Design and Implementation of Ring Oscillator Physically Unclonable Function

Shabbir Aglodiya (21BEC006@nirmauni.ac.in) | Harsh Agrawal (21BEC007@nirmauni.ac.in)

Department of Electronics &
Communication Engineering
Institute of Technology, Nirma University
Ahmedabad, India

Abstract
Developing a hybrid ring oscillator based Physical Unclonable Function (PUF) in FPGA is the main work of this paper. Each structure has its own virtuous characteristics thus the desirable characteristics are selected from each PUF then they are combined together to get the hybrid PUF structure. Mainly PUF structures are used as random number generator which should be same at all the generation times. This property makes the circuit suitable for secure cryptographic structure applications. In this paper hybrid ring oscillator structure is proposed for enhancing uniqueness and reliability. The experiments were conducted on Xilinx or Altera DE2 FPGAs with the certain challenging set and produce unique response only for the concerned chip. In the experimental analysis, the proposed design increases the circuit complexity, but the power consumption seems to be same with the traditional designs.

Keywords:
Hardware Security, Uniqueness, Random Number Generation, Intrinsic PUFs, IP Protection, IC Authentication.

INTRODUCTION

In the forthcoming years, the identity number generation and process of creating authentication to the corresponding person becomes a very important task. To increase authenticity and confidentiality, many techniques have been used in the field of cryptography. Still there is a possibility of adversary's actions such as man in the middle attack, malicious code injection or vital code part deletions. To overcome this issue Biometrics is the only solution for creating unique identity generation, but when the biometric chip is replaced or modified, the whole system responds only to the adversary. Nowadays, usage of RFID tags has become the important part for security. In RFID tags, small amount of storage is available for storing information. Within that restricted storage area, feeding large biometric information is not possible. Additionally, adversaries can also clone the chip of RFID in easy manner. The only solution for the problems in RFID tag is to make the unique chip for each person. Physical Unclonable Functions (PUFs) is an emerging cryptographic primitive to produce device fingerprint. A Physical Unclonable Function (PUF), which is a unique challenge-response function, is an emerging hardware primitive for secure applications. PUF make use of manufacturing process variations in a die to generate exclusive signatures out of a chip. This enables chip authentication and cryptographic key generation. PUFs properties are bounded to the underlying hardware. It can be easily evaluated by authorized parties within the device only. The responses are not easily predictable by adversary. It is tamper evident and have a resistance to physical attacks. The chip uniqueness can be implemented by PUF method. Hence an enhanced PUF method has been proposed. Recent works [1][3] on PUF show that it is possible to avoid the malicious attacks and maintain the confidentiality and authenticity within the crypto processor of PUF. The Fig.1 shows the functional description of PUF circuit.

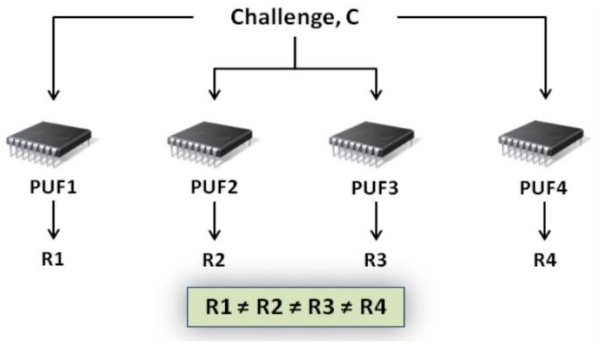

Figure 1. Functional Description of RO PUF

RELATED WORKS

The identity of any chip is defined by PUF technology. PUF is also known as hardware one-way function i.e., the response generated by one chip is not same with other chip with the same input. However, the quality factors

of this PUF, which include uniqueness, reliability and attack resiliency, are negatively affected by environmental noise and systematic variations in the die. This technology can be either built at integrated circuit fabrication level (explicitly introduced randomness) or it can be constructed by some logic circuits (intrinsic randomness). The classification of PUF is shown in Fig.2.

