The ISC Int'l Journal of Information Security

July 2021, Volume 13, Number 2 (pp. 163–176)

http://www.isecure-journal.org

Intensive Analysis of Physical Parameters of Power Sensors for Remote Side-Channel Attacks **

Milad Salimian [1,*] and Ali Jahanian [1]

[1] *Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran.*

ARTICLE INFO.

Article history:
Received: December 18, 2020
Revised: March 21, 2021
Accepted: May 26, 2021
Published Online: June 26, 2021

Keywords:
CPA, FPGA, Side-Channel, Power Sensor, TDL, TDC

Type: Research Article

doi: 10.22042/isecure.2021.262549.591

dor: 20.1001.1.20082045.2021.13.2.7.2

Abstract

Side-channel analysis methods can reveal the secret information of digital electronic systems by analyzing the dependency between the power consumption of implemented cryptographic algorithms and the secret data. Recent studies show that it is possible to gather information about power consumption from FPGAs without any physical access. High flexibilities of modern FPGAs cause that they are used for cloud accelerator in Platform as a Service (PaaS) system; however, new serious vulnerabilities emerged for these platforms. Although there are some reports about how switching activities from one region of FPGA affect other regions, details of this technique are not analyzed. In this paper, we analyzed the strength of this kind of attack and examined the impact of geometrical and electrical parameters of the victim/attacker modules on the efficiency of this attack. We utilized a Zynq-based Xilinx platform as the device under attack. Experimental results and analyses show that the distance between the victim module and the sensor modules is not the only effective parameter on the quality of attack; the influence of the relational location of victim/attacker modules could be more considerable on the quality of attack. Results of this analysis can help the FPGA manufacturer and IP developers to protect their systems against this serious attack.

1 Introduction

Increasing the tendency to employ computing hardware with more performance to speed up the computation, from one side, and reducing the cost of preparation and maintenance of these types of platforms, on the other side, creates a desire for cloud acceleration platforms. On the other hand, new advances in the semiconductor industry have increased the performance and flexibility of Field Programmable Gate Array (in terms of higher speed logic elements, logic/memory density, and more flexible architectures) makes these devices as a right choice alongside CPUs and GPUs for heterogeneous acceleration platforms. Currently, FPGAs are used in many of emerging systems like IBM Netezza to accelerate SQL tuples processing [1] or cloud computing platforms like Amazon AWS, to serve as a near general-purpose accelerator for subscribers of this service [2]. However, using these platforms raised new vulnerabilities that originate from the impact of transient voltage fluctuation on the timing characteristics of FPGA primitives [3]. It is worth mentioning that in addition to the known vulnerabilities such as hardware Trojan injection at HDL-

* Corresponding author.
**This article is an extended/revised version of an ISCISC'17 paper.
Email addresses: m.salimian@mail.sbu.ac.ir, jahanian@sbu.ac.ir
ISSN: 2008-2045

level [4] and Bitstream level [5, 6], sharing the FPGA resources between multiple users elevates the feasibility to misuse of these new vulnerabilities. Implementation of cryptographic algorithms such as AES may become insecure, even if mathematically proven to be unbreakable. Internal information of cryptosystems may leak from unwanted channels (known as side-channel information) such as power consumption and electromagnetic interference of their implementation. Power-based side-channel analysis (SCA) let the malicious user extract the key of these cryptographic algorithms by gathering traces from the power consumption of the implemented cryptosystems. Two well-known power-based SCA methods are differential power analysis (DPA) [7] and correlation power analysis (CPA) [8]. Required information for power analysis based attacks, usually achieve by real-time measurement of the circuit power consumption using an oscilloscope. While sharing FPGA resources between multiple users, let the malicious user implement a sensor, and measure delay changes of primitives to infer needed information of the circuit power consumption. Based on this idea, in [9], authors present a sensor to reveal the key of AES. Moreover, [10] used a ring oscillator based sensor to exploit the RSA key. Since a sudden increase in switching activity can create a significant voltage drop on the internal circuit, a ring oscillator based scheme used in [11] to cause a restart in the system remotely (denial-of-service). The voltage drop can increase the gates delay and consequently violate the circuit timing, so it is possible to inject fault using multiple ring oscillator with more control over ring oscillator parameters like activation frequency and duty cycle [12, 13]. Moreover, *write collision* in true dual-port memory (DPRAM) can cause a voltage drop that can be used for fault injection. When the write circuit of the DPRAM tries to write different logic levels on the same memory cell, creates transient short circuits [14]. As the voltage fluctuations caused by switching activities of a circuit propagate throw the whole power delivery network of the chip, malicious users can sense and use it to create a high-speed internal covert channel in multi-tenant FPGA-based systems [15]. Intentional voltage drop can be so large as much as it be detectable by other chips which are placed on other printed circuit boards (PCB) that use the same power supply unit (PSU). Hence, covert-channel over the PSU of a whole system is also possible [16]. This paper is an extended version of our previous work [17]. Here, we investigate the influence of different sensor features on the efficiency and quality of remote side-channel attacks. The main contribution of this paper is investigating how physical properties of cryptographic IP cores and sensors can impact on the quality of these attacks. Three other contributions are:

- Defining a new parameter (*Relative Key Correlation*) as a criterion to evaluate the quality of attack.
- Improving the operation of the sensor presented in [9] by increasing its sensitivity.
- Presenting some tips for designers to minimize the power-based internal side channel leakages in order to protect their IPs against remote side-channel attack.

We hope these analyses help the FPGA manufacturers/FPGA IP developers to minimize side-channel leakage in their designs against this kind of attack. Also, managers of multi-tenant FPGA based server can consider these tips for design placements of each user. We used a TDL-based sensor for our experiments; thus, Section 2 peruses the employed sensor and its operations. Section 3 presents our experiment analyses and results. Section 4 demonstrates our architecture of the sensor used in experiments, also explains the testbench setup, and finally, Section 5 concludes this paper.

2 Background and Related Works

2.1 Time to Digital Converter (TDC)

Time to Digital Converter is a digital circuit that measures the time interval between two events. A straightforward method to measure the time interval is time interpolation, which relies on the propagation delay of logic elements [18]. Most implementations of TDC on FPGA are based on tapped delay lines (TDL), which use the dedicated carry line components (i.e., CARRY4 on Xilinx FPGAs) for smallest granularity [19–21]. Designing an FPGA-based TDC needs more consideration than a typical digital design [22]. Covering these considerations is not in the scope of this paper. *Bubble Error* is the most severe problem in FPGA based TDC, especially in modern manufacturing technology like 28nm [23]. This problem exists in TDL based voltage sensor, so it discussed in Section 4.1. Given that the delay of TDL elements depends on power supply voltage, temperature, and process variations [3], a post-processing step needed to calibrate results for applicable TDC. Supply voltage fluctuations depend on the transient activity of the circuit, and these fluctuations can follow circuit speed [3]. Hence, delay dependency on voltage changes may cause elevation of side-channel leakage.

2.2 Voltage Fluctuation Impact on TDC Operation

Authors of [3] used an implementation of TDL-base sensor to measure the voltage fluctuations on Power Distribution Network (PDN). It is worth noting that switching activity and sub-threshold leakage current

(static power consumption) of digital circuits, drawn current from the power grid, caused voltage drops on the power supply and power grid of the chip. The value of voltage drop in a PDN consist of two components; IR drop, which is the steady-state voltage drop (R is the resistance of the PDN), and $L\,di/dt$ caused by switching activity of circuit (L is the inductance of the PDN) [24]. Based on [3], the impact of transient voltage variations on changing delay line output value is higher than process and thermal variations (on 40nm based Xilinx Virtex 6), the impact of the other two are in the order listed above. Because switched-mode voltage regulator modules (VRM) are working in lower frequency than implemented digital circuits, sudden switching activities can lead to higher transient voltage drop and higher increase in delay of primitives [3].

