# FROPUF: How to Extract More Entropy from Two Ring Oscillators in FPGA-Based PUFs

Qinglong Zhang, Zongbin Liu, Cunqing Ma, Changting Li, and Jiwu Jing

Institute of Information Engineering, Chinese Academy of Sciences, China.

**Abstract.** Ring oscillator (RO) based physically unclonable function (PUF) on FPGAs is crucial and popular for its nice properties and easy implementation. The compensated measurement based on the ratio of two ring oscillators' frequencies proves to be particularly effective to extract entropy of process variations. However from two ring oscillators only one bit entropy is extracted and RO PUFs will occupy numerous resource with the size of private information increasing. Motivated by this inefficient resource usage, we propose an elegant and efficient method to extract at least 31 bits entropy from two ring oscillators on FPGAs by utilizing the fine control of programmable delay lines (PDL). We call this construction Further ROPUF (FROPUF). In this paper, we present in detail how to take advantage of the underlying random process variation which derives from the lookup tables (LUT) of two ring oscillators, and show that the in-depth variation can be extracted by a similar second order difference calculation. In addition, we reveal the consistency of the evaluation results from Xilinx FPGAs (e.g. Virtex-5, Virtex-6, Kintex-7) and those by simulation of FROPUF. The responses of our new construction have a nominal bit-error-rate (BER) of 1.85% at 27 °C and FROPUF greatly promotes the number of entropy with equivalent reliability of the general ROPUF.

**Key words:** PUFs, Ring Oscillator, Entropy, FPGA

## 1 Introduction

With flourishing development of embedded devices in modern age, people are favor of FPGA to implement cryptographic algorithms on hardware because of FPGA's reconfigurable nature. An indispensable premise for the security of cryptographic primitives is the ability to securely generate, store and retrieve private keys. In tradition, it is ascribed to a protected memory which can reliably store the private information while shielding it completely from unauthorized parties, but this requirement is non-trivial to achieve in practice [1]. Recently, physically unclonable function is attracting wider attention as a technique to provide physical roots of trust in embedded systems [2–4]. Due to the submicron process variation during manufacturing, every identical logic circuit has slightly different physical properties. The concept of PUF is to utilize these intrinsic process variations to extract a unique electronic fingerprint to solve issues such

as cryptographic key generation [5], intellectual property (IP) protection [6, 7], device authentication [8–10] and trusted computing.

A variety of PUFs have been proposed, such as SRAM PUF [11], Buffterfly PUF [12], Glitch PUF [13], Flip-Flop PUF [14], Ring Oscillator PUF [15] and so on. However, some kinds of these PUFs are not so easy to be implemented on commercial FPGAs. In the current state of the art Xilinx and Altera FPGAs, the start-up values of SRAM are reset to a known value by the chip manufacturers, which leads to SRAM PUF's unavailability on FPGAs. Moreover, many PUF designs like Butterfly PUF and Arbiter PUF require a careful routing symmetry that is difficult to implement on FPGAs. Even, the fundamental element for Butterfly PUF, a latch with a preset signal and a clear signal, is not provided on Xilinx's newest 6-series and 7-series FPGAs. RO PUF which is first proposed by Suh and Devadas [15] has been widely used due to its sensitivity to process variation, and particularly the hard-macro design technique simplifies the implementation of several identical ROs on FPGAs. However, besides these advantages, Maiti [16] pointed out that some factors like the systematic or correlated process variation and the regional environmental noise may degrade the uniqueness and the reliability of RO PUF responses.

Up to now, there are many researches [15–21] aiming to improve and strengthen the properties of RO PUF. In DAC 2007, Suh and Devadas [15] applied a post-processing technique called 1-*out-of-k* masking to greatly enhance the reliability of the responses, but it comes at a relatively large resource overhead. In J.Cryptol.2011, Maiti et al. [16] proposed a configurable ring oscillator technique to produce nearly 100% error-free PUF outputs over varying environmental condition without post-processing. This technique is quite effective to resolve PUF reliability issues on FPGAs. However, two configurable ROs only output one bit response in order to make a trade off between reliability and the number of responses. Meanwhile, to select the most stable pair consumes several comparison calculations.

Generally it is more difficult to carry out attacks on PUFs which have more entropy. Unfortunately the fuzzy extractor for PUFs will cause entropy loss [22, 23]. As a result, the amount of entropy in PUFs is also another necessary evaluation index. Habib et al. [24] proposed an FPGA PUF based on programmable LUT delays to acquire more than one bit entropy from two ROs and presented that with LUT's input varying from '000' to '111', the eight frequencies' order is different from other ROs'. While in CHES 2011 [25] and [26], it is stated the delay values with input '111' are on average about 10 pico-seconds larger than the delay values with input '000'. Habib et al. gave one reason that the device used in [24] is Spartan-3E, while in [25] and [26], Virtex-5 devices are used. However, if the frequencies in varied LUT's input have a rough order, the method used in [24] would not be valid.

In this paper, by utilizing the fine adjustment of LUT on FPGAs [25], we propose a comprehensive model to extract more available process variations for the generation of PUF's responses by a similar second order difference calculation and we can achieve at least 31-bit entropy from only two ring oscillators.

What's more, through this second order difference calculation, we can efficiently reduce the effect of the systematic process variation and the regional environmental noise. In order to verify its validity, we achieve evaluations of our proposed model with both simulation and practical experiments. The evaluation results demonstrate that our proposed PUF possesses excellent reliability and uniqueness under varied temperatures.

Although RO PUF can be attacked by modeling attacks, modeling attacks can be successfully disabled if one uses a secure one-way hash over outputs of the PUF to create a Controlled PUF [27]. The main topic of this paper is how to extract more entropy from two ROs, and to use a secure access to the outputs of PUFs is not the topic of this paper.

In summary, we make the following contributions.

- We propose an elegant method to extract fine process variations by second order difference calculation which can efficiently reduce systematic process variation and regional environmental noise to guarantee our proposed PUF's reliability.
- We design a new construction named Further RO PUF, which can generate responses with at least 31-bit entropy from only two ring oscillators on FPGAs.
- We conduct both simulation and practical experiments to demonstrate that our new proposed PUF has a bit-error-rate of 1.85% at 27 °C and an average inter-distance of 49.32%.

The rest of the paper is organized as follows. Section 2 presents preliminaries for our paper. Section 3 describes our model for RO PUFs with fine control of LUT's inputs and proposes our new construction with second order difference calculation. Section 4 evaluates the performance of our PUF from simulations and practical experiments. Section 5 gives some further discussions on our PUF. Finally, we conclude this paper in section 6.

