TERO-based Detection of Hardware Trojans on FPGA Implementation of the AES Algorithm

Paris Kitsos
Digital IC dEsign and Systems Laboratory (DICES Lab)
Computer and Informatics Engineering Department
TEI of Western Greece, Greece
e-mail: pkitsos@teimes.gr

Kyriakos Stefanidis
Industrial Systems Institute
"Athena" Research and Innovation Center in ICT and Knowledge Technologies
Platani, Patras, Greece
e-mail: stefanidis@isi.gr

Artemios G. Voyiatzis
SBA Research
Vienna, Austria

***Abstract*—A Transient Effect Ring Oscillator (TERO) is a special case of a Ring Oscillator (RO) design that exhibits increased sensitivity to intrinsic noise. It can serve as a basis for implementing a True Random Number Generator (TRNG) or a Physically Uncloneable Function (PUF). Also, as a digital sensor for detecting insertion of malicious hardware logic (Trojans) in digital circuits. Here, we explore the application of TERO for detecting hardware Trojans injected in FPGA implementations of the AES cryptographic algorithm. Experiments and comparisons are reported in terms of the frequency as a function of the TERO length. Our findings indicate that TERO-based digital sensors can be used to efficiently detect the presence of the Trojan.**

Keywords— FPGA security; time analysis; ring oscillators; Transition Effect Ring Oscillator; hardware Trojan horse.

I. INTRODUCTION

It is very hard, nowadays, to guarantee that a hardware design is not infected by some malicious logic as a consequence of the globalization of integrated circuit (IC) manufacturing and the distributed supply chains involved in the design of a system.

The malicious logic inserted in the design is often mentioned as "hardware Trojan horse" or simply a "Trojan". The malicious logic may perform different actions, depending on the motivations of its designer, ranging from circuit destruction to control take over and leakage of sensitive information through the primary outputs or side channels [1-2]. Hence, the need to integrate appropriate defenses in the IC design so as to detect the presence of malicious logic and defend against their potential activity, even if not detected in first place. Integrated mechanisms for Trojan detection are very useful given that exhaustive testing of all possible circuit states under all environmental conditions is not feasible in most of the cases.

A Ring Oscillator (RO) is a closed loop chain of an odd number of inverters. An example RO of length five is depicted in Fig. 1. An RO oscillates at a fixed frequency depending on the exact components, the size of a circuit, the operating characteristics (e.g., voltage) and the environmental conditions (e.g., temperature). Even minimal modifications of the circuit can result in a frequency change, rendering it very sensitive to process variations [3-4]. Many research works already proposed the integration of an RO close to sensitive parts of the circuit so as to detect Trojans [5-8].

A Transient (or Transition) Effect Ring Oscillator (TERO) is, in principle, a more sensitive variant of a ring oscillator. Previous works proposed the use of TEROs for implementing True Random Number Generators (TRNGs) and Physically Uncloneable Functions (PUFs) [9-10].

We introduced the use of TERO as a digital sensor for Trojan detection and studied its applicability in the case of simple Trojans against the cryptographic algorithms SNOW3G and Mosquito in [14, 15]. In this paper, we perform a comparison on the timing sensitivity of TERO against RO, towards introducing TERO as an alternative means for detecting Trojans implanted in FPGAs.

The rest of the paper is organized as follows. In Section II, we discuss RO and TERO implementations in FPGAs. In Section III, we describe the experimental setup and in Section IV, we analyze the results of our experiments. Finally, Section V provides the conclusions of this paper.

II. BACKGROUND

A ring oscillator with a sequence of five inverters in a closed loop (i.e., an RO with length 5) is depicted in Fig. 1. A counter fetches the output of the RO in order to measure the oscillation frequency.