EXPLICITLY INTRODUCED RANDOM PUFS

The production process technology is decided by particular application and availability of materials. PUF production process produce each IC's with a unique characteristic. As per [3] the production process of PUF is classified here. Optical PUF's contain a transparent substrate with light scattering particles. When the light scattering particles were irradiated with the laser beam, PUF's can create the unique speckle pattern. The uncontrolled placement of the scattering particles and interaction between the laser and the particles is unpredictable. Memristor is considered as a fourth element in passive components which exhibits different properties under different doping condition, in turn, it can produce different electrical characteristics, through that different random functions can be created. The concentration of the doping material (titanium oxide (Tio2)) for each of the IC must be different; which affects the resistance of the device, which can be characterized by,

$$R_{eq}(t) = \frac{w(t)}{D} R_{on} + \left(1 - \frac{w(t)}{D}\right).R_{off}$$

where, Ron is the least resistance value when more dopants are added or the whole device is doped. Similarly, Roff is the resistance when only the small area is doped or the entire area is undoped. The width of the doped region is mentioned as w, whereas the total width is D. When the width variable w(t) = D, the equivalent resistance is just Ron. Similarly, when w(t) = 0, then the equivalent resistance is Roff. The varying nature of resistance can be utilized for creating PUFs as per [5].

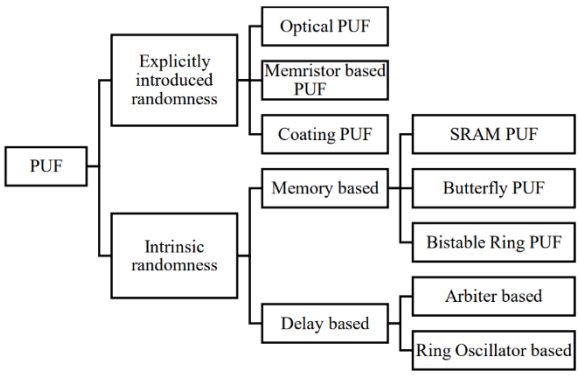

Figure 2. PUF Classification

For coating PUF a dielectric material with different random permittivity (a mixture of Tio2 and Tin) is spreaded over the top layer of the normal IC which acts as PUF [6]. An array of sensors arranged in comb like structure reveals different capacitance values. Hence the random number generation depends upon the location of dielectric particles and charging and discharging properties of capacitors.

INTRINSIC RANDOMNESS PUFS

Explicitly introduced random PUFs should be constructed during the manufacturing process, but intrinsic randomness PUFs can be included in any FPGA design module. Intrinsic randomness PUF circuits contain an array of circuits either as a form of memory elements or some set of delay elements. Hence, they are classified as memory based PUF and delay based PUF in intrinsic randomness PUFs.

Memory Based PUF:

The SRAM PUF works on the principle of startup behaviour [2] [4] of SRAM cells. The threshold voltage mismatch (even in the same manufacturing process) of the MOS transistors creates two stable states either as '0' or '1'. It can be used directly as a key to protect sensitive data. The idea described in [2] classify the memory cells of the standard SRAMs into two disjoint events, one suitable for generating identifiers or secret keys, and the other suitable for the generation of random numbers. This classification method under different operating condition proves that misclassification of cells due to aging problems may be reduced. Kumar in [3] used Butterfly based PUF Internet Protocols for handling sessions which is feasible to implement in FPGA. Butterfly PUF (BPUF) contains two cross coupled D flip flops which act as latches. The challenge or excitation input is given as clear input for one latch, preset input for next latch. The final output is determined by the random delay

variations in the challenge signal path. In eight stages Bistable Ring Oscillator PUF the possible response may be “10101010” or “01010101” based on the input challenge lines. If the reset input is ‘1’ means all the outputs will be ‘0’ from each BR cell. If reset input is ‘0’ then each it works as inverter. This makes the behaviour of BRO PUF is easy for machine learning attacks and cryptanalysis. To overcome this problem, challenge lines are applied to the certain number of BR cells simultaneously as suggested by [1]. The results from all the cells are XORed together to get a single response which will prevent brute force attack.

