Semi-invasive EM Attack on FPGA RO PUFs and Countermeasures

Dominik Merli, Dieter Schuster,
Frederic Stumpf
Fraunhofer Research Institution AISEC
Munich, Germany
firstname.lastname@aisec.fraunhofer.de

Georg Sigl
Institute for Security in Information Technology
TU München
Munich, Germany
sigl@tum.de

ABSTRACT

It is often argued that Physical Unclonable Functions (PUFs) are resistant against invasive and semi-invasive attacks since these attacks would damage the underlying PUF structure resulting in a different PUF response. In this paper, we demonstrate exemplarily that this assumption does not hold for a Ring Oscillator (RO) PUF implemented on a Xilinx Spartan 3 FPGA, where we were able to perform a semi-invasive attack. We present analysis methods to identify ring oscillator frequencies and to map them to their corresponding oscillators. We practically prove that it is possible to recover the generated RO PUF response bits with this approach. To harden RO PUFs against side-channel analysis, we also propose a RO PUF concept not leaking useful information through the side-channel of electro-magnetic radiation.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Security, Measurement, Experimentation

Keywords

Physical Unclonable Function, PUF, Field Programmable Gate Array, FPGA, Ring Oscillator, Semi-invasive Attack, Side-Channel Analysis, SCA, Electro Magnetic Analysis

1. INTRODUCTION

Field Programmable Gate Arrays (FPGAs) are important components for a considerable number of embedded systems. While their properties facilitate high performance, fast prototyping and hardware updates even after market launch, theft of Intellectual Property (IP) poses a serious threat to FPGA systems.

The use of Physical Unclonable Functions (PUFs) in FPGAs seems to be a promising solution for IP protection [13, 3]. PUFs are physical structures which map challenges to responses according to unique unavoidable material variations. These sub-micron manufacturing variations do not allow to manufacture a second device with the same properties, even by the original manufacturer. Therefore, IP blocks can be bound to a specific device and refuse functioning on any other device.

Another reason for using PUFs on FPGAs is their ability to securely store a secret key and to reliably reconstruct it with the help of a fuzzy extractor [1]. These algorithms cancel noise present in PUF responses during an information reconciliation phase. Afterwards, a privacy amplification is performed to obtain a uniformly distributed key which can be used for conventional cryptography.

During the last years, several PUF architectures [2, 15, 3, 5, 4] have been proposed for FPGAs. PUFs based on Ring Oscillators (ROs) have one advantage [6, 10] over other FPGA PUF structures, which enables them to extract higher quality PUF responses. They do not require balanced routing, which is hard to achieve in FPGA designs. Further, the quality of RO PUFs is improved continuously [7, 9]. Therefore, RO PUFs can be considered as one of the most important FPGA PUFs.

Often, PUFs are assumed to naturally protect a chip from tampering and invasive attacks. This might be true for a Coating PUF [16], because it covers the whole surface of an integrated circuit, but has not been analysed for Silicon PUFs yet. Therefore, in order to establish RO PUFs as a high security primitive on FPGAs, their sensitivity to FPGA decapsulation and their vulnerability to side-channel attacks has to be investigated and, if necessary, suitable countermeasures have to be developed.

In this contribution, we show that the ring oscillator frequencies measured on an FPGA, only slightly change when the device is decapsulated. Since there is no significant influence on the physical structure, it is feasible to obtain side-channel information about an RO PUF by semi-invasive attacks like near-field electro-magnetic (EM) cartography methods [12]. Further, we give analysis methods and demonstrate that it is possible to extract a full model of an RO PUF from EM measurements and present concepts to eliminate side-channel leakage of RO PUFs.

The remainder of this paper is organised as follows. Section 2 gives a brief overview over previous attacks on RO PUFs. The basic concept and recent enhancements of the RO PUF are explained in Section 3. In Section 4, the decapsulation of an FPGA without significantly influencing RO frequencies is demonstrated. Section 5 presents the EM analysis methods for RO PUFs, followed by Section 6 describing our practical measurements and results. To counteract these attacks, we propose an improved RO PUF concept in Section 7. Our contribution is completed by a conclusion in Section 8.

2. RELATED WORK

Until now, not much work regarding PUF attacks or side-channel analysis of PUFs has been published.

Tuyls et. al proposed a Coating PUF architecture [16] to protect the integrated circuits inside a chip package from tampering and invasive attacks. Further, they exploit the random coating capacitances to generate a unique key. The coating is measured from sensors lying under it, which results in a tamper-proof PUF device. However, this property is not generalisable to all PUF types. Our work evaluates the influences of decapsulation and semi-invasive attacks on an RO PUF implemented on an FPGA.

In [11], Rührmair et al. demonstrated that RO PUFs can be modelled, e.g. by machine learning algorithms, if an attacker is able to collect a sufficient number of Challenge-Response Pairs (CRPs). Also, a full read-out of RO PUFs is possible as remarked by the authors. However, these attacks become infeasible for key generation scenarios where a fixed sequence of challenges is used to generate a PUF response bitstring which is processed by a fuzzy extractor [1], except the attacker is able to inject challenges and extract responses invasively.

One of our previous works [8] theoretically investigates potential side-channel leakage in RO PUFs. We supposed that RO PUFs are vulnerable to side-channel attacks because the frequencies of the ring oscillators themselves, the operations of the measuring counters or the used comparator might leak critical side-channel information. These ideas are picked up in this contribution and a proof-of-concept attack is implemented to confirm these weaknesses.