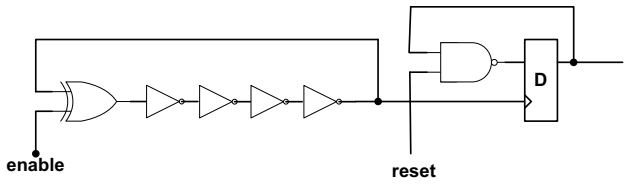

Fig. 1. Ring oscillator in FPGA

A Transient Effect Ring Oscillator (TERO) is composed of an SR flip-flop implemented with two XOR gates and two AND gates [9]. This architecture has two control signals, for start and reset. The correct place-and-routing for a TERO is important so as to ensure the same length of the interconnections between the XOR gates.

978-1-5090-2817-7/16 $31.00 © 2016 IEEE
DOI 10.1109/DSD.2016.47

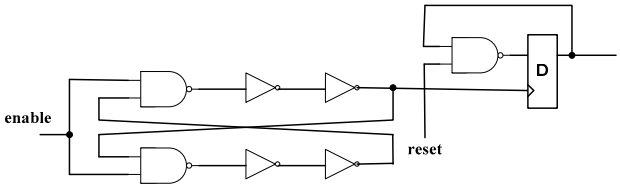

Fig. 2. TERO circuit with one control signal

Here, we use a simpler TERO architecture, where the XOR and AND gates are merged into NAND gates with some inverters in the feedback loop, as depicted in Fig. 2. The advantage of this approach is that only one control signal is used either for resetting or oscillating the TERO circuit.

The reset occurs when the control signal, *enable*, is set to '0' and drives the loop to the same initial conditions before generating its output. When the control signal switches from '0' to '1', the TERO circuit starts to oscillate. An asynchronous counter is used to measure the TERO frequency.

III. Experimental Setup

In order to investigate the effectiveness of the TERO, we realized a hardware Trojan horse against an FPGA-based implementation of the well-known AES cryptographic algorithm. We chose for our experiments the AES implementation that is provided on a Spartan 6 (XC6SLX75-2CSG484C) FPGA part of the SAKURA-G board [16]. The hardware architecture is discussed in detail in [17].

We designed a combinational Trojan, as depicted in Fig. 3. The Trojan trigger part comprises a tree of AND gates that monitors the values of a randomly selected subset of 8 out of the 128 key bits (namely, positions 3, 20, 41, 62, 75, 90, 100, and 119).

The Trojan payload part consists of a XOR gate that drives the *enc_dec* signal for defining the mode of operation (encryption or decryption) for the AES. Once the malicious logic is triggered (when the key bits match the pattern for Trojan), it inverses the selected mode of operation. This effectively creates a denial-of-service attack.

All the designs were captured in VHDL and were synthesized in the same FPGA board. The AES design covers a region defined by slices X0Y38 and X32Y87 in the FPGA. This region was further split in nine smaller sub-regions, as depicted in Fig. 4. A TERO or RO sensor is used in each sub-region in order to detect the presence of a Trojan. The exact placement of the ROs or TEROs are summarized in Table 1.

In order to achieve accurate and comparable measurements, we built designs that share the same place-and-route. We succeed in this by following the steps outlined in [15]. This procedure produces an AES design without the Trojan and includes the RO or TERO and an AES design with the Trojan and RO or TERO. We experimented with four different RO/TERO lengths, varying between 15 and 27 with a step equal to 4.

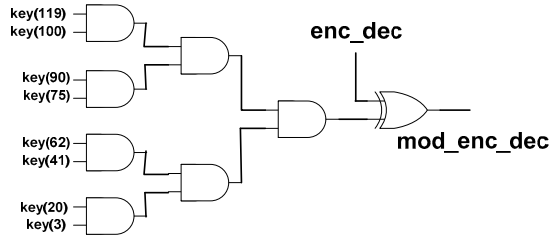

Fig. 3. Trojan design

The Trojan occupied a very small percentage of the available FPGA area compared to that needed for the AES implementation. Each slice in Spartan 6 comprises four 6-input Look-Up-Tables (LUTs) and eight flip-flops (FFs). The implementation of the RO sensor consumes 4-7 slices, while the one for a TERO sensor consumes 8-14 slices.

