A comparison of PUF cores suitable for FPGA devices

Ugo Mureddu, Lilian Bossuet, Viktor Fischer
Univ Lyon, UJM-Saint-Etienne, CNRS, Laboratoire Hubert Curien UMR 5516,
F-42023, SAINT-ETIENNE, France
{ugo.mureddu,lilian.bossuet,fischer}@univ-st-etienne.fr

Abstract

A PUF extracts a unique identifier per die using physical random variation caused by variability of the manufacturing process. PUFs can be used for hardware authentication, but also as generators of confidential keys. This paper presents the comparison of RO-PUF and TERO-PUF cores implemented on Xilinx Spartan 6 FPGA. The objective is to evaluate their design when operating at the same conditions. We show that no ideal PUF exists and therefore designers will always have to choose the PUF matching the security application. In addition to design parameters like area, number of bits per challenge and power consumption, we discuss the feasibility of the design in FPGAs. This will help designers select the best PUF according to their requirements.

1. Introduction

A PUF extracts a unique identifier for similar dies using physical random variation, such as the mismatch between transistors, caused by variability of the manufacturing process (MPV). PUFs can be used for hardware authentication in a challenge-response protocol, but also as generators of device-specific confidential keys. The security of cryptographic systems is mainly linked to the protection of confidential keys. Consequently, if the security system is implemented in a chip, the keys should be generated inside the same chip, i.e. the logic device. Our aim is to select suitable PUFs for FPGA and to fairly evaluate the difficulties related to their implementation, the area and the power consumption. Selected PUFs are Ring Oscillator PUF (RO-PUF) [1] and Transient Effect Ring Oscillator PUF (TERO-PUF) [2]. To compare PUF principles as fairly as possible, the evaluation boards should have the same topology, use as few noisy components as possible, and be powered using low noise power supplies.

The paper is organized as follows. In Section 2, we describe the principle of PUF cores selected. In Section 3, we describe the strategy of implementation and evaluation of the PUF cores. In Section 4, we describe the implementation of PUFs in the FPGA. In Section 5, we discuss the results. Section 6 concludes the paper and describes the future outlook.

2. Selection of PUF cores

We selected PUF cores that use oscillating circuitries as all of them should be feasible in recent and future FPGA families and they use simple sources of randomness.

The first selected PUF principle is the RO-PUF. ROs are digital oscillators consisting of a chain of inverting logic elements connected to form a loop. They are composed of an odd number of inverters to ensure permanent oscillation. RO-PUF extracts MPV in a digital circuit by comparing the oscillation frequencies of two identically implemented ROs.

Behavior of the transient effect ring oscillators (TEROs) corresponds to a very specific configuration of ROs in which the number of inverters is even. Due to MPV, by propagating two events at the same time in the loop, one event will be faster than the other and collision of both events will make the oscillations stop [3]. In this configuration, we compare the number of oscillations of two identically implemented TEROs.

3. Implementation and evaluation strategy of PUF cores

Our objective is to use the same hardware configuration for all PUF cores and for different FPGA families. The system we use has three components: an FPGA device with the target of evaluation (TOE), the acquisition board and the PC running the software. The TOE implemented in FPGA devices is connected to the acquisition card using a simple serial interface. The generated bit streams is saved by the acquisition card and sent to the PC using the USB bus. The device used is a Xilinx Spartan 6 FPGA.

To reduce the vulnerability of the PUF to manipulations, clock signals are generated inside the TOE using a ring oscillator with appropriate topology. One of the parameters use for design evaluation is power consumption. The power consumption of a PUF core is low so we first implemented a reference design in which an input static signal just crossed the device. With this reference project the Spartan 6 consumed 3.5 mW. This power is subtracted from the total power consumption measured in all the experiments. The results presented in the following sections are thus the net power consumption of the designs.
4. Implementation of PUF cores in FPGA

4.1. RO-PUF

The architecture of the RO-PUF implemented in FPGA is composed of two groups of 16 ROs having the following topology: each RO contains one NAND gate and twelve non-inverting buffers. The controller generates the time base signal from the internal clock. Depending on this time base signal, it selects one of sixteen couples of ROs using two multiplexers.

Number of periods of the oscillating signal present at the output of each multiplexer is counted in a synchronous 10-bit counters during each time base period. The arbiter detects which counter reaches its maximum value as the first one. If it is the one of Channel A, the output of the arbiter is set to one, if it is the one of Channel B, the arbiter output is reset to zero.

Since in the first version of the RO-PUF all ring oscillators oscillated permanently, the power consumption was relatively high. However, at any time, just two out of 32 rings are needed. Therefore, in the power optimized version, the unused rings are stopped using demultiplexers.

4.2. TERO-PUF

The TERO-PUF implemented in selected FPGA is composed of two groups of 16 TEROs: each TERO contains one NAND gate and six buffers in both branches. The controller generates the time base signal from the input clock. Depending on this time base signal, it selects one of sixteen couples of TEROs using two multiplexers.

Number of oscillation periods of the two TERO cells selected by multiplexers is counted in two synchronous 10-bit counters during each measurement interval. The counter values are then subtracted to give up to 3-bit response of the PUF.

5. Discussion on FPGA implementation of selected PUF cores

In the following paragraphs, we evaluate and briefly discuss the first results obtained. We use the following PUF characteristics:

- **Area** – the total area is expressed as the number of LUTs and registers occupied by the design.
- **Power consumption** in mW – this parameter gives the power consumption of the PUF core (without data interface).
- **Number of bits per challenge** – gives number of bits obtained at the output of the PUF function as response to the challenge.
- **Power consumption per output bit** – this parameter characterizes the power needed per generation of one output bit.
- **Feasibility and repeatability** – this evaluation parameter reflects the difficulty of the design and its repeatability across different FPGA families.

The implementation results of selected PUF designs are presented in Table 1. It can be observed that the RO-PUF occupies relatively small area, but consumes relatively high power. It is relatively easy to implement (high feasibility and repeatability), but it gives only one bit per challenge. The RO-PUF optimized for power occupies slightly bigger area, but the power consumption is considerably lower. Although the TERO-PUF consumes a little bit more power, it gives more bits per challenge. The power consumption per bit is thus lower but the TERO-PUF requires manual placement and routing for each device individually.

<table>
<thead>
<tr>
<th>PUF type</th>
<th>Area (LUTs + registers)</th>
<th>Power [mW]</th>
<th>Nbits/ch</th>
<th>Power/bit [mW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>RO</td>
<td>214</td>
<td>14.2</td>
<td>1</td>
<td>14.2</td>
</tr>
<tr>
<td>RO Power Opt.</td>
<td>230</td>
<td>2.9</td>
<td>1</td>
<td>2.9</td>
</tr>
<tr>
<td>TERO</td>
<td>382</td>
<td>3.8</td>
<td>3</td>
<td>1.27</td>
</tr>
</tbody>
</table>

Table 1. Implementation results of the PUFs

6. Conclusions

In this paper, we presented and discussed implementation of selected PUF cores in Xilinx Spartan 6 FPGA family. The results confirm that the preselected PUF designs are feasible. However, the TERO-PUF is not suitable for use in practice in his current form since it require some manual intervention. The results also confirm that no ideal PUF exists the most suitable PUF must be selected according to requirements of the security application and some compromise must always be done. Presented results can help designers to make their choice and test the designs on their own hardware. Future work will be to implement those PUFs on other FPGA families.

Acknowledgements

This project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement no. 644052.

References