3. RING OSCILLATOR PUF

In 2007, Suh and Devadas introduced the Ring Oscillator PUF (RO PUF) [15] as shown in Figure 1. Material and manufacturing variations present in FPGAs are exploited by measuring ring oscillator frequencies. A challenge selects two ring oscillators, RO_a and RO_b, from an array of oscillators and connects each of them to a counter, which is clocked by the connected oscillator signal and therefore represents its frequency. After a fixed runtime has elapsed, the counter values are compared and a PUF response bit is generated according to the following rule: $counter(RO_a) > counter(RO_b) \rightarrow 0$ and $counter(RO_a) < counter(RO_b) \rightarrow 1$.

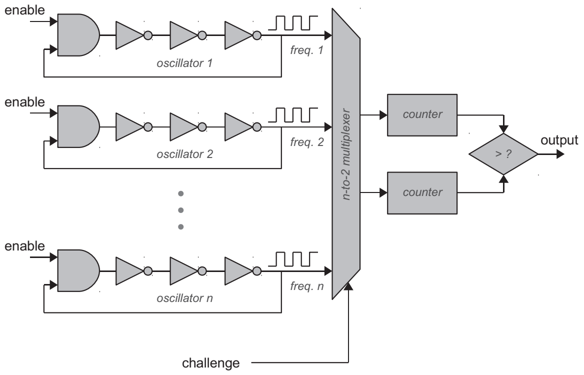


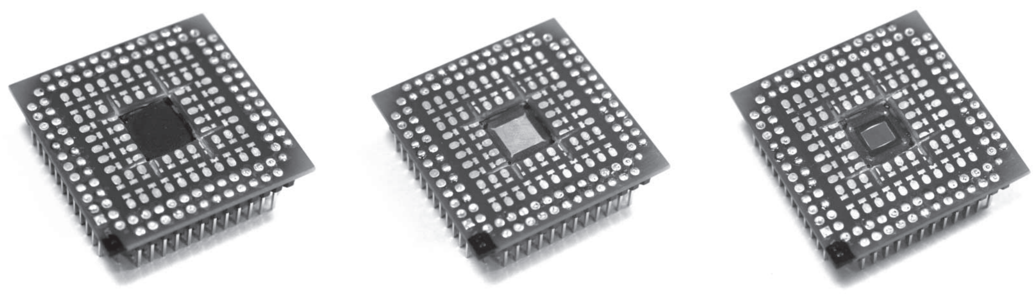
Figure 1: RO PUF architecture

Maiti and Schaumont [7] extended the basic RO PUF concept to optimize the quality, namely reliability and uniqueness, of RO PUF responses. Their approach does not use simple ring oscillators, but configurable ones where different delay components can be selected by a 3-bit configuration interface, allowing eight different configurations. In an enrolment phase, all configurations of an oscillator are tested and the most reliable one is stored and can be seen as a kind of helper data. This concept leads to higher quality responses, but the frequency measuring architecture stays the same and is therefore as vulnerable to side-channel attacks as the standard version.

When considering RO PUFs for secret key generation using fuzzy extractors, it is necessary to extract a substantial amount of PUF response bits (e.g. more than 1000 bits for a 128-bit key). In [9], a chaining method is proposed to compare only adjacent oscillators, because of location dependent intra-die conditions. With this method, it is possible to extract an RO PUF response R with $M-1$ bits from M oscillators. The chain of oscillators is evaluated pair by pair, i.e. $eval(RO_1, RO_2)$, $eval(RO_2, RO_3)$, ..., $eval(RO_{M-1}, RO_M)$, which can be realized by a single counter generating all necessary challenges.

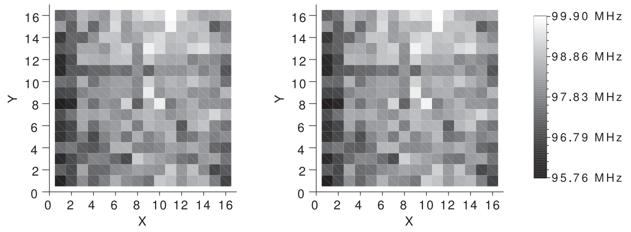
4. INFLUENCES OF DECAPSULATION ON RO PUFS IMPLEMENTED ON FPGAS

A popular argument for using PUFs instead of conventional key storage like non-volatile memory is the natural tamper resistance property of PUFs. This assumption holds true if an attacker aims at manipulating the PUF structure itself, but does not automatically protect a microchip and its package from tampering. In this section, we show that decapsulating the backside of a *Xilinx Spartan XC3S200* FPGA in a VQ100 package, does not have a significant influence on internal ring oscillator frequencies. As a result, semi-invasive attacks like on-die EM cartography become feasible.



(a) (b) (c)

Figure 2: Step-by-step backside decapsulation of an FPGA



(a) before (b) after

Figure 3: Frequency maps for ROs with 7 inverters

Backside Decapsulation. In [14], Skorobogatov suggests to open an FPGA package from the backside, which can be done manually and without the application of acids. First, the FPGA was soldered on a printed circuit board having a hole in the center allowing to access the backside of the package as shown in 2(a). We used a drill to remove the plastic packaging down to the copper lead-frame of the package, resulting in Figure 2(b). After cutting the corner connections of the ground-plate with a carpet cutter, we were able to take away the plate, exposing the backside of the die. Finally, we scraped off the remaining glue and cleaned the die's surface with acetone. Figure 2(c) shows the final result.

