An Easy-to-Design PUF based on a Single Oscillator: the Loop PUF

Zouha Cherif[1,2], Jean-Luc Danger[1], Sylvain Guilley[1], Lilian Bossuet[2]
[1]Institut MINES-TELECOM, TELECOM ParisTech,
CNRS LTCI, 46 rue Barrault 75 634 Paris, France.
<{cherif,danger,guilley}@telecom-paristech.fr>
[2] Université de Lyon
CNRS, UMR5516, Laboratoire Hubert Curien 42 000 Saint-Etienne, France.
<lilian.bossuet@univ-st-etienne.fr>

***Abstract*—This paper presents an easy to design Physically Unclonable Function (PUF). The proposed PUF implementation is a loop composed of N identical and controllable delay chains which are serially assembled in a loop to create a single ring oscillator. The frequency discrepancies resulting from the oscillator driven by complementary combinations of the delay chains allows to characterize one device. The presented PUF, nicknamed the Loop PUF (LPUF), returns a frequency comparison of loops made of N delay chains ($N \geq 2$). The comparisons are done sequentially on the same structure. Unlike others PUFs based on delays, there is no specific routing constraints. Hence the LPUF is particularly flexible and easy to design. The basic use of the Loop PUF is to generate intrinsic device keys for cryptographic algorithms. It can also be used to generate challenge response pairs for simple authentication. Experiments have been carried out on CYCLONE II FPGAs to assess the performance of the LPUF, such as randomness, uniqueness and steadiness. They clearly show both the easiness of design and the quality level of the LPUF. The measurement time *vs* steadiness, as well as resistance against side-channel and modeling attacks are discussed.**

Keywords: PUF, key generation, authentication, randomness, steadiness, uniqueness, FPGA, ASIC.

I. Introduction

The function of a PUF is to provide a signature specific to each integrated circuit. Their invention has been motivated to obtain both low-cost authentication methods and robust structures against physical attacks. The PUF signature is used either via a Challenge-Response Pair (CRP) protocol for authentication, or to generate a private key or random variable in a ciphering operation. It can avoid the use of digital memory to store a key imposed by the IC manufacturer or user. Hence they are well suited in low-cost devices as the RFIDs or smartcards. However the responses given to the CRP protocol could be the base of powerful attacks based on Machine Learning (ML) algorithms to create modeling attacks [1]–[4]. The side-channel attacks based on the observation of the PUF activity is another potential attack. Moreover the PUFs have to be reliable against operating conditions (temperature, voltage, *etc.*) modifications which could be either natural or malevolent in case of attacks. Consequently PUF are often associated with protocols or structures specifically designed to thwart the attacks or enhance the reliability. The structure of PUFs can take advantage of special technological process as the Optical PUF [5], [6] and the Coating PUF [6]. The Optical PUF consists of a transparent material containing randomly distributed scattering particles allowing to deviate the laser light. The Coating PUF uses an opaque material randomly doped with dielectric particles and placed on top of the IC. The Silicon PUF is certainly the simplest PUF as it does not require any technological modification. It takes advantage of randomness introduced definitively during the manufacturing process. Indeed, the dispersion between the wires and transistors is perceptible from one circuit to another, even if they are part of the same silicon wafer. The first silicon PUF introduced by Gassend *et al.* is the Arbiter PUF [7] which compares the delay between two identical controlled paths. The Arbiter PUF can be derived in XOR PUF suggested in [8] and Lightweight Secure PUF [9] which is a composition of Arbiter PUFs. To solve problems of same delay PUFs that it is easy to predict the relation between delay information and generated information, the Glitch PUF is introduced by D. Suzuki et al. in [10]

The Ring-Oscillator (RO) PUF introduced by Suh *et al.* [8] is a set of ring oscillators pairs which are compared in frequency. Guajardo *et al.* introduced the SRAM PUF [11] which is linked with the state of the SRAM at power up. The Butterfly PUF [12] works as the SRAM PUF but the memory point is based on two flip-flops. A classification given in [11] and [4] considers Strong PUFs and Weak PUFs according to their number of challenges and the difficulty to read the responses out. One special type is the Controlled PUF [13] which adds control logic to improve the PUF robustness. The Arbiter PUF is particularly fast but needs a design care to balance the delays between the two paths. The RO PUF is simple to implement as it is made of identical ring oscillator pairs but it could be sensitive to EM attacks [14].

