improvement leads to an even more important reduction: we reduced the consumption of the PUF described in [14] by 96.6%.

Moreover, we showed that there are other phenomena that influence and jeopardize the integrity of the PUF. We argued why the "locking" phenomenon is affecting the response of the PUF and we placed an exclamation mark on the fact that not all challenges can be used at any moment. Apart from the locking, our experimental results show that noisy motherboards can increase the intra-chip variation for the PUF.

VI. Acknowledgement

The work presented in this paper was realized in the frame of the SecReSoC project number ANR-09-SEGI-013, supported by the French National Research Agency (ANR). The work was partially supported also by the Rhones-Alpes Region and Saint-Etienne Metropole, France. The authors would like to thank Malick Boureima and Nathalie Bochard for their help with numerous experiments.

References

[1] R. Anderson and M. Kuhn. Tamper resistance: a cautionary note. In *Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce-Volume 2*, page 1. USENIX Association, 1996.

[2] R. Anderson and M. Kuhn. Low cost attacks on tamper resistant devices. In *Security Protocols*, pages 125–136. Springer, 1998.

[3] P. Barreto and V. Rijmen. The Whirlpool hashing function. In *First open NESSIE Workshop, Leuven, Belgium*, volume 13, page 14, 2000.

[4] S. Drimer. Volatile FPGA design security–a survey. *IEEE Computer Society Annual Volume*, pages 292–297, 2008.

[5] V. Fischer, F. Bernard, N. Bochard, and M. Varchola. Enhancing Security of Ring Oscillator-based RNG implemented in FPGA. In *Field-Programable Logic and Applications (FPL)*, pages 245–250, September 2008.

[6] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, page 160. ACM, 2002.

[7] J. Guajardo, S. Kumar, G.J. Schrijen, and P. Tuyls. FPGA intrinsic PUFs and their use for IP protection. *Cryptographic Hardware and Embedded Systems – CHES 2007*, pages 63–80, 2007.

[8] S. Kumar, J. Guajardo, R. Maes, G.J. Schrijen, and P. Tuyls. Extended abstract: The butterfly PUF protecting IP on every FPGA. In *IEEE International Workshop on Hardware-Oriented Security and Trust, 2008. HOST 2008*, pages 67–70, 2008.

[9] D. Lim, J.W. Lee, B. Gassend, G.E. Suh, M. Van Dijk, and S. Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1200–1205, 2005.

[10] R. Pappu. *Physical one-way functions*. PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2001.

[11] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026, 2002.

[12] P. Sedcole and P.Y.K. Cheung. Within-die delay variability in 90nm FPGAs and beyond. In *Field Programmable Technology, 2006. FPT 2006. IEEE International Conference on*, pages 97–104. IEEE, 2006.

[13] S.P. Skorobogatov. Semi-invasive attacks-a new approach to hardware security analysis. *Technical report, University of Cambridge, Computer Laboratory*, 2005.

[14] G.E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *44th ACM/IEEE Design Automation Conference, 2007. DAC'07*, pages 9–14, 2007.