## 2 Preliminaries

A typical example of ring oscillator based PUF is shown in Figure 1. An RO PUF circuit consists of n identically laid-out ROs, $RO_1$ to $RO_n$, with frequencies, $f_1$ to $f_n$, respectively. In general, the challenge of this RO PUF is (i, j) as the select bits of the multiplexers to select a pair of ROs, $RO_i$ and $RO_j$ $(i \neq j)$. Due to intrinsic process variation, $f_i$ and $f_j$ will differ from each other. Based on the compensated measurement proposed by Gassend et al. [2], a response bit $r_{ij}$ can be produced from two ROs by the comparison expression as follows:

$$r_{ij} = \begin{cases} 1 & if \quad f_i > f_j, \\ 0 & otherwise. \end{cases} \tag{1}$$

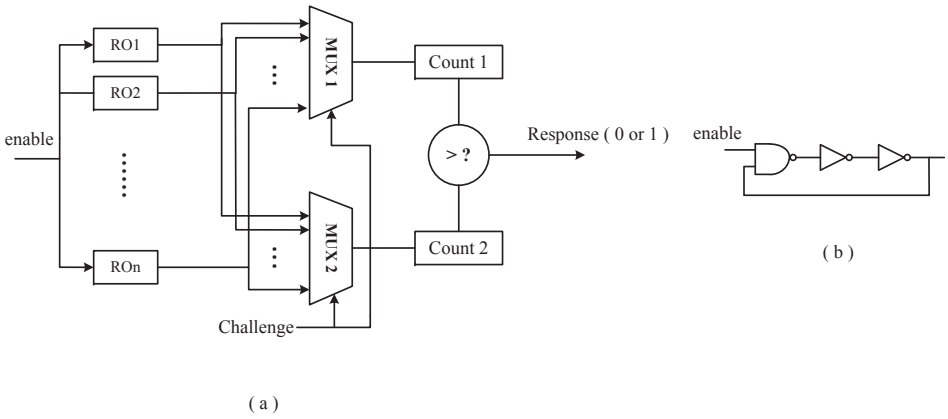


Fig. 1: (**a**)A typical example of RO PUF. (**b**) A three-stage ring oscillator

### 2.1 Evaluation Scheme of RO PUF

The evaluation scheme of a PUF is usually divided into three basic aspects, reliability, uniqueness and security [16].

- Uniqueness estimates how uniquely a PUF can distinguish different entities based on the generated responses.
- Reliability evaluates how stable the responses of a PUF are when the environmental variable (such as temperature, supply voltage) varies.
- Security is the ability of a PUF to prevent an adversary from stealing the PUF secrets.

Uniqueness can be measured through inter-distance. As defined in [28], for a particular challenge, the inter-distance between two different instantiations is the hamming distance (HD) between the two responses resulting from applying this challenge once to both PUFs. We estimate the uniqueness of a PUF by the average inter-distance over a group of chips. With two PUF instantiations, $i$ and $j$ ($i \neq j$), both having a n-bit response, $R_i$ and $R_j$ respectively, the average inter-distance $\mu_{inter}$ among $k$ chips is calculated as

$$\mu_{inter} = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100\% \tag{2}$$

Reliability can be evaluated through intra-distance. For a particular challenge, the intra-distance between two evaluations on one single PUF instantiation is the hamming distance between the two responses resulting from applying this challenge twice to one PUF. Although we expect a PUF response to be static, there are environmental factors like temperature variation, supply voltage fluctuation and so on, which may affect the reproducibility of a PUF response. To evaluate the reliability of a PUF, we achieve n-bit response $m+1$ times from the

chip $i$ at some environmental condition and select the first n-bit response as the reference response $R_i$ and the other responses as $R_{i,j}$ $(1 \leq j \leq m)$. The average intra-distance $\mu_{intra}$ calculated as follows can be used to evaluate the reliability of the PUF in this environmental condition.

$$\mu_{intra} = \frac{1}{m} \sum_{j=1}^{m} \frac{HD(R_i, R_{i,j})}{n} \times 100\% \tag{3}$$

### 2.2 Systematic Variation

In J.Cryptol.2011, Maiti et al. [16] pointed out that the total delay in a ring oscillator loop can be modeled as follows:

$$d_{LOOP} = d_{AVG} + d_{RAND} + d_{SYST} \tag{4}$$

where $d_{AVG}$ = the nominal delay that is the same for all the identical ROs; $d_{RAND}$ = delay variation due to process variation; $d_{SYST}$ = delay variation due to the systematic variation. Then the difference between two ring oscillators, $a$ and $b$, can be calculated as follows:

$$\begin{aligned} \Delta d_{LOOP} &= (d_{AVG} + d_{RAND_a} + d_{SYST_a}) - (d_{AVG} + d_{RAND_b} + d_{SYST_b}) \\ &= \Delta d_{RAND} + \Delta d_{SYST} \end{aligned} \tag{5}$$

From formula (5), a single response bit $r_{ab}$ between these two ring oscillators is not only decided by the random process variation, but also by the systematic variation. Maiti et al. showed that the systematic process variation can lead to a gradual change in the delay as a function of the physical location of ROs, and the existing of systematic variation results in the loss of uniqueness. In [16], a method is given that two closely located ROs will have similar $d_{SYST}$ in (4), and the $\Delta d_{SYST}$ is a very low value in (5).

### 2.3 Programmable Delay Lines

On FPGAs, LUT is the main programmable delay logic unit and the construction of a 3-input LUT is shown in Figure 2. The LUT is composed of a set of SRAM cells and a tree-like structure of multiplexers (MUXs). The former stores the intended functionality and the latter enables selection of each individual SRAM cell content. One LUT can be implemented as an inverter, whose output $(O)$ is always an inversion of its first input $(A_1)$, and the inputs ($A_2$ and $A_3$) are configured to have no effects on the relationship between $A_1$ and $O$. In CHES 2011, Majzoobi et al. [25] propose a novel technique to vary the signal propagation path length in minute increments/decrements by only using a single LUT on reconfigurable FPGA platform. The mechanism changes the propagation path inside the LUT by altering the inputs for the LUT. Although the inputs $A_2$ and $A_3$ have no influence on the logic function of this inverter, their values affect