The proposed PUF is a single ring oscillator or a "Loop" based on controllable delay elements. Contrary to the traditional approaches based on differential and parallel comparisons (Arbiter PUF, RO PUF), the Loop PUF compares multiple elements sequentially. This offers greater flexibility and design easiness. The greater measurement time (a few ms) could be largely acceptable for many applications (*e.g.* generation of cryptographic keys). Experiments have been carried

out on CYCLONE II FPGAs. Quality metric as randomness, uniqueness and steadiness have been measured and discussed. Also enhancement of steadiness and robustness against attacks has been investigated.

The article is organized as follows: Section II presents the principle of the Loop PUF and the operating modes to take advantage of its structure. The Section III describes experiments and results carried out on CYCLONE II FPGAs. Finally, conclusions and perspectives are discussed in Section V.

II. Proposed PUF

A. *Structure*

The proposed PUF, the "Loop PUF" is a silicon PUF based on N delay chains forming a loop. When closed by an inverter this loop oscillates as a single ring oscillator. a delay chain is composed of a series of M controlled delay elements connected to each other. A single controlled delay element is shown in Fig. 1. Every delay chain i receives a control word C_i of M bits. Fig. 2 represents the structure of the LPUF with $N = 3$. Each bit C_i^j, $i \in [1, N]$, $j \in [1, M]$, selects a delay value of the associated j delay element in the chain i.

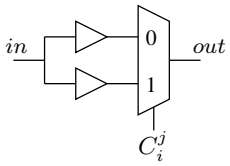


Figure 1. Basic delay element in a Loop PUF.

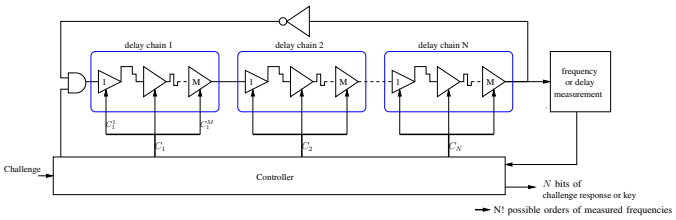


Figure 2. Loop PUF structure.

Compared to the RO PUF [8] the LPUF has only one oscillator and there is no delay chain pairing (N can be greater than 2). The noise introduced to the LPUF impacts all delay chains and the local cross-coupling is limited as there is only one oscillator. Compared to the Arbiter PUF illustrated in Fig. 3 and introduced by Gassend et al. [7], the structure of the LPUF is simpler as there is no need to cross wires in the delay elements and extra logic to balance the two chains as in [15].

The only design constraint imposed to build the LPUF is to duplicate the delay chain N times with a faithful reproduction of the placing and routing. This constraint is quite easy to meet in ASIC. In FPGA we can be doubtful as the routing structure is unknown and well protected by some FPGA manufacturers. Our experiments conducted in Sec. III show that it is easy to

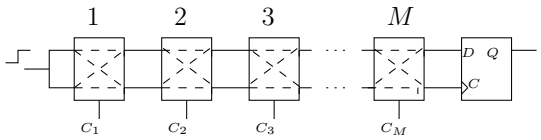


Figure 3. Arbiter PUF structure.

duplicate small structures such as delay chains in ALTERA FPGAs. Moreover Fig. 2 shows that the delay chain has no internal Place and Route constraints. The only requirement is a perfect N times duplication of the reference delay chain. The LPUF controller is in charge of extracting the result, that is either an intrinsic key or the response of a CRP challenge. To do so it has to drive the LPUF loop by a set of N control words C_i, and measure the corresponding frequency or period.

B. *LPUF control*

*1) **Principle**:* The controller measures and compares the loop oscillation frequency for different combinations of control words C_i associated to each of the N delay chains. A fixed timing window of the LPUF signal is used as a reference for the measurement. Then, the number of system clock periods are counted during this measurement window. This number is used as a result as it is directly correlated to the LPUF frequency. For a given set of control words, called "Challenge" $C_1, ..., C_N$, the controller applies different combinations of the control words and measures the frequency f or the delay d of the oscillating loop. The result should remain the same for all permutations of C_i if the delay chains are perfectly balanced. But in physical devices there is a slight frequency discrepancy because of CMOS variability is be exploited to build silicon PUFs. As an example we can consider $N = 2$ and the delay element j illustrated in Fig. 4.