Delay Based PUF:

The idea behind the delay based PUFs are that two circuit path delays try to win themselves to reach the destination path first. If the first path reaches first means the response is ‘1’, otherwise ‘0’. An Arbiter PUF [4] (APUF) is composed of two identical configured delay paths that are stimulated by a clock signal. The multiplexer select lines are challenge line inputs. Each challenge line has to select two 2:1 multiplexer output. The edge triggered flip flop latch is used for measuring the delay difference between the two propagation paths. This difference can’t be predictable due to manufacturing process variation present in the multiplexer (two AND gates and one OR gate) and latch. In Ring Oscillator PUF (RO-PUF), variations in the frequencies of identically constructed ring oscillators are utilized to build the PUF. The RO frequencies are monitored and transmitted using multiplexer. Number of inverters in each ring oscillator can be three or five or any odd numbers. In a frequency counter, the frequencies are counted and the frequency deviation in the set of ring oscillators is transformed into binary outputs by a simple comparison method. As shown in Fig.3, challenge lines act as the multiplexer select lines. Common enable signal is given to all RO structures. The multiplexer select lines are challenge lines which are going to select any of the rows from upper and lower ring oscillator set for comparison. Due to manufacturing variations each ring oscillator frequency will be varied and this variation is recorded by frequency counters and comparators. A large amount of area is occupied by two frequency counters and comparator circuit as by the circuit suggested in [3]. Hence they can be replaced by a frequency divider network as described in [5] where the modulus value is considered as response. A ring oscillator (RO) based PUF is a promising solution for FPGA platforms. In this paper typical ring oscillator structure form is modified. Hence the previous works on the ring oscillator structure is focused. The single chip secure processor introduced by Sue and Devedoss known as AEGIS secure processor [1] [6] in which the encryption key from PUF secret security kernel and main processor is implemented on the same chip. Consider a situation where ID is generated in separate module and transferred to the main module for verification and further calculation where they are communicated by cable, it can be tracked by attacker. The hackers can stole the data using cables without the knowledge of customer. Hence to avoid this cryptographic module and main processor module are built with the same module. With this AEGIS processor, PUF plays the main role of creating the random number. When the ring oscillators chosen for comparison are varying only by small amount then the performance of them will be decreased with respect to temperature. When 1 out of k-masking scheme is used, the particular set of the ring oscillator pair will be selected in upper and lower paths whose frequency variation is high. With this masking scheme the performance of PUF won’t be degraded with respect to temperature. But these masking circuits will occupy more area and circuits. In addition to that the fixed RO-pair frequencies can be modelled and it is vulnerable to the attacker.

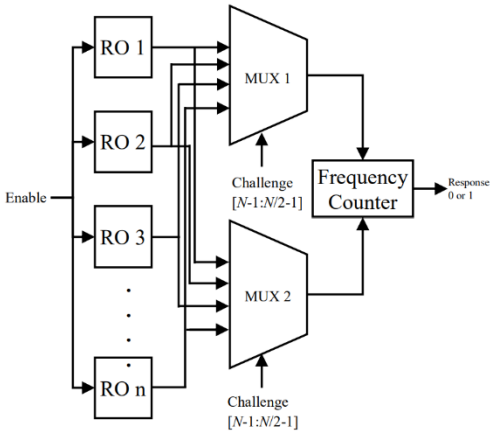

Figure 3. Typical Ring Oscillator

This problem is overcome by Maiti's [10] reconfigurable PUF where each inverter block contains two NOT gates and 2:1 multiplexer where challenge bits act as select lines. For a ring oscillator with three stages (each stage two inverters) 8 different configurations are possible. Minimum delay is noted for '000' challenge line and '111' challenge line exhibits maximum delay path. The various paths are selected only by challenge lines. Xin et al. [5] expands the idea of Maiti by inserting additional multiplexer in the ring oscillator package and by splitting the challenge lines for selecting inverters in the ring oscillator packages and multiplexer selection. Xin increases the number of possible configurations to 256. In all the ways the RO PUF works on the process of comparing frequencies. For each RO PUF one output bit is produced. For n output bits n RO PUF packages should be constructed. This PUF matrix value should be unique for every chip. Further it can't be traced by any of the mathematical modelling attack. In this paper ring oscillator based PUF concept with modification known as challenge line delay logic is proposed and discussed in section 5. In addition to that the delay in ring oscillator is increased by a basic logic of bistable ring oscillator PUF. In section 6 the results are compared with typical Ring Oscillator based PUF.

Ring Oscillator

Figure 4 shows a schematic of the Ring oscillator formed using one NAND Gate and three inverters. RO will generate a square waveform. The frequency of the square wave depends on the net delays and propagation delay of each gate.