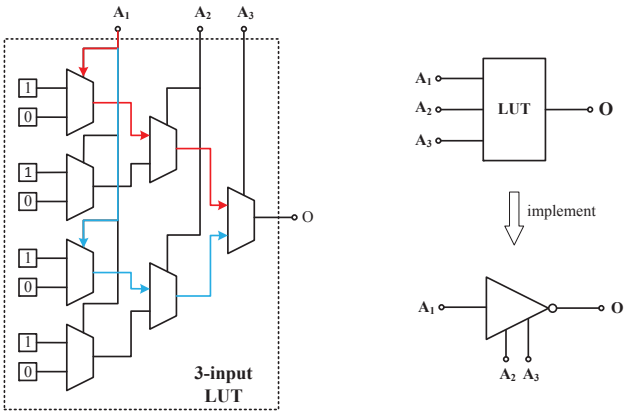


Fig. 2: Programmable delay lines using an LUT

the signal propagation path from input $A_1$ to output $O$. Majzoobi et al. point that as shown in Figure 2, for $A_2A_3 = 00$ and $A_2A_3 = 11$, the two propagation paths from $A_1$ to $O$ are the shortest and the longest respectively. The new Xilinx series products, Virtex-5,6,7 and Spartan 6, utilize 6-input LUTs. Therefore, as the method proposed by Majzoobi, a programmable delay inverter can be implemented with at most $2^5 = 32$ discrete levels for controlling the propagation delay. For example, it is an example of this fine control for 5-stage ROs and the LUTs are 6-input in Figure 3. Five of these inputs are configured as delay control.

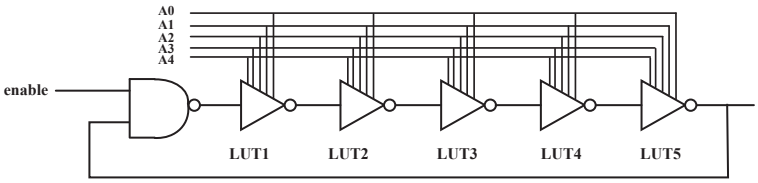


Fig. 3: An example of this fine control for 5-stage ROs

Based on the multiple control of the LUT's propagation delay, Habib et al. [24] try to extract more entropy from two ring oscillators. According to their experiment results that the frequency varies significantly depending on the LUT's input sequence and the frequency's changing pattern is different from another ring oscillator, Habib et al. propose a method to extract more entropy by comparing between two ring oscillators with any configuration from '000' to '111' for LUTs. However, in the experiment results of Majzoobi et al. [25], the propagation delays of input '11111' are on average about 10 pico-seconds larger than the corresponding values of input '00000'. Habib et al. analyze that the

reason might be that they use Spartan 3E devices which are based on 90nm technology, while Majzoobi et al. use Virtex-5 devices which are 65nm technology. Therefore, if the frequency varies in a rough order depending on the LUT's input sequence, the result of the proposed method in [24] will be not efficient on Virtex-5 devices.

## 3 Our Proposed Further ROPUF

Based on the model of the total delay in a ring oscillator proposed by Majzoobi et al. [25], we present a model which is involved with the fine process variation of different LUT's inputs. Ring oscillator $l$ consisting of 6-input LUTs can be modeled as follows:

$$d_{LOOP(l,j)} = d_{AVG} + d_{RAND(l,j)} + d_{SYST(l,j)} \quad (1 \leq j \leq 32) \tag{6}$$

where $d_{AVG}$ is the nominal delay which is the same for all the identical ROs; $d_{RAND(l,j)}$ is the delay variation due to the random process variation when LUTs are driven by the $j^{th}$ input; $d_{SYST(l,j)}$ is the delay variation due to the systematic variation. The variables $d_{RAND(l,j)}$ and $d_{SYST(l,j)}$ could be positive and negative. For a ring oscillator with different LUT inputs, $in_{j_1}$ and $in_{j_2}$, these two $d_{SYST(l,j_1)}$ and $d_{SYST(l,j_2)}$ are extremely close as shown in [16]. Therefore, in formula (6), the subscript of $d_{SYST(l,j_1)}$ and $d_{SYST(l,j_2)}$ can be modified to $d_{SYST(l)}$, where $l$ is just related to the location of ring oscillators. And formula (6) can be changed to formula (7) as follows.

$$d_{LOOP(l,j)} = d_{AVG} + d_{RAND(l,j)} + d_{SYST(l)} \quad (1 \leq j \leq 32) \tag{7}$$

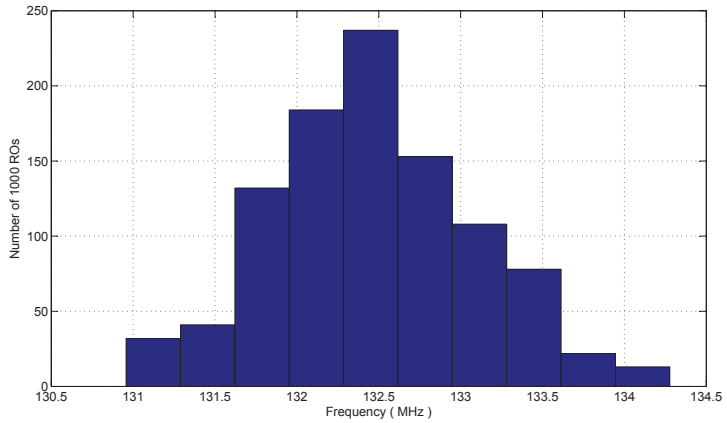


Fig. 4: The histogram distribution of 1000 ROs' frequencies

Moreover, for a group of $L$ ring oscillators with the same LUT input $in_j$, there are $L$ values $d_{RAND(1,j)}$, $d_{RAND(2,j)}$, $\cdots$ ,$d_{RAND(L,j)}$. In ReConFig 2008 [29] and

HOST 2011 [30], authors show that the distribution of these values approaches Gaussian. Figure 4 shows the distribution of our experimental results, and it also seems a normal distribution and in section 4, we will describe our experiments in detail. Therefore, we assume that these $L$ values are distributed normally. Apply this assumption to other LUT input configurations and we can achieve 32 normal distributions as follows:

$$(d_{RAND(1,j)}, d_{RAND(2,j)}, \cdots, d_{RAND(L,j)}) \sim N(\mu_j, \sigma_j^2) \quad (1 \leq j \leq 32) \tag{8}$$

That is to say, the random variable $d_{RAND(j)}$ is a normal distribution with mean $\mu_j$ and standard deviation $\sigma_j$.