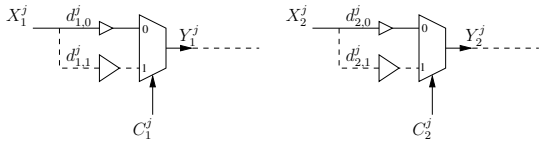


Figure 4. Delay element j for two delay chains.

If an oscillation period measurement is done with the combination $C_1^j = 0$ and $C_2^j = 1$, then with $C_1^j = 1$ and $C_2^j = 0$, the difference of the two measured delays is:

$$D^j = (d_{1,0}^j + d_{2,1}^j) - (d_{1,1}^j + d_{2,0}^j).$$

D^j should be ideally equal to zero, but this is never the case because of the process dispersion.

Hence for $N = 2$, if we consider the control words C_1 and C_2, the PUF identity ID can be expressed by:

$$\begin{aligned} ID &= sign(D_{C_1C_2} - D_{C_2C_1}). \\ &= sign\left(\sum_{j=1}^{M}(d_{1,C_1^j}^j + d_{2,C_2^j}^j) - (d_{1,C_2^j}^j + d_{2,C_1^j}^j)\right). \end{aligned}$$

If the frequency is measured instead of the time, the same equations apply by using the frequency difference rather than the delay difference.

*2) **Control strategy with** $\mathbf{N > 2}$:* The controller generates automatically the combination of control words from the initial challenge, in order to perform pairwise comparisons, as for $N = 2$. For instance, the controller rotates N times the control words in order to get N identity bits.

This is illustrated in Fig. 5 with N=3. The challenge inputs the LPUF with three control words $C_1C_2C_3$, say ABZ. Then the LPUF controller makes 3 rotations of ABZ and measures the delay for each. The LPUF returns an ID code of at least 3 bits corresponding to the code of 3! possible orders.

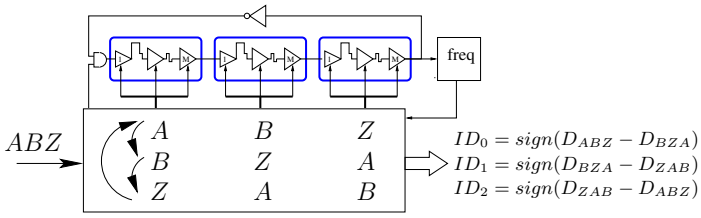


Figure 5. LPUF control example with N = 3.

In this case the 3 bits of the LPUF identity are expressed by:

$$\begin{aligned} ID_0 &= sign\,(D_{ABZ} - D_{BZA}) \\ &= sign\left(\textstyle\sum_{j=1}^{M}(d^j_{1,A^j} + d^j_{2,B^j} + d^j_{3,Z^j}) - (d^j_{1,B^j} + d^j_{2,Z^j} + d^j_{3,A^j})\right). \\ ID_1 &= sign\,(D_{BZA} - D_{ZAB}) \\ &= sign\left(\textstyle\sum_{j=1}^{M}(d^j_{1,B^j} + d^j_{2,Z^j} + d^j_{3,A^j}) - (d^j_{1,Z^j} + d^j_{2,A^j} + d^j_{3,B^j})\right). \\ ID_2 &= sign\,(D_{ZAB} - D_{ABZ}) \\ &= sign\left(\textstyle\sum_{j=1}^{M}(d^j_{1,Z^j} + d^j_{2,A^j} + d^j_{3,B^j}) - (d^j_{1,A^j} + d^j_{2,B^j} + d^j_{3,Z^j})\right). \end{aligned} \quad (1)$$

*3) **Choice of control words**:* In equation (1), we can see that the control words, $A^jB^jZ^j$, determine the number of delays which contribute to the IDs. For instance if $A^j = B^j \neq Z^j$, the two delays d^j_2 and d^j_3 are used to calculate ID_0. The reliability is enhanced if more delays are used as the variance of the resulting distribution increases proportionally to the number of delays. This property is studied in subsection III-C1 about the steadiness indicator. The difference between each control word is expressed by the Hamming distance H:

$$H = \sum_{i=1}^{N}\sum_{i'>i}^{N} HW(C_i \oplus C_{i'}), \quad (2)$$

where HW is the Hamming Weight function of C_i XOR $C_{i'}$. As H should be maximum, it can be shown that for $M = 1$ (words of one bit), the H maximum H_{max} is given by this formula:

$$\begin{aligned} N\ odd &\Rightarrow H_{max} = \tfrac{(N^2-1)}{4}. \\ N\ even &\Rightarrow H_{max} = \tfrac{N^2}{4}. \end{aligned} \quad (3)$$