Before and After. In order to investigate the influences of decapsulation, we implemented an array of $M = 256$ oscillators on a Xilinx Spartan XC3S200 FPGA occupying 256 Configurable Logic Blocks (CLBs). Additionally, a multiplexer, a 16-bit counter and control logic were integrated to enable frequency measurements for all oscillators. Each oscillator was measured 10,000 times over a period of 4096 clock cycles at 20 MHz in a temperature-controlled room to precisely characterise the given RO frequencies. Measurements were conducted for ROs consisting of 3, 5 and 7 inverters before and after decapsulation. Besides generating frequency maps, as exemplarily shown in Figure 3, to compare the frequency fingerprint visually, the mean of all frequencies $\bar{f}$ and their mean relative standard deviation $\overline{\sigma_{rel}}$ were calculated. Further, the resulting $(M-1)$-bit PUF responses $R_i, i = 1...10000$ were determined by frequency comparison as defined for an RO PUF. Additionally, the mean RO PUF response $\bar{R} = \sum_{i=1}^{10000} R_i/10000$ was calculated. The mean value $\overline{HD}_{rel}(\bar{R}; R_i)$ of the relative Hamming distance $HD_{rel}(\bar{R}; R_i) = HD(\bar{R}; R_i)/(M-1) \times 100\%$ indicates the level of present noise.

Table 1: Characterisation of RO frequencies

ROs with 3 inverters			
measurement	***unit***	***before***	***after***
$\bar{f}$	MHz	197.96	198.07
$\overline{\sigma_{rel}}$	%	1.39	1.83
$\overline{HD}_{rel}(\bar{R}; R_i)$	%	1.24	1.25
$\max(HD_{rel}(\bar{R}; R_i))$	%	11.81	15.35
$\bar{f}_{after} - \bar{f}_{before}$	MHz	+0.11	
$HD_{rel}(\bar{R}_{before}; \bar{R}_{after})$	%	0.79	
ROs with 5 inverters			
measurement	***unit***	***before***	***after***
$\bar{f}$	MHz	132.56	132.70
$\overline{\sigma_{rel}}$	%	0.88	0.05
$\overline{HD}_{rel}(\bar{R}; R_i)$	%	1.93	2.63
$\max(HD_{rel}(\bar{R}; R_i))$	%	7.87	6.69
$\bar{f}_{after} - \bar{f}_{before}$	MHz	+0.14	
$HD_{rel}(\bar{R}_{before}; \bar{R}_{after})$	%	2.76	
ROs with 7 inverters			
measurement	***unit***	***before***	***after***
$\bar{f}$	MHz	98.10	98.20
$\overline{\sigma_{rel}}$	%	0.05	2.20
$\overline{HD}_{rel}(\bar{R}; R_i)$	%	1.60	2.60
$\max(HD_{rel}(\bar{R}; R_i))$	%	5.12	11.42
$\bar{f}_{after} - \bar{f}_{before}$	MHz	+0.10	
$HD_{rel}(\bar{R}_{before}; \bar{R}_{after})$	%	2.36	

After decapsulating and again characterising the RO frequencies, the most meaningful characteristic, the resulting frequency maps, did not show significant changes, as shown in Figure 3 for 7-inverter ROs. As shown in Table 1, we found that the differences of the mean frequencies $\bar{f}_{after} - \bar{f}_{before}$ were smaller than 0.15 MHz and the Hamming distance $HD_{rel}(\bar{R}_{before}; \bar{R}_{after})$ between the mean response before and after were in the area of the mean relative Hamming distance $\overline{HD}_{rel}(\bar{R}; R_i)$ and below the level of maximum noise $\max(HD_{rel}(\bar{R}; R_i))$, which are both measures to determine a suitable error correction code for a fuzzy extractor. This means, that a decapsulation alters only a few bits of an RO PUF response, and as PUF responses are inherently noisy, this influence has to be tolerated by the subsequent key generation algorithm in any case to guarantee reliable reproduction. Therefore, decapsulation does not lead to significant change in RO PUF behaviour and does not hinder an attacker from applying semi-invasive attacks.

5. EM ANALYSIS OF RO PUFS

As indicated in [8], RO PUFs suffer from several potential side-channel leakage sources. This analysis focusses on the most critical threat, which is the extraction and allocation of ring oscillator frequencies. This would give an attacker the possibility to fully characterize a given RO PUF.

The following analysis methods can be performed step by step. First, the frequency range of the analysed ROs has to be determined. Then, the die area of highest RO frequency leakage can be identified. After limiting frequency range and die location, two distinct frequencies for each comparison of two ROs can be found. The last step links these frequencies to their corresponding ROs, thus generating a full RO PUF model.

The EM emanation of an RO PUF FPGA design strongly depends on the measurement location. The presented analysis methods always require a grid $G = \{g_{x,y} | x = 1...X, y = 1...Y\}$ of EM measurement traces $g_{x,y}$ at locations (x, y) over an FPGA die. The required resolution of this grid, width X and height Y, is determined by device, equipment and implementation parameters.