### 3.1 Second Order Difference Calculation

According to the above description, for a group of $L$ ring oscillators, by varying the LUT's input from '00000' to '11111', we can get $32 * L$ different $d_{LOOP(l,j)}$ which has the similar form in formula (9). We propose an elegant method to generate responses based on second order difference calculation.

$$d_{LOOP(l,j)} = d_{AVG} + d_{RAND(l,j)} + d_{SYST(l)} \quad (1 \leq j \leq 32, 1 \leq l \leq L) \tag{9}$$

Our proposed method can be divided into two steps and here we present a neat example to illustrate our method.

1. For a ring oscillator $l$, select $d_{LOOP(l,j)}$ and $d_{LOOP(l,j+1)}$, then calculate the difference value $\Delta d_{LOOP(l,j)}$, $1 \leq j \leq 31$.
2. For two ring oscillators $l_1$ and $l_2$, generate one bit $r_{l_1,l_2,j}$ $(1 \leq l_1 \neq l_2 \leq L, 1 \leq j \leq 31)$ as follows.

$$r_{l_1,l_2,j} = \begin{cases} 1 & if \quad \Delta d_{LOOP(l_1,j)} > \Delta d_{LOOP(l_2,j)}, \\ 0 & otherwise. \end{cases} \tag{10}$$

Through these two steps, we will get a 31-bit response from two ring oscillators. Among these $L$ ring oscillators, 31*(L-1) bits can be extracted. Based on formula (9), $\Delta d_{LOOP(l,j)}$ is calculated from as follows:

$$\Delta d_{LOOP(l,j)} = d_{LOOP(l,j)} - d_{LOOP(l,j+1)}$$

$$= d_{AVG} + d_{RAND(l,j)} + d_{SYST(l)} - (d_{AVG} + d_{RAND(l,j+1)} + d_{SYST(l)})$$

$$= d_{RAND(l,j)} - d_{RAND(l,j+1)} \tag{11}$$

From formula (11), the systematic variation can be neatly removed by this first order difference calculation. According to the assumption condition (8), both

$d_{RAND(j)}$ and $d_{RAND(j+1)}$ are normally distributed, and assume the correlation coefficient between these two random variables is $r_j$, we can get the distribution of the random variable $\Delta d_{LOOP(j)}$ as follows:

$$\Delta d_{LOOP(j)} \sim N(\mu_j - \mu_{j+1}, \sigma_j^2 + \sigma_{j+1}^2 - 2 * r_j * \sigma_j^2 * \sigma_{j+1}^2) \quad (1 \leq j \leq 31) \quad (12)$$

Suppose that $\mu_{LOOP(j)}$ is $\mu_j - \mu_{j+1}$ and $\sigma_{LOOP(j)}$ is $\sigma_j^2 + \sigma_{j+1}^2 - 2 * r_j * \sigma_j^2 * \sigma_{j+1}^2$. On the base of the random variable $\Delta d_{LOOP(j)}$'s distribution, through the second step of second order difference calculation, we can calculate the distribution of the random variable $R_{l_1,l_2,j}$ as follows.

$$R_{l_1,l_2,j} \sim N(0, 2 * \sigma_{LOOP(j)}^2) \quad (1 \leq j \leq 31) \quad (13)$$

Based on formula (13), $r_{l_1,l_2,j}$ will be equally likely between '0' and '1' (probability = 0.5) as the result of our second order difference calculation method. Therefore, the 31-bit response extracted from two ring oscillators has 31-bit entropy. Theoretically, based on (13), every bit has probability 50 % to be '0' and 50 % to be '1' and if these 31 bits have no correlation, it can be stated that 31-bit entropy is extracted from these two ring oscillators.

In order to evaluate the randomness and entropy of responses, we will carry out NIST test suits on the responses generated by FROPUF in section 4.4.

### 3.2 Analysis of the Second Order Difference Calculation

The key point to extract more entropy from two ROs is to extract more process variations which may be smaller, but the magnitude of these process variations is close to that of noise. Therefore, the method to extract more entropy should reduce the effect of noise to the greatest extent.

In the traditional architecture of RO PUF, an LUT is used as an inverter, namely, which is just implemented with one signal propagation paths. In programmable delay line model, the fine control of LUT's inputs leads to different signal propagation paths. Habib et al. [24] have tried to extract more responses by utilizing the change of LUT's inputs. However, these different signal propagation paths will result in a rough order in Xilinx Virtex-5,6,7 series devices. On these devices, although the simple comparison between the corresponding inputs of two ring oscillators can lead to a 32-bit response, the correlation between these bits may cause entropy loss, even gives rise to only one bit entropy.

In our second order difference calculation method, through the first order difference calculation between different inputs of the same ring oscillator, the result can reduce the negative effect of systematic variation because inside the same ring oscillator the systematic variation is fairly close. The result computed from the first order difference calculation can be regarded as a combination of the characteristics from the compared two signal propagation delays. Then through the second order difference calculation, this result is affected by the combination of the process variations from the corresponding two signal propagation delays between two ring oscillators. As is shown in the two steps described in section

3.1, the response is decided by the sign of formula (14). $l$ denotes the serial number of ring oscillators and $j$ denotes the $j^{th}$ configuration inputs for ROs.

$$(d_{LOOP(l,j)} - d_{LOOP(l,j+1)}) - (d_{LOOP(l+1,j)} - d_{LOOP(l+1,j+1)}) \quad (14)$$

Formula (14) can be written as another form as follows.

$$(d_{LOOP(l,j)} - d_{LOOP(l+1,j)}) - (d_{LOOP(l,j+1)} - d_{LOOP(l+1,j+1)}) \quad (15)$$

In a word, from formula (15), the second order difference calculation is used to extract the variation of process variation.

Furthermore, another advantage of second order difference calculation is that the method can strengthen the reliability of our proposed PUF. The primary idea to counter the influence of environmental conditions is proposed by Gassend et al. [2]. They utilize the comparison of two ROs' frequencies to generate one bit response to reduce the environmental changes' negative effect. The second order difference calculation is involved with two difference functions which reduce not only the influence of the systematic variation but also the influence of the environmental fluctuations.

### 3.3 Simulation of Second Order Difference Calculation

In order to reflect the individual difference, these factors, process variation and environmental change, should be considered during simulation. Process variation is generally classified into systematic variation and random variation. Systematic variation is mainly affected by the location in a wafer or a chip. For example, in the architecture of RO PUFs systematic variation leads to the result that the frequencies of the ROs in one region are average larger than those in another region [16]. On the contrary, random variation has no relationship with components' spatial location.