In addition to the requirement of having H maximum, there is another constraint which is to avoid equivalent control words. For instance if $N = 2$ and $M = 3$, the ID obtained from the challenge $(0, 1)$ is the same as $(2, 3)$, $(4, 5)$ and $(6, 7)$. Eq. 4 with $N = 2$ and $M = 2$ shows that the challenge $(0, 1)$ is equivalent to $(2, 3)$.

$$\begin{aligned} &D_{(00,01)} - D_{(01,00)} \\ &= ((d^1_{1,0} + d^1_{2,0}) - (d^1_{1,0} + d^1_{2,0})) + ((d^2_{1,0} + d^2_{2,1}) - (d^2_{1,1} + d^2_{2,0})) \\ &= ((d^2_{1,0} + d^2_{2,1}) - (d^2_{1,1} + d^2_{2,0})) \\ &= ((d^1_{1,1} + d^1_{2,1}) - (d^1_{1,1} + d^1_{2,1})) + ((d^2_{1,0} + d^2_{2,1}) - (d^2_{1,1} + d^2_{2,0})) \\ &= D_{(10,11)} - D_{(11,10)} \end{aligned}$$

The constraint to avoid equivalent challenges can be formalized by this equation:

$$\forall j \in [1, M] \prod_{i=1}^{N} C_i^j = 0.$$

Even with this constraint the number of possible challenges is much greater with regards to the Arbiter PUF. The number of challenges for an Arbiter PUF having M elements is 2^M, whereas the LPUF has a total of 2^{NM} challenges minus the combination which does not meet the condition expressed in Eq. 4. Tab.I shows the maximum number of possible challenges for $N = 2$, $N = 3$ and for different values of M.

A minimum number of challenges have to be chosen to generate an ID with nb_{bits} number of bits. For instance to obtain an ID of **64-bit** with $N = 3$ and rotations on control words as shown in figure 5, the number of challenges is:

$$\left\lceil \frac{64}{\log_2 3!} \right\rceil = 26\ challenges.$$

III. Experiments and LPUF evaluation

Experiments have been carried out to check the easiness of design in FPGA, its complexity, reliability and quality indicators. The targeted FPGAs are ALTERA CYCLONE II running on DE2 boards.

A. *Design easiness, complexity and measurement time*

First the feasibility to duplicate the same delay chain N times in ALTERA is investigated. In this technology the Copy/Paste operation of the placed/routed blocks is not so obvious than in ASIC or XILINX FPGAs. The placement of the LPUF delay chains is constrained by using "LogicLocks" and node locations declaration. If the delay chain does not exceed the height of one row, the routing performed by the Quartus CAD tool remains the same on all the delay chains. This is corroborated by both the routing result and the delay values. The routing result is given in the file `<project.rcf>` if the command `quartus_cdb <project> --back_annotate=routing` has been set first.

Fig. 6 illustrates the placement of the LPUF with $N = 3$ chains and $M = 8$ delay elements. Every delay element uses two logic cells, one of which is a multiplexer driven by the control bit of the delay element. The right side of Fig. 6 shows

Table I
NUMBER OF CHALLENGES.

		M									
		2	3	4	5	6	7	8	10	12	16
$Ch_{arbiter}$		4	8	16	32	64	128	256	1024	4096	65536
Ch_{LPUF}	$N=2$	4	13	40	121	364	1093	3280	29524	∼250K	∼21M
	$N=3$	4	44	360	2680	19244	∼130K	∼1M	∼45M	∼2G	∼5000G

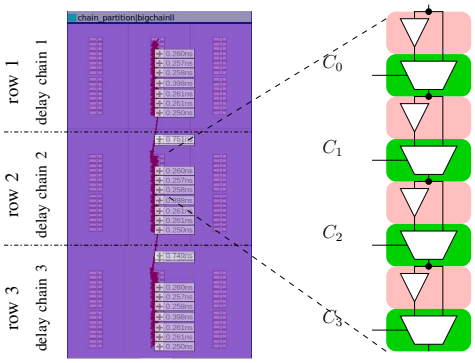


Figure 6. Placement of 3 delay chains in the loop.

4 delay elements. The 8 elements of a chain are placed in the same cluster of cells (called Logic Array Block for ALTERA).