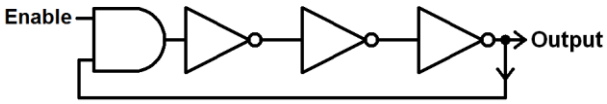

Figure 4. Ring Oscillator

Figure 6 shows the basic block diagram of the 1-bit RO PUF. Ring oscillator PUF design comprises 32 ROs. The first sixteen ROS connected to one of the 16:1 multiplexer another 16 ROs connected to another 16:1 multiplexer. Select lines of multiplexers will decide which RO to be selected. Both multiplexers will use select lines, i.e., 'challenge' input, to choose any two ROs at the same time. Multiplexers connected clock input to two distinct 12-bit counters via the output of multiplexers. The counting will be carried out on every affirmative edge of ROs. The comparator compares the counter output. As previously said, each RO generates frequencies with distinct periods. Even if both counters are linked to the multiplexers, counting may start at the respective incoming positive edge. It causes any one of the counters may get overflows first. If the upper counter gets overflowed first than the lower counter then comparator output will be '1' otherwise '0'. The output of the comparator is a 'response' bit. ROs on two different devices have different frequencies. This difference allows the RO PUF to characterize devices to authenticate them [1, 16]. This is an implementation of the RO-PUF in 8 bits. the time simulation of a single bit RO PUF after it has been implemented. The inputs are Enable and Challenges. The output of each module can be seen here. ROs will produce square waves, as previously indicated. The outputs of Ring Oscillators are frw and frwm. Mux out1 and mux out2 are the mux outputs. The mux outputs are mux out1 and mux out2, and two multiplexers can simultaneously select any two ROs. Two counters use the clock input from these mux outputs as well. Every positive edge of the incoming clock will be counted as well. The outputs of the two counters are referred to as Ct1 and Ct2. Because no counters are overflowing, the response bit remains '0'. If Ct1 reaches its maximum value before Ct2, the response bit will be set to '1'. In this Fig. 9, we can observe the 'response' bit. As soon as the first counter gets overflows response bit became '1'.

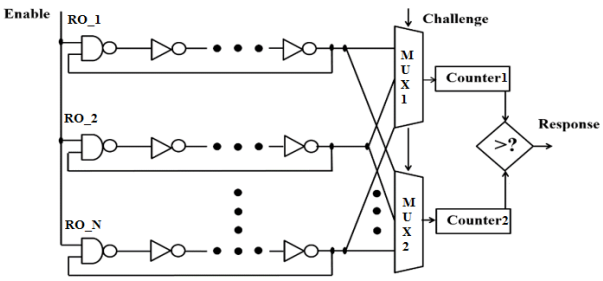

Figure 5. Ring Oscillator PUF

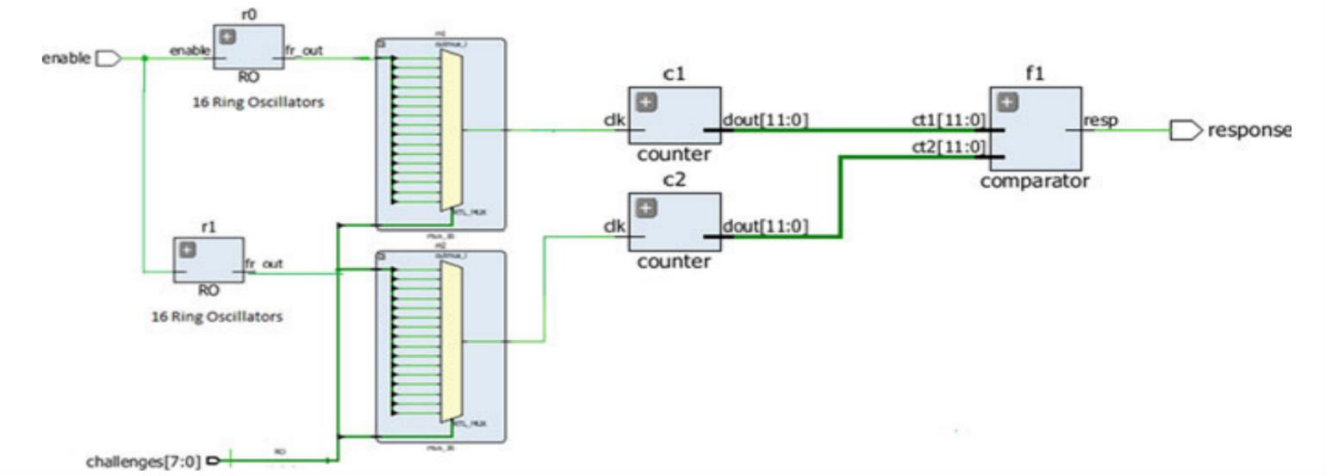

Figure 6. One Bit RO PUF Schematic

EXPERIMENT AND RESULTS

We have designed the RO PUFF using Verilog Hardware Description Language in Quartus II and ModelSim Software and obtained its RTL View (Fig. 7) & Technology Map View and also observed the simulation using timing waveform analysis.