The information of the parameters used in simulation can be extracted by observing the frequencies on FPGA platform. The parameters are as follows.

- Systematic delay $d_{SYST}$ affected by spatial location: $\sim$ N(0,$\sigma^2_{syst}$).
- Component delay $d_{RAND}$ affected by random variation: $\sim$ N($\mu_j$,$\sigma^2_j$), where $j$ represents the $j^{th}$ input for LUTs.

From the parameters defined above, the delay value for different LUT inputs of different ring oscillators can be simulated and the second order difference calculation can be carried out by Algorithm 1,where sampling y from a distribution N($\mu$,$\sigma^2$) is denoted as $y \longleftarrow$ N($\mu$,$\sigma^2$).

**Algorithm 1:** Simulation Algorithm of Second Order Difference Calculation

**Settings:** $\cdot MAX_{NumRO}$ is the number of ring oscillators.
$\cdot MAX_{NumIn}$ is the number of different LUT's inputs.

**Output:** $r_{l,j}$, $0 \le l \le MAX_{NumRO} - 1$, $0 \le j \le MAX_{NumIn} - 1$

**for** l = 1 to $MAX_{NumRO}$ **do**
  $d_{SYST(l)} \leftarrow \mathrm{N}(0,\sigma^2_{syst})$
  **for** j = 1 to $MAX_{NumIn}$ **do**
    $d_{RAND(l,j)} \leftarrow \mathrm{N}(\mu_l,\sigma^2_l)$
  **end for**
**end for**
**for** l = 1 to $MAX_{NumRO} - 1$ **do**
  **for** j = 1 to $MAX_{NumIn} - 1$ **do**
    $\Delta_{LOOP(l,j)} \leftarrow (d_{AVG} + d_{RAND(l,j)} + d_{SYST(l)}) - (d_{AVG} + d_{RAND(l,j+1)} + d_{SYST(l)})$
    $\Delta_{LOOP(l+1,j)} \leftarrow (d_{AVG} + d_{RAND(l+1,j)} + d_{SYST(l+1)}) - (d_{AVG} + d_{RAND(l+1,j+1)} + d_{SYST(l+1)})$
    **if** $\Delta_{LOOP(l,j)} \ge \Delta_{LOOP(l+1,j)}$ **then**
      $r_{l,j} \leftarrow 1$
    **else**
      $r_{l,j} \leftarrow 0$
    **end if**
  **end for**
**end for**

## 4 Evaluation

In this section, we present the evaluations of our FROPUF from 15 Virtex-5 XC5VLX110T-1FF1136 FPGAs, 10 Virtex-6 XC6VLX240T-1FF1156 FPGAs and 5 Kintex-7 XC7K325T-2FFG900 FPGAs. According to practical measurements, we achieve the parameters for simulation. From both the simulation and practical experiments, we reveal the consistency of our simulation model and the practical architecture.

Figure 5 shows our experimental evaluation system which uses Virtex-5 FPGA. A 50-MHz clock signal generated by an on-board oscillator is applied to the reference counter which provides a fixed time interval to record the counters of the compared ring oscillators. In Figure 5, we place 200 ring oscillators and each of them is composed of 16 LUTs. 15 LUTs serve as inverters with 5 configuration inputs and 1 LUT serves as an enable switch. All 16 LUTs are put into four slices, and that is to say, a ring oscillator is composed of 4 slices. Hard Macro technique is used to construct our ring oscillator to make sure that these 200 ring oscillators identical to a large extent. The whole experimental evaluation system is mainly controlled by the control module. The control module is responsible for the changing of inverters' five configuration inputs and the selection for the multiplexer, and provides an enable control signal based on the

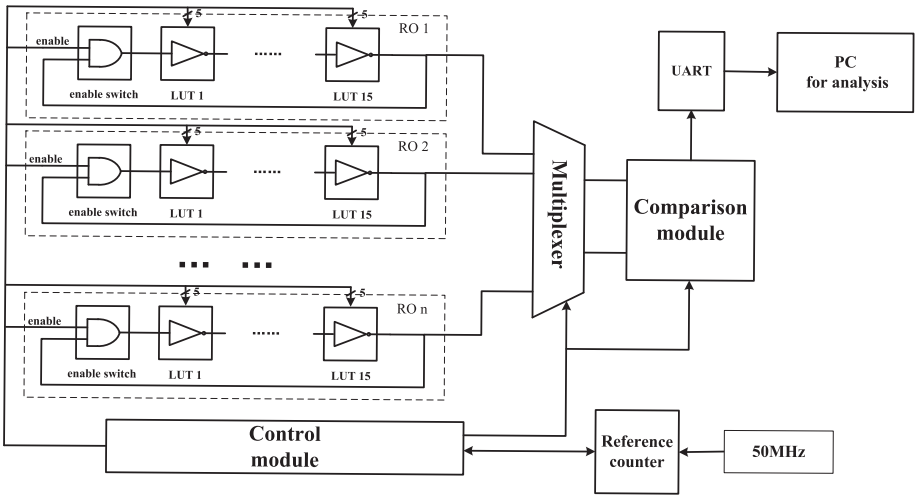


Fig. 5: The experimental evaluation system

reference counter. In order to evaluate the responses generated by our FROPUF, we utilize UART interface to transmit these responses to PC for analysis.

In our evaluation system, we select 15-stage RO whose frequency is about 132 MHz because the frequency of 15-stage RO is medium and if the frequency is too high, it may result in more instability. To demonstrate the validity of our design, the configuration inputs for all LUTs are the same, and that is to say, the configuration input space is $2^5$. In our evaluation system, there are 200 ROs and these ROs occupies 800 slices. The other control module and the UART module have 213 and 126 slices separately.

In normal environmental condition, we perform a basic experiment on delay characteristics described in section 3 in order to extract the parameters needed for the simulation of FROPUF and the second order difference calculation. The parameters are shown in Table 1. Table 1 shows the standard deviation of LUT's $j^{th}$ configuration input random delay. Through these parameters and our simulation model, we can acquire the *intra-distance* and *inter-distance* to evaluate the reliability and uniqueness of FROPUF .