The LPUF **complexity** in CYCLONE II with these parameters: $N = 3$, $M = 7$, $H = 9$, is of **49** cells for the loop itself and **349** for the LPUF controller.

The **measurement time** depends on the number of ID bits, and thus the number of challenges as given by equation 4. Another important factor is the reliability of the measurement. The PUF ID is more reliable if the measurement time is increased. For instance to obtain 16 reliable bits, **20 ms** are required for an error probability of 10^{-4} with a single parity bit. A more robust Error Correcting Code (ECC) could be used to reduced either the measurement time or the error probability. This point is discussed in Sec. IV-A.

B. *Inter-Chip Study*

This experience is to check the distributions of delays and the uniqueness of IDs between L PUFs placed on different devices. It has been performed on $L = 24$ FPGAs with the LPUF parameters $N = 2$, $M = 15$, each delay chain being placed in the same row and in two adjacent columns. In this experiment the frequency of oscillations is measured by counting the number of oscillations in a time window of fixes size. The ID depends on the frequency difference F^j when the challenge $\{C_1, C_2\} = \{0, 2^j\}$ is applied, where j is the delay element index. 16 delay elements makes up the delay chain but this 16th element is not considered as it cannot be balanced. This comes from the different routing between the two delay chains, as shown in Fig. 2 for the Nth delay chain. 15 challenges are provided to the LPUF, from $\{C_1, C_2\} = \{0, 2^1\}$ to $\{0, 2^{15}\}$. Hence each LPUF produces a 15-bit ID which is compared with the other LPUF IDs by using the Hamming Distance (HD). From the 276 ($^{24\times23}/_{2}$) pairwise comparisons, the Probability Mass Function is drawn for every HD, as illustrated in Fig. 7. The x-axis represents the Inter-Chip variation expressed in Hamming Distance of PUF IDs between two FPGAs, and the y-axis represents the probability of each HD. This distribution should be equivalent to a binomial distribution. We obtain the average HD of 7.51, which is very close to the ideal average of 7.5 ($= M/2$).

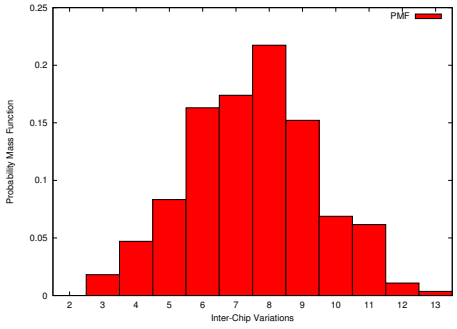


Figure 7. Inter-Chip variation on 24 LPUFs with $N = 2$, $M = 15$.

According to the process dispersion the frequency difference F^j variables follow a normal distribution.

$$F^j \sim \mathcal{N}(0, \sigma^2)\,,$$

where σ^2 is the variance due to the process dispersion. Fig. 8 represents the distribution of the F^j for the 24 boards and $M \times 24 = 360$ values of F^j. We notice a shape very close to a Gaussian distribution. The standard deviation is $\sigma = 60.8$ kHz obtained with a measurement window of 250 μs at 20°C and at the nominal power supply.

C. *Performance indicators*

The goal of this setup is to assess the quality of the LPUF according to formal indicators as:

- **Randomness**: expresses the distribution of ID is balanced (as many '1' as '0').
- **Uniqueness**: quantifies how two LPUFs in the same device (intra) or different devices (inter) return different results.
- **Steadiness**: measures the reliability against the noise and environmental conditions.

The metrics presented by Hori *et al.* [16] are based on statistical processing of the logical IDs. Other metrics presented

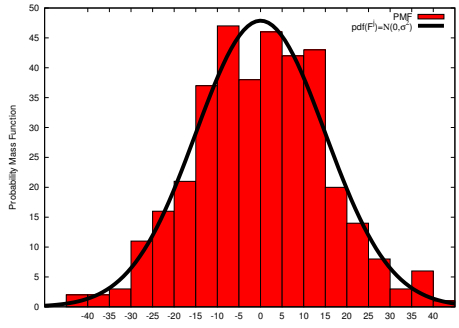


Figure 8. Distribution of F^j.

in [17] gives an estimate of the PUF with statistical evaluation of delay elements. Evaluation has been performed with this method as it provides accurate probabilities for each indicator. Moreover it is faster because it does not need to run many challenges, it relies only on delay measurements.