In the following, we present EM analysis methods for an RO PUF generating an N bit response from N comparisons of two ring oscillators $RO_{a,n}$ and $RO_{b,n}$ for $n = 1...N$. For an RO PUF, a comparison n is defined as the comparison between the two counters measuring the RO frequencies: $\mathit{counter}(RO_a) > \mathit{counter}(RO_b) \rightarrow 0$ and $\mathit{counter}(RO_a) < \mathit{counter}(RO_b) \rightarrow 1$.

5.1 RO Frequency Range

The first step in RO PUF EM analysis is to identify the frequency range in which the ring oscillators operate. Since clock signals, their harmonics and other disturbances are spread over the whole frequency spectrum, it is essential to concentrate on a fixed frequency range for analysis.

Our approach is based on the idea that every comparison has two specific frequencies, corresponding to the two compared oscillators. All other frequencies are constant or noisy for all comparisons and can therefore be neglected. Thus, we calculate a frequency amplitude spectrum $\mathbf{s_n}(g_{x,y}), n = 1, ..., N$, which is a vector consisting of K frequency amplitude values $s_{n,k}(g_{x,y}), k = 1, ..., K$ generated by Fourier transforming every trace $g_{x,y}$ for every comparison time slot n. Then, for every frequency, we accumulate the differences of frequency amplitudes between all N comparisons in each trace. Finally, the absolute values of the accumulations represent the amplitude difference spectrum $\mathbf{s_{diff}}$.

$g_{x,y} \in G$

$\mathbf{s_n}(g_{x,y}) = [s_{n,1}(g_{x,y}),\ \dots\ , s_{n,K}(g_{x,y})]$

$\mathbf{s_{diff}} = [s_{\mathit{diff},1},\ \dots\ , s_{\mathit{diff},K}]$ *with*

$$s_{\mathit{diff},k} = \Big|\sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{i=1}^{N-1} \sum_{j=i}^{N} (s_{i,k}(g_{x,y}) - s_{j,k}(g_{x,y}))\Big|,\ k = 1...K$$

There, high difference peaks indicate frequencies which are present in some comparisons but missing in others, i.e. the behaviour of these frequencies is as we would expect it for RO frequencies. With the highest amplitudes in this plot, the RO frequency range $[f_{lo}\ ...\ f_{hi}]$, including K' spectrum values, can be identified.

5.2 Area of Leakage

One way of generating a map $P = \{p_{x,y} | x = 1...X, y = 1...Y\}$ depicting location dependet leakage is to plot the mean difference between comparisons in the identified frequency range $[f_{lo}\ ...\ f_{hi}]$ for every trace $g_{x,y}$. Based on the reduced amplitude spectra $\mathbf{s'_{x,y,n}}$, every map pixel $p_{x,y} \in P$ is calculated as:

$$\mathbf{s'_n}(g_{x,y}) = [s'_{n,1}(g_{x,y}),\ \dots\ , s'_{n,K'}(g_{x,y})]$$
$$= [s(f_{lo}(g_{x,y})),\ \dots\ , s(f_{hi}(g_{x,y}))]$$

$$p_{x,y} = \frac{\sum_{k=1}^{K'} \Big|\sum_{i=1}^{N-1} \sum_{j=i}^{N} (s'_{i,k}(g_{x,y}) - s'_{j,k}(g_{x,y}))\Big|}{K'}$$

Another possibility is the exploitation of the leaking amplitudes in the identified frequency range of all ROs. One can plot the mean of all amplitudes in the found frequency range for every measurement point. This map gives a hint which area of a design discloses high amplitudes of RO frequencies.

$$p_{x,y} = \frac{\sum_{k=1}^{K'} \sum_{i=1}^{N} s'_{i,k}(g_{x,y})}{K' \times N}$$

With both of the two presented map generation methods, all traces $g'_{x,y}$ where a high leakage is detected, can be combined into a group $G_{leak} \subset G$.

5.3 Distinct RO Frequencies

In the third step, we focus only on the found frequency range and leaking traces in G_{leak}. We calculate a frequency amplitude spectrum $\mathbf{s_n}(g'_{x,y}), n = 1...N$ for every comparison n in each trace $g'_{x,y} \in G_{leak}$ as extracted from the maps generated in the previous step. We average all spectra to obtain a noise reduced average spectrum $\bar{\mathbf{s}}_\mathbf{n}$ for every RO frequency comparison n.

$g'_{x,y} \in G_{leak}$

$$\bar{\mathbf{s}}_\mathbf{n} = \frac{\sum_{g'_{x,y} \in G_{leak}} \mathbf{s_n}(g'_{x,y})}{|G_{leak}|}$$

Afterwards, one should be able to extract two distinct frequencies f_1 and f_2 (with high amplitudes) in every spectrum from left to right. These frequencies represent the two RO signals of each comparison. A list L with elements l_1, l_2, ..., l_N containing the comparison n linked with the found frequencies $f_{1,n}$ and $f_{2,n}, n = 1...N$ will be generated as preparation for the last analysis step.

$L = \{l_n = (n; f_{1,n}; f_{2,n})\},\ n = 1...N$

5.4 RO PUF Modelling

In the last step, the challenges C_1, ..., C_N consisting of RO pairs $RO_{a,n}$ and $RO_{b,n}$ of comparison n come into play. They constitute the missing link between the found frequencies and their ring oscillators.