Table 1: Parameters for simulation in normal environmental condition

| Parameter | Value |
|---|---|
| Standard deviation of systematic delay $\sigma_{d_{SYST}}$ $(\%^2)$ | 3.5336 |
| Standard deviation of component random variation $\sigma_{d_{RAND}}$ $(\%^2)$ | 4.7636 |

### 4.1 Reliability

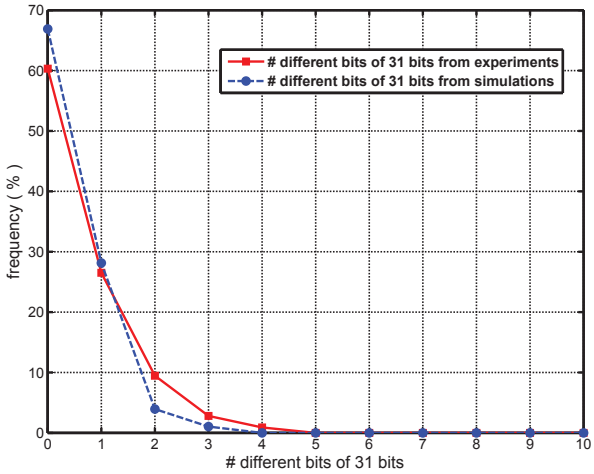


Fig. 6: The intra-distance evaluation from practical experiments and simulations

Reliability is mainly evaluated by intra-distance and it is extremely significant for a PUF to show how stably it can reproduce the same response for the same challenge. Figure 6 plots the evaluation results from simulation and experimental results of our evaluation systems. The simulation parameters are from the Table 1. The average error rate is around 1.25%. Then Figure 6 shows the evaluation results of practical experiments. The steps to calculate the average intra-distance of practical experiments are as follows.

1. Let every two ROs output 31-bit response 200 times and record each response $\mathrm{RES}_{l,k,t}$. $1 \leq l \leq 15$ denotes the serial number of FPGA boards, $1 \leq k \leq 100$ denotes the number of RO pairs and $1 \leq t \leq 200$ denotes the times of the record for the same RO pair.
2. For one RO pair, among the 200 responses, choose any two responses to record the number of different bits of 31 bits.
3. Carry out the same calculation for all the RO pairs of 15 FPGAs.
4. Achieve the percentage of the number of different bits in 31 bits and the average intra-distance.

The measurement are at normal temperature (27 °C) for 15 Virtex-5 XC5VLX110T-1FF1136 FPGAs. The average error rate is also around 1.85%. The error rate is close to the other RO PUF designs [5, 18]. In addition from Figure 6, it indicates that the behavior of the error rate can be assessed by simulation with high accuracy.

The change of temperature is a negative factor for ring oscillator based PUF and experiments are conducted at different temperatures. Figure 7 shows that

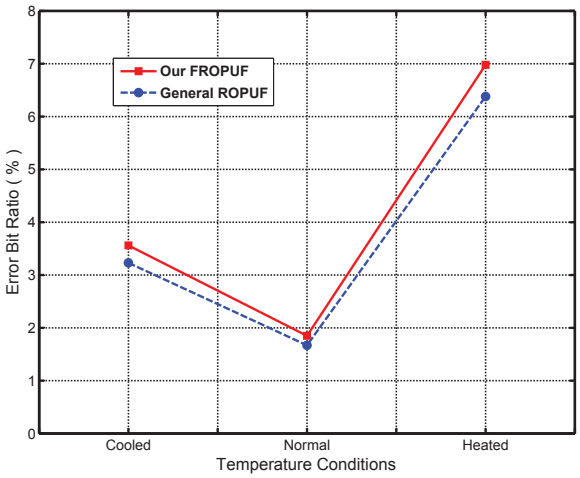


Fig. 7: The error bit ratio under different temperature conditions

with the temperature arising to (about 70 °C), the intra-distance changes to be about 6.98%, which is about the half of 15% assumed in [31]. From Figure 7, the intra-distance of FROPUF is a little higher than general ROPUF. Because FROPUF extracts the variation of process variation and its magnitude is close to that of noise, and the second order difference calculation can greatly reduce the effects of noise and systematic variations, FROPUF achieves almost the same error bit rate with the general ROPUF.

### 4.2 Uniqueness

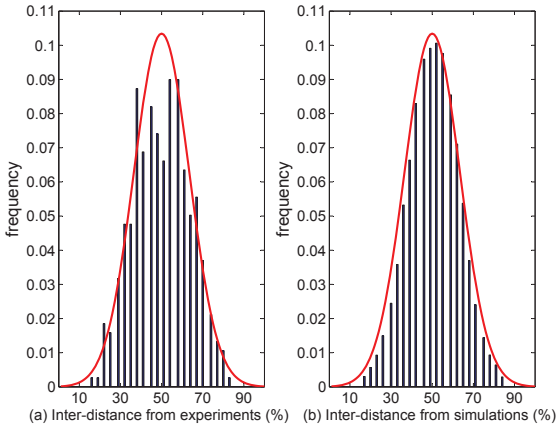


Fig. 8: The inter-distance evaluation from practical experiments and simulations

Uniqueness is mainly evaluated by inter-distance and Figure 8 (a) is a histogram of hamming distances between different PUF instantiations' responses of different FPGAs in practice. Every instantiation is composed of two ring oscillators and outputs a 31-bit response. We deploy 100 instantiations on each of 15 Virtex-5 FPGAs. This part of experiments can be used to view how different these PUF instantiations are. The result shows that the average of inter-distance is about 49.32%. Figure 8 (b) shows the result of the same evaluation by simulation. Through Figure 8 (a) and Figure 8 (b), we conclude that the simulation is able to evaluate the randomness of responses generated by PUF instantiations.

### 4.3 The Randomness Evaluation

NIST test suites are carried out to evaluate the randomness of the responses generated by FROPUF. The length of the response generated by each instantiation is 31 bits and we get totally 46500 bits responses. Because the limit of the response's length, we use 9 basic NIST tests and the result is shown in Table 2. The *Frequency* test indicates that the responses has nearly 50% to be '1' and 50% to be '0' and this practical result is similar to the theoretical analysis in section 3.1.

Table 2: The result of NIST for FROPUF's responses

| Statistical test | P-VALUE | PROPORTION |
|---|---|---|
| Frequency | 0.350485 | 10/10 |
| BlockFrequency | 0.911413 | 10/10 |
| CumulativeSums(forward) | 0.739918 | 10/10 |
| CumulativeSums(backward) | 0.035174 | 10/10 |
| Runs | 0.534146 | 10/10 |
| LongestRuns | 0.628713 | 10/10 |
| Rank | 0.122325 | 10/10 |
| FFT | 0.523478 | 10/10 |
| Serial($\nabla^1$) | 0.712378 | 10/10 |
| Serial($\nabla^2$) | 0.328793 | 10/10 |
| LinearComplexity | 0.189283 | 10/10 |

Table 3: Comparison of the entropy extracted from two ROs

| | Our Work | General RO PUF | Habib et al. [24] | Maiti et al. [16] |
|---|---|---|---|---|
| Number of ring oscillators | 2 | 2 | 130 | 512 |
| Average independent response bits | 31 | 1 | 318 | 511 |
| Bits per Ring | 16.5 | 0.5 | 2.44 | $\approx 1$ |

In Table 3, we list some designs to extract responses from ROs and make comparisons of the variable *Bits per Ring*. The result shows that in our architecture, we can extract 16.5 bits entropy per ring, which is 7 times larger than that of Habib et al. [24], and moreover it is 31 times larger than that of general RO PUF.