*1) **Metrics for the LPUF**:* In this section, we study the theoretical evaluation of the LPUF.

Randomness:

It depends on the error probability when the Gaussian distribution D of all delay elements are not perfectly centered in 0. According to [17] the Arbiter PUF randomness is given by:

$$Randomness_{APUF} = 1 - \left| erf(\frac{E(D)}{\sigma\sqrt{2 \cdot M}}) \right| ,$$

with σ being the standard deviation of the delays, and M the number of delay elements.

The LPUF randomness is theoretically maximal as the IDs are built from delay difference. Thus if we consider the complementary challenges, the ID results are complementary. For instance, for $N = 2$ with the challenge words C_1, C_2 the ID is:

$$ID_{C_1,C_2} \quad = \quad sign(D_{C_1C_2} - D_{C_2C_1})$$

which is complementary to the ID with challenge words C_2, C_1:

$$ID_{C_2,C_1} \quad = \quad sign(D_{C_2C_1} - D_{C_1C_2})$$

As these two IDs are correlated, it does not make sense to use both a challenge and its complement. If only one is used randomly, either the chosen challenge or its complement, the randomness should remain statistically perfect.

$$Randomness_{LPUF} \approx 100\ \% .$$

Uniqueness:

The uniqueness metric is to check if a correlation exists between PUF from different devices (Inter) or intra device (Intra). The experience presented in III-B gives some results about the inter-uniqueness on 24 FPGAs. The indicator here is based on a probability corresponding to a comparison of distributions.. The general delay distribution D is compared with each distribution D_j^L of delay elements j among the L different PUFs.

$$Uniqueness_{APUF} = \frac{1}{M} \sum_{j=1}^{M} (D_j^L = D).$$

For the LPUF, the number of comparisons to perform is $M \cdot N$. Then, the metric for the Uniqueness is defined by:

$$Uniqueness_{LPUF} = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} (D_{i,j}^L = D).$$

with $D_{i,j}^L$ representing the distribution of element j in the chain j for the L different instances. The details of calculation for the distribution difference $(D_j^L = D)$ is explained in [17].

Steadiness: Every delay difference of element i in chain j of the LPUF, $(d_0^{i,j} - d_1^{i,j})$, is measured T times.

The noise impact on reliability is based on T measurements of each delay element composing the chains. The noise standard deviation S is measured and considered common to all elements. The steadiness formula studied in [17] is:

$$\begin{aligned} Steadiness &= 1 - Pr(error) \\ &= 1 - Pr(error|delay < |\lambda|) \cdot Pr(delay < |\lambda|). \end{aligned} \tag{4}$$

Where λ represents a threshold delay above which there is no error.

Compared to the Arbiter PUF, the LPUF ID is obtained by computing the sum of H_{max} differences between delay elements, as explained in section II-B3. Hence the delay variance for each LPUF ID is $\sigma'^2 = H_{max} \cdot \sigma^2$. We have also to compute the variation of delay measurement S_L of a Loop PUF delay element. We consider that all delay elements have the same variation of measurement.

As the steadiness for the Arbiter PUF calculated in [17] is given by:

$$Steadiness_{APUF} \quad = \quad 1 \quad - \quad \frac{12\sqrt{2\pi} - 9}{8\pi} \times \frac{S}{\sigma}.$$

By changing the variable σ in σ' and S by S_L, we obtain the steadiness expression for the LPUF:

$$Steadiness_{LPUF} \quad = \quad 1 \quad - \quad \frac{12\sqrt{2\pi} - 9}{8\pi\sqrt{H_{max}}} \times \frac{S_L}{\sigma}.$$

*2) **Metrics Results**:* In order to measure the intra-uniqueness and steadiness, experiments have been carried out on a design with 8 LPUFs embedded in a CYCLONE II FPGA. Each LPUF has $N = 3$ chains with $M = 7$ elements. The challenges sent to the PUF are $C_1, C_2, C_3 = 0, 0, 2^j$, with j being the index of the element. The LPUF controller takes this challenge to operate the rotations and give the difference of delay for each $M \times N$ elements. The layout is shown in Fig. 9 where every chain is placed in a specific row in order to be balanced, as explained in section III-A.