2.3 History of Using Internal Digital Sensor in FPGAs

Different researches employ ring oscillator-based (RO-based) or TDL-based sensors to extract physical characteristics of systems. In [25] authors used an RO-based sensor to extract process variation characteristic of an FPGA. Using RO-based sensors as a temperature monitor proposed in [26, 27]. In [28], authors used RO-based sensors for online monitoring of physical characteristic of system like temperature and voltage drop, which can be useful for healthy critical systems. Besides the mentioned contributions of using the internal sensors, some researchers propose utilizing digital sensors to increase the systems' security. RO-based or TDL-based sensor can be used for detecting power analysis attacks [29], power supply glitch attacks [30] and even hardware Trojan detection [31].

2.4 Related Works

Schellenberg *et al.* [9] used an internal sensor based on [3] to measure delay changes that tightly related to power consumption (Figure 1a). Based on Figure 1a, the TDL part consists of two sections. *Initial* delay implemented by Latches and LUTs to save logic resources of the FPGA, and *observable* delay is based on CARRY4 primitives for finest granularity measurement, similar to most Xilinx's FPGA based TDCs. They used 16 CARRY4 primitives (64 bins) for the observable part of TDL. As the number of required latches and LUTs depend on the sampling frequency and exact delay of these primitives. For simplification, the length of this part adjusted manually. They revealed the key of an implemented AES by running CPA on gathered samples from the sensor. Based on their work, the sampling frequency can

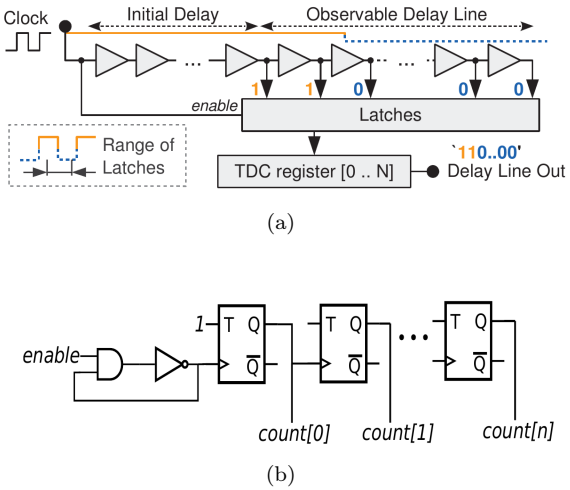


Figure 1. (a) High level architecture of sensors used in [3] and [9] (b) Ring oscillator based sensor used in [10]

be as low as the clock frequency of the AES to do CPA successfully. They also showed that even placing the AES, and the sensor far from each other, side-channel leakage on samples is enough to reveal the key. However, our work shows the direction of the distance between the sensor and the victim circuit is an essential parameter, especially for the recent generation of Xilinx's FPGAs. In [32], the same group of authors took a step beyond and prove that power analysis might be feasible when the sensor and the cryptographic IP cores implemented in different chips (aka Inter-Chip), which placed in the same circuit board and used shared power supply. In [33], authors showed that utilizing a similar type of sensor in ASIC design can act as a hardware Trojan and jeopardize the security of security-critical ICs, too. Mark Zhao and G. Edward Suh in [10] introduced a ring oscillator based sensor to detect the power consumption of FPGA. Since switching activities affects the FPGA resources delay, the frequency of the ring oscillator is inversely proportional to the switching activity of the implemented digital circuit. They used T-Flip-Flop (TFF) based counter, clocked by the ring oscillator output, to measure the frequency of ring oscillator. Figure 1b demonstrates the architecture of the sensor. Another counter used to create sufficient time intervals to sample the TFF-counter, hence clock frequency for the second counter, must be constant. The implementation used for their power analysis consists of 20 instances of mentioned RO-based sensors that distributed on the reconfigurable area of the FPGA to increase resolution and coverage area. They recover private keys of an implemented RSA on Xilinx Zynq SOC by a simple power analysis (SPA) attack. They also showed that it is possible to recover the key with no placement constraint on the sensor and also when RSA runs on the processing system (PS) of the Zynq

SOC. Since this type of sensor needs long enough time to detect small changes in switching activities, the sampling frequency must be low for the sake of accuracy. This limitation may cause these types of sensors do not be efficient enough for power analysis on high throughput implementations like AES. On the other hand, as the employed structure of ROs in [10] compose combinational loops, and these types of circuits are restricted by cloud FPGA providers like Amazon, their sensor structure seems unrealistic in real word [34]. It should be noted, [34] represents an architecture for ROs, which utilize memory element; hence, does not consider as a combinational loop.