When generating a secret response from a number M of oscillators, a response length of $M-1$ bits can be achieved when applying a chaining method [9] with $M-1$ comparisons. There, the chaining rule $RO_{a,n} = RO_{b,n-1}, n = 1...N-1$ (second RO in one comparison will be first RO in following comparison) guarantees independent RO comparisons. However, this rule can be exploited to correctly allocate our measured frequencies to their corresponding ring oscillators because it leads to the condition that one frequency must be common to subsequent comparisons:

$$(f_{1,n} = f_{1,n-1}) \vee (f_{1,n} = f_{2,n-1}) \vee$$
$$(f_{2,n} = f_{1,n-1}) \vee (f_{2,n} = f_{2,n-1})$$

Therefore, the frequency $f_{RO_{a,n}}$ of the oscillator $RO_{a,n}$ can be determined as follows:

$$f_{RO_{a,n}} = \begin{cases} f_{1,n}: & \textit{if } (f_{1,n} = f_{1,n-1}) \vee (f_{1,n} = f_{2,n-1}) \\ f_{2,n}: & \textit{if } (f_{2,n} = f_{1,n-1}) \vee (f_{2,n} = f_{2,n-1}) \end{cases}$$

Clearly, $f_{RO_{b,n}}$ must be the other frequency of the two possible frequencies. With this allocation, it is possible to solve the comparisons and obtain the RO PUF response bits.

To obtain a full model of an RO PUF, the list of all M ROs linked with their frequencies f_{RO_m} has to be calculated:

$$f_{RO_m} = \begin{cases} f_{1,m}: & \textit{if } (f_{1,m} = f_{1,m-1}) \vee (f_{1,m} = f_{2,m-1}) \\ f_{2,m}: & \textit{if } (f_{2,m} = f_{1,m-1}) \vee (f_{2,m} = f_{2,m-1}) \end{cases}$$

The frequencies of the very first and very last oscillator in the chain will only appear once in the list L, but knowing the allocation of the RO compared to them, the frequency can be linked to the left over oscillator.

5.5 Discussion on Analysis Limits

The main scenario of our analysis methods is secret key generation based on RO PUFs. There, a fixed sequence of N challenges is applied step by step as described in [9] to efficiently use the number of integrated ROs. We assume that this challenge sequence is not concealed, but can be regarded as known. However, even in the case that only a part or nothing about the challenge sequence is known, but an efficient challenge sequence is used ($M-1$ oscillators are measured at least twice), then the list L can be sorted in a way that subsequent comparisons contain one common frequency. Afterwards, the analysis can be continued as described.

Further, the runtime of a single RO frequency comparison is assumed to be known, i.e. it is possible to separate subsequent frequency comparisons. Even if the runtime would not be known, an attacker could estimate it by dividing the whole measurement time by the number of generated response bits.

In case of very close RO frequencies in one comparison, it might not be possible to distinguish between them. Since this effect is also responsible for noise in RO PUF responses, the uncertainty of our analysis only reflects this noise. Algorithms like fuzzy extractors or error-tolerant protocols can handle a specified amount of noise anyway, which means, that an attacker only has to extract the PUF response with a certain precision and eventual extraction errors can be seen as noise contained in the PUF response.

6. PRACTICAL RESULTS

For practical verification of our formal analysis methods, we implemented an RO PUF prototype with chained RO comparisons to extract $M-1$ bits from M ring oscillators as described in [9]. It also has an RO select logic, which saves energy compared to enabling all oscillators during every measurement.

We implemented 9 ring oscillators, each built out of 7 inverters. During 8 comparisons, each of them lasting 4096 cycles of the 20 MHz system clock, 8 PUF response bits were generated. We reduced the number of oscillators to focus on the evaluation of the basic attack while keeping the effort low.

The design was loaded on a decapsulated *Xilinx Spartan XC3S200* FPGA, for which we have shown in Section 4 that removing the backside package does not influence the ring oscillators' frequencies.

6.1 Measurement Setup

In order to precisely measure the near-field EM emanation of our FPGA design, we used a *Langer ICR HH 150* probe. It consists of a horizontal coil with an inside diameter of 150 μm and has a specified resolution of 100 μm. Since, we aimed at capturing the frequencies of very small components, we placed the probe so close to the FPGA die that it almost touched the die's backside surface as shown in Figure 4. An x-y-table allowed us to move the FPGA board in tiny steps under the fixed EM probe. We chose a grid of 50 × 42 measurement points over the 4.8 mm × 4.0 mm die to obtain a map of location dependent EM radiation of our RO PUF design.