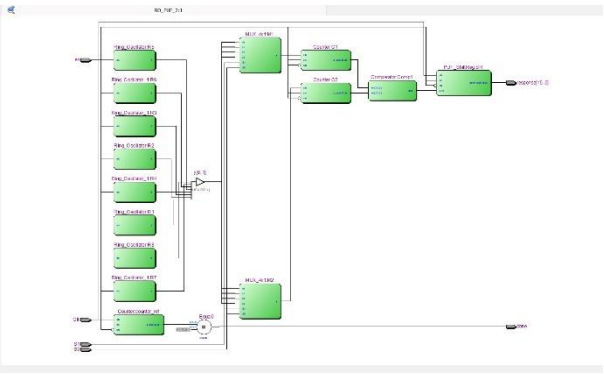

Figure 7. Ring Oscillator PUF RTL View

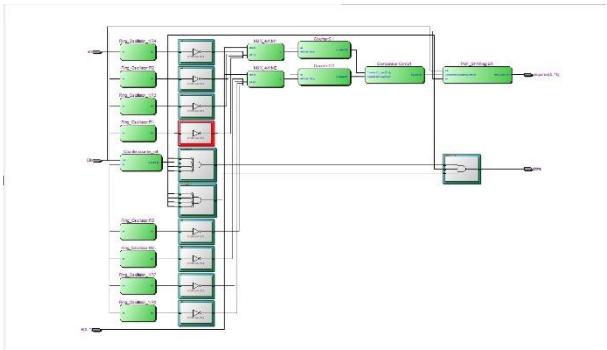

Figure 8. Ring Oscillator PUF Technology Map View

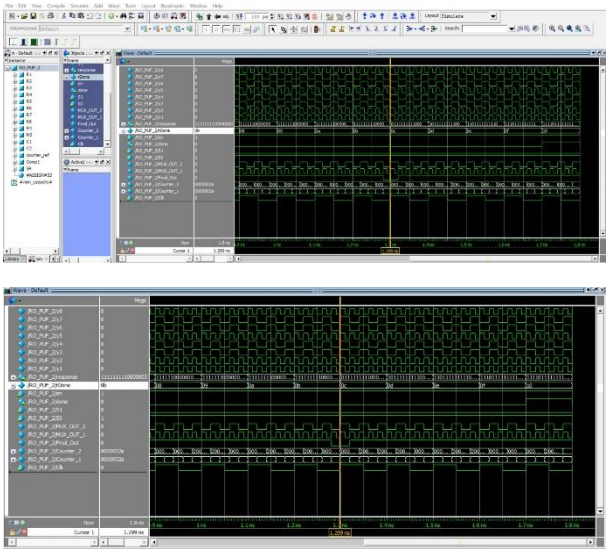

Figure 9. Simulated Waveform

VERILOG CODE

```
module RO_PUF_2(en,S1,S0,Clk,done,response);
input en,Clk;
input S1,S0;
output done ;
output [15:0] response;
//output Final_Out;
(*keep=2*)wire y1,y2,y3,y4,y5,y6,y7,y8;
wire MUX_OUT_1,MUX_OUT_2;
wire [31:0]Counter_1,Counter_2;
wire [8:0] rDone;
wire Final_Out;
Ring_Oscillator R1 (en,y1);
Ring_Oscillator R2(en,y2);
Ring_Oscillator_1 R3(en,y3);
Ring_Oscillator_1 R4(en,y4);
Ring_Oscillator R5(en,y5);
Ring_Oscillator_1 R6(en,y6);
Ring_Oscillator_1 R7(en,y7);
Ring_Oscillator R8(en,y8);
MUX_4x1 M1(y1,y2,y3,y4,S1,S0,MUX_OUT_1);
MUX_4x1 M2(y5,y6,y7,y8,S1,S0,MUX_OUT_2);
Counter C1(.clk(MUX_OUT_1),.en(en),.rst(~en),.c_out(Counter_1));
Counter C2(.clk(MUX_OUT_2),.en(en),.rst(~en),.c_out(Counter_2));
Counter counter_ref(.clk(Clk), .en(en),.rst(~en), .c_out(rDone));
Comparator Comp1(Counter_1,Counter_2,Final_Out);
PUF_ShiftReg SR(.clk(Clk), .en(en),.rst(~en),.s_in(Final_Out), .shift_out(response));

  assign done = (rDone == 9'd256) ? 1'b1 : 1'b0;
```

```
endmodule

module Ring_Oscillator(en,y);
input en;
output y;
(*keep=1*)wire w1,w2,w3,w4,w5;
and(w1,en,y);
not(w2,w1);
not(w3,w2);
not(w4,w3);
not(w5,w4);
not(y,w5);
endmodule

module Ring_Oscillator_1(en,y);
input en;
output y;
(*keep=1*)wire w1,w2,w3,w4,w5;
and(w1,en,y);
not(w2,w1);
not(w3,w2);
not(w4,w3);
not(w5,w4);
not(y,w5);
endmodule
module MUX_4x1(I0,I1,I2,I3,S1,S0,y);
input I0,I1,I2,I3;
input S1,S0;
output y;
assign y= ~S1 & ~S0 & I0 | ~S1 & S0 & I1 | S1 & ~S0 & I2 | S1 & S0 & I3 ;
endmodule

module Counter(

input clk, en, rst ,
output reg [31:0] c_out
);

always@(posedge clk or posedge rst)
begin
if(rst) c_out <= 32'd0;
else
begin
if(en)
c_out <= c_out + 32'h1;
else
c_out <= c_out;
end
end
endmodule


module Comparator(in1,in2,out);
input [31:0] in1,in2;
output reg out;
always @ (*)
begin
if(in1==in2)
out = 1'b0;
else
out = 1'b1;
end
endmodule

module PUF_ShiftReg(

input clk, s_in, en, rst,
output reg [15:0] shift_out

);

always@(posedge clk or posedge rst)
begin
if(rst) shift_out <= 16'd0;
else
begin