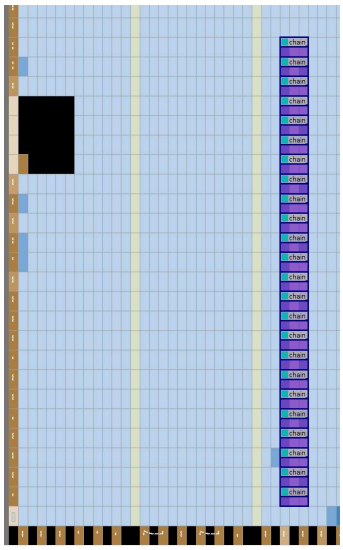


Figure 9. Layout of 8 LPUFs with $N = 3$ in CYCLONE II.

Table II
THE EXPERIMENTAL RESULTS OF THE INTRA-DEVICE EVALUATION OF THE LOOP PUF.

Performance indicator	Loop PUF
Randomness	$\approx$ 100%
Intra-Uniqueness	95%
Steadiness	98.7%

Table II gives the results for 8 Loop PUFs with M=8 and N=3. $T = 128$ tries are performed to study the steadiness.

The Loop PUF is naturally random. Although measurements have been done with challenges whose $H = 1$ (refer to Eq. 2) and not $H_{max} = 2$ for $N = 3$, we had a good uniqueness 95%. In normal condition, we obtain a good steadiness value. But to ensure that our PUF is perfect, more hostile environment with greater range of temperature and voltage is needed.

IV. DISCUSSION ABOUT SPEED, RELIABILITY AND ROBUSTNESS

A. Speed vs reliability

The experiments have been performed with an average latency of 250μs per bit per LPUF. This speed can be enough for many applications but it can also be reduced or increased according to the reliability requirements. The useful method to enhance the reliability of the PUF is performed by adding an error correction code (ECC) which is the base of the secure sketch and fuzzy extractor function as described in [18]. The steadiness gives an idea of the ECC strength as it represents the opposite of the error probability. As the LPUF uses a time measurement, the reliability can be improved by increasing the number of tries. Indeed according to the steadiness metrics (the opposite of the error probability) estimated from S and σ, the measurement window can be reduced or enlarged. Table III indicates the time to get reliable bits according to the error probability with this configuration: $N = 3$, $M = 7$, $H = 9$, number of bits = 16, CYCLONE II FPGAs. Hence for a given error probability there is a trade-off between the ECC complexity and the LPUF latency time. For instance a simple ECC like the use of a single parity bit can be enough if we accept longer LPUF responses. In this case all the 16 bits are reliable with an error probability of 10^{-4}. In this case $20ms$ is used as measurement time. In FPGAs the parity bit can merely be part of the bitstream and in ASICs like RFID tags it could be a specific pin state.

Table III
AVERAGE NUMBER OF RELIABLE BITS AMONGST 16, ACCORDING TO THE ERROR PROBABILITY AND LATENCY TIME, WITH A SINGLE PARITY CHECK

.

	LPUF latency time			
	5ms	10ms	15ms	20 ms
P(error)= 10^{-4}	12	14	15	16
P(error)= 10^{-7}	10	12	14	15

B. Discussion about Robustness against attacks

The modeling attacks are described in details by Rührmair *et al.* [4]. They are based on a model derived from Challenge Response Pairs collected by eavesdropping and represents one of the Achille's heel of Silicon PUF. One solution to thwart them could be to use cryptographic function. They can be placed either in the challenge path or the response path. Therefore it becomes difficult to build a model by a machine learning technique. For instance every word of the challenge word can be transformed by a Non-Linear (NL) function like a substitution Box (SBox). As the LPUF intrinsic key is secret, or known only by the authorized LPUF user, it can be added to the challenge word before entering the SBox, similarly to the AES datapath. However this extra logic should not be too complex to harm the low-cost interest of the PUF compared to signature in ROM or anti-fuse.

Another type of attack is the Side Channel attacks where the adversary observes the PUF activity via the current or the electromagnetic field. Hence the oscillation can be counted and the PUF ID can be deduced. This attack could be thwarted by using a random number generator (RNG) which selects the order with which the time measurement is done by the controller. As the LPUF controllers generates itself the RNG, it knows what is the real challenge. This protection could also be efficient against fault attack as the adversary does not know the real challenge. Moreover the perturbation noise affects all the delay chains in the same manner and it should not affect the result. Fig. 10illustrates a possible countermeasure for both the modeling and side-channel attack.

V. CONCLUSION

The Loop PUF based on N controlled identical delay chain has been introduced. It has been shown that this structure is easy to design and offers a huge number of challenges as it is based on sequential comparisons on N delay chains.