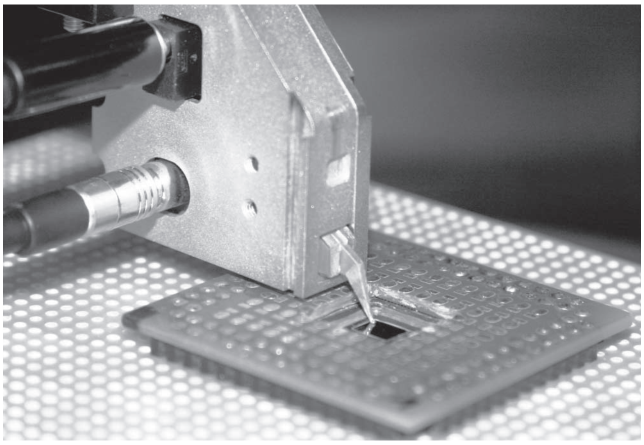


Figure 4: Near-field probe close to FPGA die

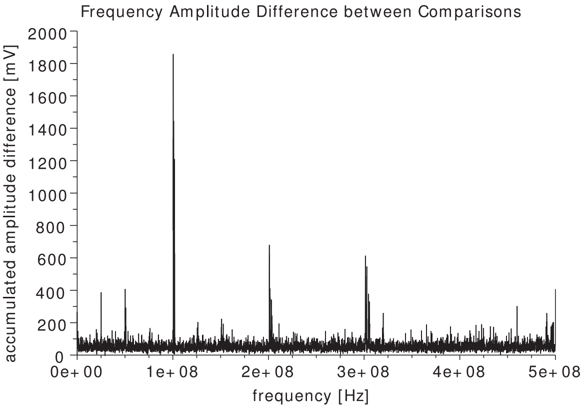


Figure 5: Frequency amplitude difference spectrum

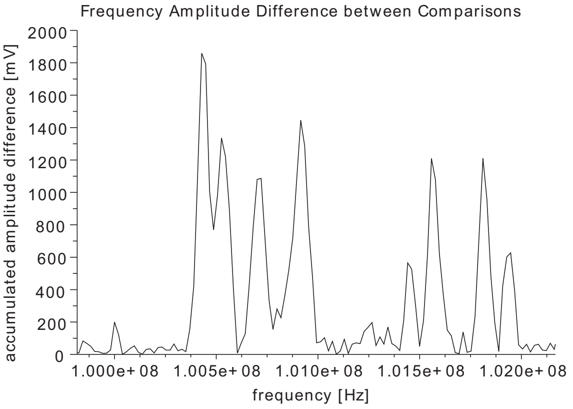


Figure 6: Spectrum range with highest peaks

The probe was connected to an oscilloscope by a 30 dB amplifier. The function generator for the device's clock signal and the oscilloscope were synchronized. The oscilloscope was set to a sampling rate of 1 GS/s and recorded 600,000 samples per trace.

6.2 RO Frequency Range

The first step to disclose the implemented ring oscillators was to identify their frequency range. We calculated the accumulated frequency amplitude difference spectrum as described in Section 5.1. Figure 5 shows the spectrum for our RO PUF implementation. One can find significant peaks in this spectrum signalizing that these frequency amplitudes change from one comparison to another. The largest peak is located at around 100 MHz. By closer inspection (see Figure 6), the exact frequency range can be determined to reach from $f_{lo} = 100.3\,MHz$ to $f_{hi} = 102.1\,MHz$. Other peaks found in the spectrum correspond to harmonics of the RO frequencies or other comparison dependant frequencies, e.g. originating from the components of the measuring counter.

6.3 Area of Interest

Next, we generated maps of our FPGA die, to find the locations where the oscillators' frequencies can be observed

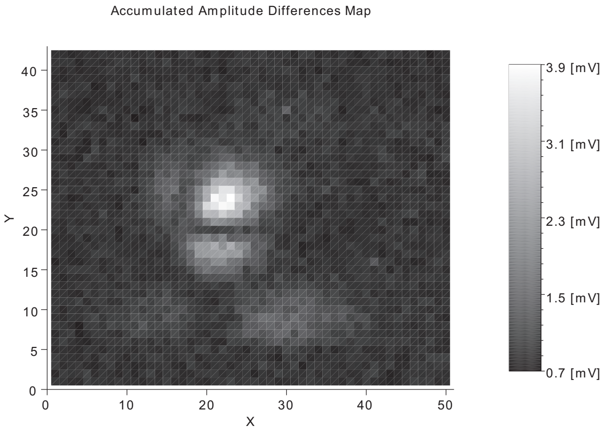


Figure 7: Frequency amplitude difference map

best. The two proposed methods from Section 5.2 were applied and produced similar results. One can see light areas in Figure 7 which indicate amplitude differences between comparisons and therefore represent points of interest for an attacker. Thus, the area of main leakage can be limited to the locations within the rectangle spanning from location (18, 15) to point (28, 29). This means, that the number of traces to process can be reduced from 2100 to 165 informative ones, which accelerates the analysis and enhances the quality of the final results at the same time.

6.4 RO Frequency List

As a last trace processing step, we calculated the mean frequency amplitude spectra for every comparison slot as described in Section 5.3 and plotted them one beneath the other as shown in Figure 8. One can see, that every comparison contains two distinct frequency peaks, representing two ring oscillators. Table 2 shows an RO frequency comparison list as extracted from the spectra depicted in Figure 8.

Table 2: RO frequencies comparison list

comparison n	$f_1[MHz]$	$f_2[MHz]$
1	100.71	100.91
2	100.93	101.56
3	101.58	101.94
4	100.87	101.94
5	100.88	101.45
6	100.55	101.45
7	100.44	100.55
8	100.44	101.82

6.5 RO PUF Model

To recover the secret response generated by our RO PUF implementation, we finally exploited the chained challenge sequence to decode the found frequencies. Since we know that the chain of oscillators is evaluated pair by pair, i.e. $eval(RO_1, RO_2)$, $eval(RO_2, RO_3)$, ..., $eval(RO_{m\text{-}1}, RO_m)$, we know that, e.g., RO_2 is common to the first and the second comparison. As one can see the frequencies 100.91 MHz

of comparison 1 and 100.93 MHz of comparison 2 are very close, while the other frequencies of these comparisons are farther away. Thus, f_{RO_2} can be averaged to 100.92 MHz. If one continues like that, a full model of the analysed RO PUF can be generated by a list as shown in Table 3.