### 4.4 The Evaluation Result in Other FPGAs

The above reliability and uniqueness are tested on Xilinx Virtex-5 FPGAs. We also conduct experiments on Xilinx Virtex-6 and Kintex-7 FPGAs. The results are that the intra-distance is about 1.68% and the inter-distance is about 49.12% on Virtex-6 FPGAs. In addition, the intra-distance is about 1.62% and the inter-distance is about 48.95% on Kintex-7 FPGAs. Therefore, our new proposed FROPUF construction can be available on these newfashioned FPGA products.

## 5 Further Discussion

**Algorithm 2:** Simulation Algorithm of Second Order Difference Calculation

**Settings:** · There are two ROs, $A$ and $B$.
· Both $A$ and $B$ have 5-bit configuration inputs.
· According to 32 different inputs, there will be $Counter_{A_j}$ and $Counter_{B_j}$, $1 \leq j \leq 32$

**Output:** · Response $r_i$, $1 \leq i \leq 496$

```
i ⟵ 0
for m = 1 to 32 do
    for n = m+1 to 32 do
        i ⟵ i + 1
        ΔCounter_A(m,n) ⟵ Counter_Am - Counter_An
        ΔCounter_B(m,n) ⟵ Counter_Bm - Counter_Bn
        if ΔCounter_A(m,n) ≥ ΔCounter_B(m,n) then
            r_i ← 1
        else
            r_i ← 0
        end if
    end for
end for
```

In section 3, our proposed second order difference calculation just gets a 31-bit response. In the description of Algorithm 2, we can get a 496-bit response from two ring oscillators based on FROPUF. Obviously, The Shanon entropy of this 496-bit response is less than 496 bits. On the observation of section 3, a lower bound entropy of this 496-bit response is 31 bits. Based on the model proposed

in section 3, we can calculate the Shanon entropy of this 496-bit response as follows.

According to the model in section 3, we have the formula (16) as follows because we assume no prior information about the response when only the process variation is present.

$$Prob(r_i = 1) = Prob(r_i = 0) = 0.5 \quad (1 \leq i \leq 496) \tag{16}$$

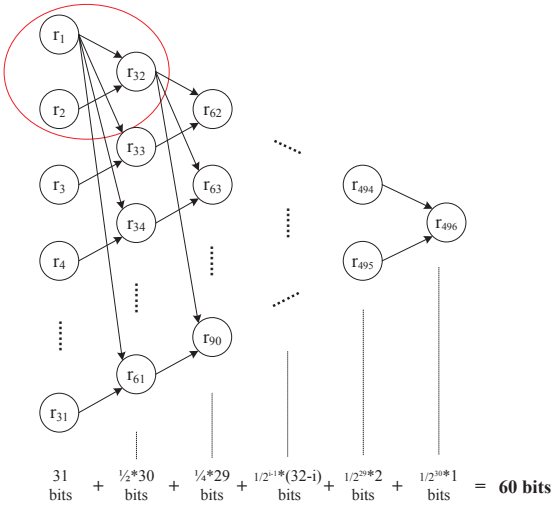


Fig. 9: The Shanon Entropy of the 496-bit response

However, when we know some bits' information of these 496 bits, the probability of the remaining bits will change. For example, if $r_1$ and $r_2$ is known, the probability of $r_{32}$ is shown as follows.

$$Prob(r_{32} = 0) = \begin{cases} 1 & if \quad r_1 = 1 \quad and \quad r_2 = 0, \\ 0.5 & if \quad r_1 = 1 \quad and \quad r_2 = 1, \\ 0.5 & if \quad r_1 = 0 \quad and \quad r_2 = 0, \\ 0 & if \quad r_1 = 0 \quad and \quad r_2 = 1. \end{cases} \tag{17}$$

Figure 9 shows the relative relationships between these 496 bits response. Based on formula (16) and Figure 9, we can calculate the Shanon entropy of the 496-bit response as follows. The 31 bits in the first row of Figure 9 have 31 bits Shanon entropy, and the 30 bits in second row have 15 bits Shanon entropy and so on. We can acquire that the responses in the $i^{th}$ row have $\frac{1}{2^{i-1}}(32 - i)$ bits Shanon entropy. As a result, the Shanon Entropy of this 496-bit response is $\sum_{i=1}^{31} \frac{1}{2^{i-1}}(32 - i) = 60$ bits.

## 6 Conclusion

In this paper, we propose a new architecture called Further RO PUF, which takes advantage of LUT's fine control, for the purpose to achieve more random process variation. Then through the second order difference calculation, we can reduce the influence of systematic variation and environmental fluctuation neatly to ensure the reliability of our new proposed PUF. The key point of FROPUF is that we can extract at least 31 bits entropy from only two ring oscillators and the Shanon Entropy of the response from two ring oscillators is 60 bits. Through both simulation and practical experiments, the intra-distance of FROPUF is just 1.85% at 27°C and will not exceed 10% with rough temperature changes. The inter-distance is about 49.32%, which can guarantee the uniqueness of different PUF instances.