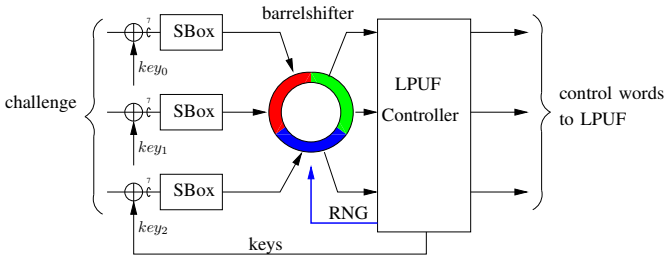


Figure 10. Countermeasure based on preprocessing of the challenge.

Consequently this permits low-cost methods for authentication and key extraction. The LPUF has been evaluated by means of experiments carried out on 24 FPGA boards and 8 LPUFs in the same FPGA. The results show that the level of speed and performance can fit in low complexity devices where a latency of a few ms is acceptable. The randomness of LPUF is perfect and the level of uniqueness and steadiness is also very high.

Future works include the study of robustness against modeling and physical attacks. Temperature and voltage tests will also be carried out to refine the reliability parameters, and check if stronger ECC are necessary.

REFERENCES

[1] Blaise Gassend. Physical Random Functions, 2003. Msc thesis, MIT.

[2] D. Lim. *Extracting Secret Keys from Integrated Circuits*. PhD thesis, MIT, 2004.

[3] M. Majzoobi, F. Koushanfar, and M. Potkonjak. Testing techniques for hardware security. In *Test Conference, 2008. ITC 2008. IEEE International*, pages 1 –10, oct. 2008.

[4] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM conference on Computer and communications security*, CCS '10, pages 237–249, New York, NY, USA, 2010. ACM.

[5] Ravikanth S. Pappu. *Physical One-Way Functions*. PhD thesis, Massachusetts Institute of Technology, March 2001.

[6] Pim Tuyls, Boris Skoric, and Tom Kevenaar. *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, December 2007. 1st Edition, ISBN 978-1-84628-983-5.

[7] Blaise Gassend, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In *ACM Conference on Computer and Communications Security*, pages 148–160, 2002.

[8] G. Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *DAC*, pages 9–14, 2007.

[9] Mehrdad Majzoobi, Farinaz Koushanfar, and Miodrag Potkonjak. Lightweight secure pufs. In *Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD '08, pages 670–673, Piscataway, NJ, USA, 2008. IEEE Press.

[10] Daisuke Suzuki and Koichi Shimizu. The Glitch PUF: A New Delay-PUF Architecture Exploiting Glitch Shapes. In *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 366–382. Springer, August 17-20 2010. Santa Barbara, CA, USA.

[11] Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, and Pim Tuyls. FPGA Intrinsic PUFs and Their Use for IP Protection. In *CHES*, Lecture Notes in Computer Science, pages 63–80. Springer, 2007.

[12] Sandeep S. Kumar, Jorge Guajardo, Roel Maes, Geert Jan Schrijen, and Pim Tuyls. The Butterfly PUF: Protecting IP on every FPGA. In Mohammad Tehranipoor and Jim Plusquellic, editors, *HOST*, pages 67–70. IEEE Computer Society, 2008.

[13] 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.

[14] Dominik Merli, Dieter Schuster, Frederic Stumpf, and Georg Sigl. Semi-invasive EM attack on FPGA RO PUFs and countermeasures. In *Proceedings of the Workshop on Embedded Systems Security*, WESS '11, pages 2:1–2:9, New York, NY, USA, 2011. ACM.

[15] M. Majzoobi, F. Koushanfar, and S. Devadas. FPGA PUF using programmable delay lines. In *Information Forensics and Security (WIFS), 2010 IEEE International Workshop on*, pages 1 –6, December 2010.

[16] Yohei Hori, Takahiro Yoshida, Toshihiro Katashita, and Akashi Satoh. Quantitative and statistical performance evaluation of arbiter physical unclonable functions on fpgas. *Reconfigurable Computing and FPGAs, International Conference on*, 0:298–303, 2010.

[17] Z. Cherif, J.-L. Danger, and L. Bossuet. Performance evaluation of physically unclonable function by delay statistics. In *NEWCAS*, Bordeaux, june 2011.

[18] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.*, 38(1):97–139, 2008.