Table 3: Full RO PUF model, RO list

RO_m	$f_{RO_m}[MHz]$
1	100.71
2	100.92
3	101.57
4	101.94
5	100.88
6	101.45
7	100.55
8	100.44
9	101.82

6.6 RO PUF Response Extraction

On the basis of Table 3, an attacker could answer all challenges correctly and also extract the secret PUF response by comparing the chain of oscillators. In our case, this leads to the bitstring $(1, 1, 1, 0, 1, 0, 0, 1)$, representing our RO PUF implementation's response, which we could confirm with the correct responses collected in a log file during measurements.

7. SCA RESISTANT RO PUF

In this section, we discuss methods to avoid side-channel vulnerabilities of RO PUFs and propose a concept of combined approaches to obtain a resistant, but still efficient RO PUF solution.

7.1 No RO Overlapping

As shown above, one crucial point allowing an attacker to map frequencies to oscillators, is the overlapping of ROs in a comparison sequence extracting $m - 1$ bits from m ROs. A straight forward approach to prevent oscillator identification is to use every oscillator only once in a single comparison, resulting in a reduced bit extraction rate of $m/2$ bits from m ROs. Hence, more oscillators have to be implemented for the same number of response bits. This leads to a higher hardware footprint, but at least the overall extraction runtime stays the same.

7.2 Parallel Comparisons

Another fact facilitating EM attacks on RO PUFs is the sequential call of all comparisons, i.e., always two frequencies are present at a time during RO measurement. Therefore, the challenge sequence can be exploited to allocate frequencies to oscillators. Comparing all m oscillators at the same time, would be resistant against the shown side-channel attack and one could still extract $m - 1$ response bits. The significant drawback of this solution is the immense hardware overhead, since every oscillator needs its own counter consisting of several flip-flops and logic gates.

7.3 Combined Concept

The ideas of non-overlapping RO comparisons and parallel RO measurements are inefficient in terms of their required hardware components if they were used alone. In the following, we show how a side-channel attack resistant RO PUF

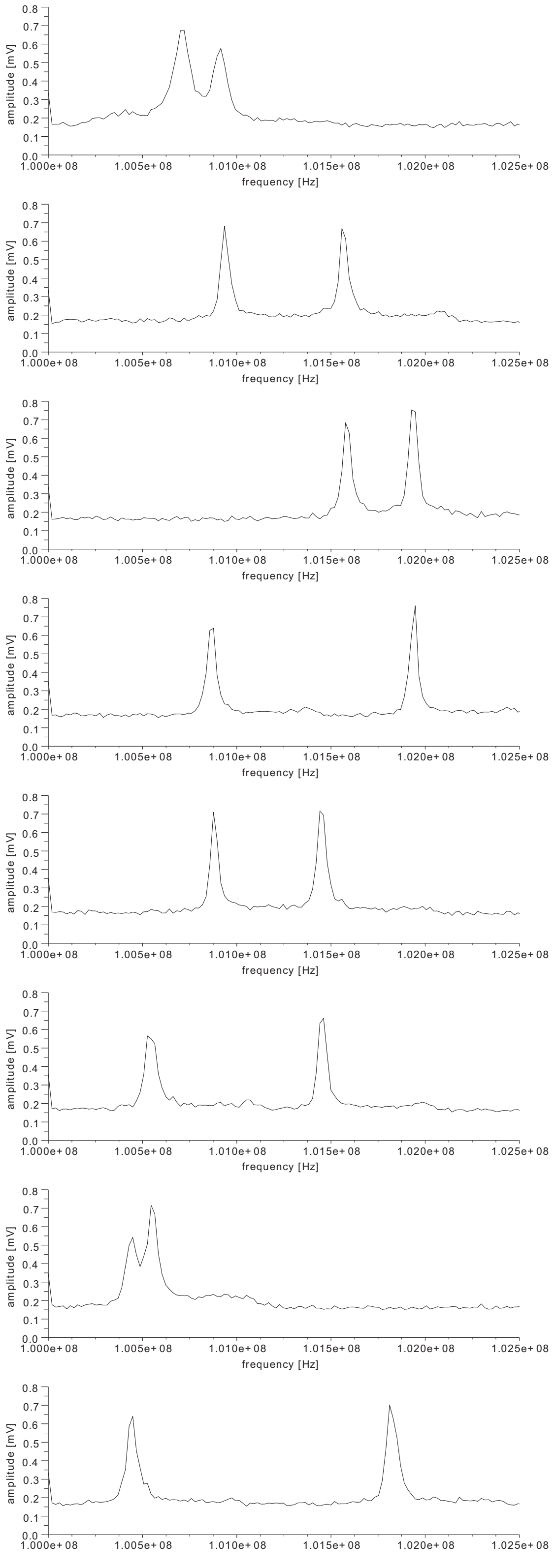


Figure 8: Frequency amplitude spectra for each of the 8 RO comparisons

Table 4: Comparison of parameters for an SCA resistant RO PUF concept

Parallel RO measurement	Number of RO groups	Generated bits	Runtime factor	Required oscillators	Required counters
2	128	128	128	256	2
3	64	128	64	192	3
4	43	129	43	172	4
5	32	128	32	160	5
6	26	130	26	156	6
7	22	132	22	154	7

concept can be achieved, while finding an efficient trade-off between required hardware elements, extractable response bits and overall measurement time.