Fig. 5 depicts the area covered by the 27-bit-long RO sensor and the 19-bit-long TERO one respectively as a white trace. The same layout for the circuits with AES and TERO/RO is achieved and the hardware resources are placed and routed on the same FPGA locations.

The left part of Fig. 6 depicts the layout of the AES design with the 19-bit-long TERO (green rectangle in sub-region 2) and the Trojan (blue rectangle next to it, in sub-region 5). The right part of Fig. 6 depicts the layout of the AES design with the 27-bit-long RO (green rectangle in sub-region 9) and the Trojan (again, blue rectangle in sub-region 2).

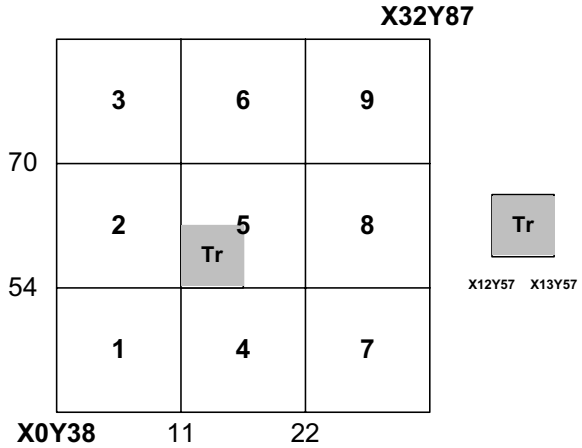

Fig. 4. RO/TERO counter value capturing

RO positions	TERO positions
3: X9Y66	3: X9Y66-X10Y66
6: X19Y66	6: X19Y66-X20Y66
9: X28Y66	9: X28Y66-X29Y66
2: X9Y57	2: X9Y57-X10Y57
5: X19Y57	5: X19Y57-X20Y57
8: X28Y57	8: X28Y57 - X29Y57
1: X9Y46	1: X9Y46-X10Y46
4: X19Y46	4: X19Y46-X20Y46
7: X28Y46	7: X28Y46 - X29Y46

Table 1. Sub-region positions of RO and TERO sensors

Fig. 5. Area covered by 27-bit RO (left) and 19-bit TERO (right)

The experimental setup comprises a personal computer interfacing the SAKURA-G board, the AES algorithm in ECB mode, the RO and TERO components with their respective counters, and the aforementioned Trojan design.

We experimented with the design using 500 randomly-generated test vectors and derived the oscillation counts for the RO- and TERO-enhanced designs. We also studied the effect of different lengths for the RO and the TERO, as proposed in [12].

In order to capture the value of the RO/TERO counters, the design implementation is instrumented with the Xilinx Chipscope (a Virtual I/O -VIO and an ICON) from the computer [13]. The counters are implemented in a DSP48A1 slice and have a width of 18 bits.

IV. Results

The oscillation counts collected in our experiments for each sub-region are summarized in Table 2 (for TERO) and Table 3 (for RO). Each experiment was repeated five times to mitigate any measurement error by averaging the counts.

The distance (difference) between the oscillation counts of the Trojan-free and the infected circuits is the best metric regarding the reliability and the detection sensitivity.

The TERO sensor is more sensitive (increased counts) when the length is smaller. The closer the TERO sensor to the Trojan, the biggest the count difference when a Trojan is inserted. The sensitivity decreases consistently as the length increases for all sub-regions. For lengths greater than 19, no sub-region sensor is able to detect any difference with the Trojan-free design.

The RO sensor exhibits a similar behavior with TERO in count differences as a function of the length. However, we noticed that the RO sensor counts are totally unstable from run to run in our experiments. Hence, they cannot be used to reliably detect the presence of the Trojan, as they produce many false positives. Also, due to this instability, it is not possible to directly compare the sensitivity of the TERO sensor against the RO one and, thus, confirm the theoretical result that a TERO would oscillate at about double the frequency of an RO of equal length [9].