## References

1. Ruhrmair, Ulrich and Holcomb, Daniel E, "PUFs at a glance," *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pp. 1–6, 2014.
2. Gassend, Blaise and Clarke, Dwaine and Van Dijk, Marten and Devadas, Srinivas, "Silicon physical random functions," *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 148–160, 2002.
3. Katzenbeisser, Stefan and Kocabaş, Ünal and Rožić, Vladimir and Sadeghi, Ahmad-Reza and Verbauwhede, Ingrid and Wachsmann, Christian, "PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon," *Cryptographic Hardware and Embedded Systems–CHES 2012*, pp. 283–301, 2012.
4. Pappu, Ravikanth and Recht, Ben and Taylor, Jason and Gershenfeld, Neil, "Physical one-way functions," vol. 297, pp. 2026–2030, American Association for the Advancement of Science, 2002.
5. Maes, Roel and Van Herrewege, Anthony and Verbauwhede, Ingrid, "Pufky: A fully functional puf-based cryptographic key generator," *Cryptographic Hardware and Embedded Systems–CHES 2012*, pp. 302–319, 2012.
6. Guajardo, Jorge and Kumar, Sandeep S and Schrijen, Geert-Jan and Tuyls, Pim, "FPGA intrinsic PUFs and their use for IP protection," *Cryptographic Hardware and Embedded Systems-CHES 2007*, pp. 63–80, 2007.
7. Roy, Jarrod A and Koushanfar, Farinaz and Markov, Igor L, "EPIC: Ending piracy of integrated circuits," *Proceedings of the conference on Design, automation and test in Europe*, pp. 1069–1074, 2008.
8. Koeberl, Patrick and Li, Jiangtao and Maes, Roel and Rajan, Anand and Vishik, Claire and Wójcik, Marcin, "Evaluation of a PUF Device Authentication Scheme on a Discrete 0.13 um SRAM," *Trusted Systems*, pp. 271–288, 2012.
9. Devadas, Srinivas and Suh, Edward and Paral, Sid and Sowell, Richard and Ziola, Thomas and Khandelwal, Vivek, "Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications," *RFID, 2008 IEEE International Conference on*, pp. 58–64, 2008.
10. Tuyls, Pim and Škorić, Boris, "Strong authentication with physical unclonable functions," *Security, Privacy, and Trust in Modern Data Management*, pp. 133–148, 2007.

11. Holcomb, Daniel E., and Kevin Fu.Springer Berlin Heidelberg, "Bitline PUF: Building Native Challenge-Response PUF Capability into Any SRAM," in *Cryptographic Hardware and Embedded Systems CHES*, 2014. 510-526.
12. Kumar, Sandeep S and Guajardo, Jorge and Maes, Roel and Schrijen, G-J and Tuyls, Pim, "The butterfly PUF protecting IP on every FPGA," *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pp. 67–70, 2008.
13. Suzuki, Daisuke and Shimizu, Koichi, "The glitch PUF: A new delay-PUF architecture exploiting glitch shapes," *Cryptographic Hardware and Embedded Systems, CHES 2010*, pp. 366–382, 2010.
14. Maes, Roel and Tuyls, Pim and Verbauwhede, Ingrid, "Intrinsic PUFs from flip-flops on reconfigurable devices," *3rd Benelux workshop on information and system security (WISSec 2008)*, 2008.
15. Suh, G Edward and Devadas, Srinivas, "Physical unclonable functions for device authentication and secret key generation," *Proceedings of the 44th annual Design Automation Conference*, pp. 9–14, 2007.
16. Maiti, Abhranil, and Patrick Schaumont, "Improved ring oscillator PUF: an FPGA-friendly secure primitive," *Journal of cryptology*, vol. no. 2 375-397, 2011.
17. Gao, Mingze, Khai Lai, and Gang Qu, "A Highly Flexible Ring Oscillator PUF," in *Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference. ACM*, 2014.
18. Rahman, Tauhidur and Forte, Domenic and Fahrny, Jim and Tehranipoor, Mohammad, "ARO-PUF: an aging-resistant ring oscillator PUF design," in *Proceedings of the conference on Design, Automation & Test in Europe*, 2014.
19. Cherkaoui, Abdelkarim, Viktor Fischer, Alain Aubert, and Laurent Fesquet, "Comparison of self-timed ring and inverter ring oscillators as entropy sources in FPGAs," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012, pp. 1325-1330. IEEE*, 2012.
20. Yin, Chi-En, Gang Qu, and Qiang Zhou, "Design and implementation of a group-based RO PUF," in *Proceedings of the Conference on Design, Automation and Test in Europe, pp. 416-421. EDA Consortium*, 2013.
21. Merli, Dominik, Johann Heyszl, Benedikt Heinz, Dieter Schuster, Frederic Stumpf, and Georg Sigl, "Localized electromagnetic analysis of RO PUFs," in *In Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium on, pp. 19-24. IEEE*, 2013.
22. Dodis, Yevgeniy and Reyzin, Leonid and Smith, Adam, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *Advances in cryptology-Eurocrypt*, pp. 523–540, 2004.
23. Bösch, Christoph, Jorge Guajardo, Ahmad-Reza Sadeghi, Jamshid Shokrollahi, and Pim Tuyls, "Efficient helper data key extractor on FPGAs," in *In Cryptographic Hardware and Embedded SystemsCCHES 2008, pp. 181-197. Springer Berlin Heidelberg*, 2008.
24. Habib, Bilal, Kris Gaj, and Jens-Peter Kaps, "FPGA PUF Based on Programmable LUT Delays," in *Digital System Design (DSD), 2013 Euromicro Conference on. IEEE, 2013.*
25. Majzoobi, Mehrdad, Farinaz Koushanfar, and Srinivas Devadas, "FPGA-based true random number generation using circuit metastability with adaptive feedback control," in *Cryptographic Hardware and Embedded Systems CHES 2011. Springer Berlin Heidelberg, 17-32*, 2011.

26. Majzoobi, Mehrdad and Koushanfar, Farinaz and Devadas, Srinivas, "FPGA PUF using programmable delay lines," *Information Forensics and Security (WIFS), 2010 IEEE International Workshop on*, 2010.
27. Rührmair, Ulrich and Sehnke, Frank and Sölter, Jan and Dror, Gideon and Devadas, Srinivas and Schmidhuber, Jürgen, "Modeling attacks on physical unclonable functions," pp. 237–249, 2010.
28. Maes, Roel and Verbauwhede, Ingrid, "Physically unclonable functions: A study on the state of the art and future research directions," *Towards Hardware-Intrinsic Security*, pp. 3–37, 2010.
29. Wold K, Tan C H, "Analysis and Enhancement of Random Number Generator in FPGA Based on Oscillator Rings," in *Proceedings of the 2008 International Conference on Reconfigurable Computing and FPGAs-Volume 00. IEEE Computer Society*, 2008: 385-390.
30. Maiti A, Casarona J, McHale L, "A large scale characterization of RO-PUF," in *Hardware-Oriented Security and Trust (HOST), 2010 IEEE International Symposium on. IEEE*, 2010: 94-99.
31. Maes, Roel and Tuyls, Pim and Verbauwhede, Ingrid, "Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs," *Cryptographic Hardware and Embedded Systems-CHES 2009*, pp. 332–347, 2009.