The idea is to measure a small number m of oscillators in parallel to keep the hardware overhead determined by additional counters low. The measured oscillators are then compared to extract $m-1$ bits. Probably, $m-1$ is smaller then the required amount of bits n, therefore $n/(m-1$ groups of m ROs have to be implemented, while there must not be an overlap between any of these groups. Table 4 shows a list of possible parameters to securely extract 128 bits from an RO PUF. Depending on the required hardware components of an RO and a counter, one could choose the optimal solution for a given device or technology.

7.4 Counter Implementation

Another way of reducing side-channel leakage in RO PUF constructions is to decrease the width of the measuring counters because less registers will generate less electro-magnetic radiation. However, their has to be a reasonable trade-off of counter size and measurement precision.

Further, we evaluated the use of asynchronous ripple counters, where only the first flip-flop is clocked by the RO signal and all others follow asynchronously. We compared the EM side-channel behaviour of RO measurements performed on asynchronous counters with synchronous counters, where every counter flip-flop is clocked by the RO signal. The result was that ripple counters had less emanation, because only one flip-flop had to be driven compared to a full synchronous counter. Therefore, asynchronous counters could be another improvement towards side-channel secure RO PUF implementations, but, in practice, their implementation has to be done carefully to avoid timing problems.

8. CONCLUSION

In this paper, we first demonstrated that decapsulation of a Xilinx Spartan XC3S200 FPGA does not significantly influence the frequencies of implemented ring oscillators, as shown by frequency maps of all implemented ring oscillators, and therefore enables semi-invasive attacks on FPGA implementations of RO PUFs.

Further, we developed analysis methods to determine the RO frequency range and the area of main RO frequency signal leakage from collected EM measurements. We also gave an explanation how to extract a complete list of RO frequencies, i.e. a full model of an RO PUF, from this measurement data, which can be used to compute the correct responses to a given challenge. Based on these methods, our practical results show that RO PUF architectures are vulnerable to side-channel attacks if they use overlapping and sequential RO measurements.

Finally, we presented a concept to make side-channel attacks on RO PUFs infeasible, while focussing on efficient implementation. This makes RO PUFs one of the first PUFs which have been analysed and optimized regarding side-channel leakage.

Acknowledgements

The authors would like to thank Johann Heyszl and Carsten Rolfes for their helpful support. This work was partly funded by the German Federal Ministry of Education and Research in the project *UMABASA* through grant number 01BY1127.

9. REFERENCES

[1] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[2] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Silicon physical random functions. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 148–160, New York, NY, USA, 2002. ACM.

[3] J. Guajardo, S. S. Kumar, G. J. Schrijen, and P. Tuyls. Fpga intrinsic pufs and their use for ip protection. In P. Paillier and I. Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 63–80. Springer, 2007.

[4] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, and P. Tuyls. Extended abstract: The butterfly puf protecting ip on every fpga. *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pages 67–70, June 2008.

[5] R. Maes, P. Tuyls, and I. Verbauwhede. Intrinsic pufs from flip-flops on reconfigurable devices. In *3rd Benelux Workshop on Information and System Security (WISSec 2008)*, page 17, Eindhoven,NL, 2008.

[6] A. Maiti, J. Casarona, L. McHale, and P. Schaumont. A large scal characterization of ro-puf. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 66–71, 2010.

[7] A. Maiti and P. Schaumont. Improving the quality of a physical unclonable function using configurable ring oscillators. In *19th International Conference on Field Programmable Logic and Applications (FPL), 2009.*

FPL '09., 2009.

[8] D. Merli, D. Schuster, F. Stumpf, and G. Sigl. Side-channel analysis of pufs and fuzzy extractors. In *4th International Conference on Trust and Trustworthy Computing (TRUST2011)*, Pittsburgh, PA, USA, June 2011. Springer.

[9] D. Merli, F. Stumpf, and C. Eckert. Improving the quality of ring oscillator pufs on fpgas. In *5th Workshop on Embedded Systems Security (WESS'2010)*, Scottsdale, AZ, USA, October 2010. ACM.

[10] S. Morozov, A. Maiti, and P. Schaumont. A comparative analysis of delay based puf implementations on fpga. Cryptology ePrint Archive, Report 2009/629, 2009. http://eprint.iacr.org/.

[11] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. 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.

[12] L. Sauvage, S. Guilley, and Y. Mathieu. Electromagnetic radiations of fpgas: High spatial resolution cartography and attack on a cryptographic module. *ACM Trans. Reconfigurable Technol. Syst.*, 2:4:1–4:24, March 2009.

[13] E. Simpson and P. Schaumont. Offline hardware/software authentication for reconfigurable platforms. In *Cryptographic Hardware and Embedded Systems - CHES 2006*, pages 311–323. 2006.

[14] S. Skorobogatov. Optical fault masking attacks. In L. Breveglieri, M. Joye, I. Koren, D. Naccache, and I. Verbauwhede, editors, *FDTC*, pages 23–29. IEEE Computer Society, 2010.

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

[16] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters. Read-proof hardware from protective coatings. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, chapter 29, pages 369–383. Springer Berlin Heidelberg, 2006.