V. Conclusions

A TERO-based design can serve as a sensor for detecting malicious hardware logic inserted in a design. We explored the case of protecting the AES cryptographic algorithm implemented on an FGPA board and in comparison with a RO sensor. The latter proved quite unstable; the reasons for this instability should be explored in the future work. The TERO sensor performed reliably in all cases. It was possible to observe significant differences in the oscillation counts when a tiny combinational Trojan was inserted in the design. These differences were more evident when sensors with small lengths were used. Thus, TERO-based sensors can be used as an efficient defense mechanism for FPGA designs.

As future work, we plan to further investigate the performance of TERO sensors of different lengths, as a means to increase their sensitivity and also study the effects of using multiple sensors of different lengths (in one sub-region or spread in different ones). Also, to confirm our findings with different samples of Trojans and on different FPGA boards.

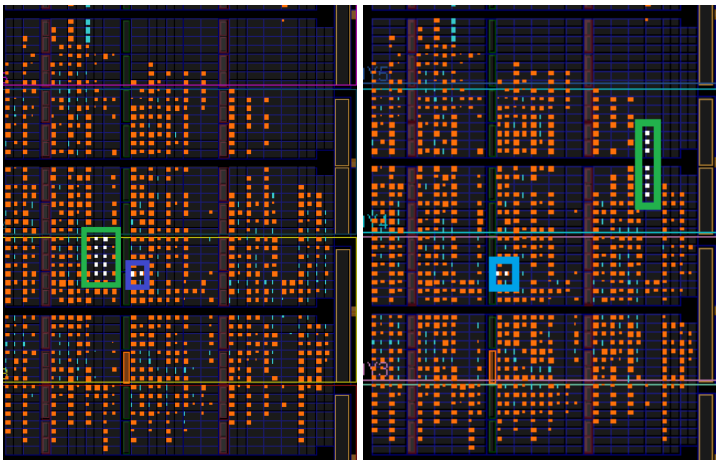

Fig. 6. Implementation layout of the AES block cipher, the TERO sensor (left, in green) and the RO sensor (right, in green). The Trojan design is in blue.

Table 2: Oscillation counts (values in hexadecimal) with TERO

Infected with Trojan									
Length	**Region 1**	**Region 2**	**Region 3**	**Region 4**	**Region 5**	**Region 6**	**Region 7**	**Region 8**	**Region 9**
15	5208	5014	3478	5014	4F3D	5014	4A38	4C4B	5014
19	30D4	3FDC	3E80	3E80	3E80	3E80	3E80	3E80	3E80
23	34BC	34BC	34BC	34BC	34BC	34BC	34BC	34BC	34BC
27	2CEC	2CEC	2CEC	2CEC	2CEC	2CE8	2CEC	2CEC	2CEC
Trojan-free									
Length	**Region 1**	**Region 2**	**Region 3**	**Region 4**	**Region 5**	**Region 6**	**Region 7**	**Region 8**	**Region 9**
15	4886	4E20	3A38	5014	4FFC	5014	4844	4E20	4E20
19	38D2	400E	3E80	4074	3E80	3E80	3E80	3E80	3E80
23	34BC	34BC	34BC	34BC	34BC	34BC	34BC	34BC	34BC
27	2CEC	2CEC	2CEC	2CEC	2CEC	2CEC	2134	2134	2CEC

Table 3: Oscillation counts (values in hexadecimal) with RO

Infected with Trojan									
Length	**Region 1**	**Region 2**	**Region 3**	**Region 4**	**Region 5**	**Region 6**	**Region 7**	**Region 8**	**Region 9**
15	EF59	5468	874C	473D	2526	D40D	34CD	4858	513B
19	459E	53E2	F4FB	3258	245C	3BB3	3709	2DF4	23D0
23	3FD3	4303	3186	3BBF	16D0	268A	2EF0	208F	2939
27	336A	3E24	31C4	203E	3690	2E8D	2296	2BF8	2B03
Trojan-free									
Length	**Region 1**	**Region 2**	**Region 3**	**Region 4**	**Region 5**	**Region 6**	**Region 7**	**Region 8**	**Region 9**
15	DD49	4107	8B5F	4661	2532	DDC4	38DD	4C56	5140
19	3982	2144	C567	25D7	2471	445E	39E1	2A09	2535
23	3BE9	2839	1E48	32FE	1DA4	26FB	234D	2050	2F4E
27	3905	2AA0	2EDC	20F3	163D	2837	3BA9	232F	2973

ACKNOWLEDGEMENT

This work was partially supported by the COMET K1 program of the Austrian Research Promotion Agency (FFG).