if(en) shift_out <= {s_in, shift_out[15:1]};
else shift_out <= shift_out;
end
end
endmodule
```

CONCLUSION

Hardware-oriented security is an upcoming field in the electronics industry. Today hardware designers are paying more focus on hardware security. Many different PUF designs are getting designed by the researchers. In the field of cryptography the hardware module PUF structure introduces a new era. Works on PUF structures are happening from the last decade, the proposed hybrid PUF structure increases the reliability and uniqueness. This takes advantage of creating MUX structure for selecting challenge lines. There are some limitations on PUF performance due to temperature variations, aging and variation of electron density in different environment conditions which induces the PUF research orientations towards that. The random number generated with this PUF structure can be utilized to implement any of the private or public key cryptosystem by designing the protocols as per the requirements.

REFERENCES

1.Kumar, M.A., Bhakthavatchalu, R.: FPGA based delay PUF implementation for security applications. In: IEEE International Conference on Technological Advancements in Power and Energy (TAP Energy). Kollam, India (2017).

2.Yin, C., Qu, G., Zhou, Q.: Design and implementation of a Group-based RO PUF. In: Automation & Test in Europe Conference & Exhibition. Grenoble, France (2013)

3. Lim, D., Lee, J.W., Gassend, B., Suh, G.E., van Dijk, M., Devadas, S.: Extracting secret keys from integrated circuits. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 13, 1200–1205 (2005)

4. Bernard, F., Fischer, V., Costea, C., Fouquet, R.: Implementation of ring-oscillators-based physical unclonable functions with independent bits in the response. Int. J. Reconfig. Comput, 2012, 13 (2012)

5. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: 44th ACM/IEEE Design Automation Conference. San Diego, CA (2007)

6. Handschuh, H., Schrijen, G.-J., Tuyls, P.: Hardware intrinsic security from physically unclonable functions. In: Sadeghi, A.-R., Naccache, D. (eds.) Towards Hardware-Intrinsic Security, Information Security and Cryptography, pp. 39–53. Springer, Berlin Heidelberg (2011). https:// doi.org/10.1007/978-3-642-14452-3_2