REFERENCES

[1] S. Mal-Sarkar, A. Krishna, A. Ghosh, and S. Bhunia, "Hardware Trojan attacks in FPGA devices: Threat analysis and effective countermeasures", in Proc. of the 24th edition of the Great Lakes Symposium on VLSI (GLSVLSI 2014), Pittsburgh, Pennsylvania, USA, 2014, pp. 287-292.

[2] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware Trojans", IEEE Computer 43(10), October 2010.

[3] G.E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in Proc. of the 44th ACM/IEEE Design Automation Conference (DAC '07), San Diego, USA, 2007, pp. 9–14.

[4] J. Guajardo, S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in Cryptographic Hardware and Embedded Systems (CHES 2007), Vienna, Austria, 2007, pp. 63–80.

[5] X. Zhang, A. Ferraiulo and M. Tehranipoor, "Detection of Trojans using a combined ring oscillator network and off-chip transient power analysis", ACM Journal on Emerging Technologies in Computing Systems (JETC), 9(3):25, September 2013.

[6] J. Rilling, D. Graziano, J. Hitchcock, T. Meyer, X. Wang, P. Jones, and J. Zambreno, "Circumventing a ring oscillator approach to FPGA-based hardware Trojan detection", in 2011 IEEE 29th International Conference on Computer Design (ICCD), 2011, pp.289-292.

[7] A. Ferraiuolo, X. Zhang, and M. Tehranipoor, "Experimental analysis of a ring oscillator network for hardware Trojan detection in a 90nm ASIC", in 2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2012, pp. 37-42.

[8] J. Rajendran, V. Jyothi, O. Sinanoglu, R. Karri, "Design and analysis of ring oscillator based Design-for-Trust technique", in 2011 IEEE 29th VLSI Test Symposium (VTS), 2011, pp. 105-110.

[9] M. Varchola and M. Drutarovsky, "New high entropy element for FPGA based true random number generator", in Cryptographic Hardware and Embedded Systems (CHES 2010), Santa Barbara, USA, 2010, pp. 351-365.

[10] L. Bossuet, X, Thuy Ngo, Z. Cherif, and V. Fischer, "A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon", IEEE Transactions on Emerging Topics in Computing, 2(1):30-36, March 2014.

[11] A. Maiti, J. Casarona, L. Mchale, and P. Schaumont, "A large scale characterization of RO-PUF", IEEE International Symposium on Hardware Oriented Security and Trust (HOST 2010), pp. 94-99.

[12] P. Kitsos and A.G. Voyiatzis, "FPGA Trojan detection using length-optimized ring oscillators", in Proc. 17th Euromicro Conference on Digital Systems (DSD 2014), Verona, Italy, 27-29 August, 2014.

[13] PlanAhead Design and Analysis Tool, available at: http://www.xilinx.com/tools/planahead.htm

[14] P. Kitsos and A.G. Voyiatzis. Investigating TERO for hardware Trojan horse detection. In TRUDEVICE 2015: Workshop on Trustworthy Manufacturing and Utilization of Secure Devices, 2015. DATE 2015 Friday Workshops, Grenoble, France, March 9-13, 2015.

[15] P. Kitsos and A.G. Voyiatzis. A comparison of TERO and RO timing sensitivity for hardware Trojan detection applications. In 18th EUROMICRO Conference on Digital System Design (DSD 2015), pp. 547-550. IEEE CPS, 2015. Madeira, Portugal, August 26–28, 2015.

[16] http://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-G.html

[17] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," in Advances in Cryptology ASIACRYPT 2001. Springer, 2001, pp. 239–254.