3 Analysis of Remote Side Channel Attack

This section discusses the effect of the physical properties of power sensors and the victim module on the quality of attack. We implemented the AES algorithm for a single byte of plaintext as a circuit under analysis (CUA). The structure of the used sensor for our experiments is based on [9, 23]. Details of the experimental setup and the architecture of the sensors and the CUA are described in Section 4. We did multiple experiments with different placement on both of them to investigate the effect of the CUA and the TDL-based sensor placement on CPA results. At first, *telative key correlation* (RKC) is defined as a criterion to compare the results, and then various physical parameters will be discussed based on the performed implementations and analyses. The value of RKC for a specific number of traces defined as:

$$RKC = \frac{max_{KC}}{max_{NKC}} \tag{1}$$

Where max_{KC} and max_{NKC} are the maximum of the key correlations and the maximum of non-keys correlations, respectively. The maximum of the key correlation is the maximum value of the absolute correlation values between the Hamming weight of correct output guesses of the CUA (for the correct key guess) and power traces. In contrast, the maximum of non-key correlations is the maximum of the absolute correlation values between Hamming weights of all incorrect output guesses (for incorrect key guesses) and the power traces. In other words, this parameter represents the ratio of hypothesis value between the correct key over the closest incorrect key. Figure 2 shows the maximum key correlation and maximum non-key correlation on an instance CPA result for various possible values for byte 0 of keys on 256000 traces. As shown in Figure 2, CPA reveals the key by the rank of 1 when RKC is greater than 1. It is worth noting that greater values of RKC are interpreted as higher attack resolution. Figure 3a shows the progressive curve of CPA results. Grey curves show the

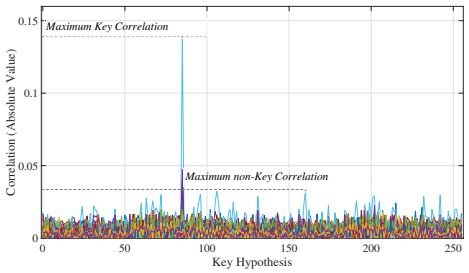
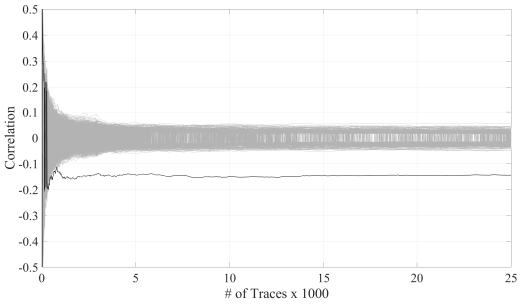
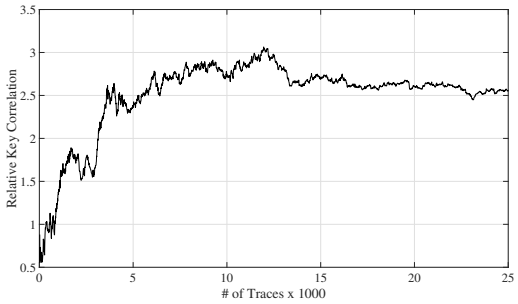


Figure 2. CPA result from 256K power traces of the S-Box outputs when the key byte is 85

correlation values of incorrect keys, while the black curve represents the correlation value of the correct key in terms of increments in the number of traces. Figure 3b shows the progressive curve of RKC. As can be seen in this figure, the RKC has very small changes using more than about 15000 traces.



(a)



(b)

Figure 3. (a) Progressive curve of CPA results, shows there are no significant changes for the maximum key correlation (black line) when the number of participant traces is more than 15K. (b) Progressive curve of *relative key correlation*; indicates the maximum non-key correlation has mild changes after 15K of traces, too. Hence cause slight changes in RKC for this attack.

3.1 Analysis 1: Distance from Sensor to CUA

An essential parameter in this kind of attack is the relational location of the sensor and the CUA. We are going to analyze this parameter as the impact of distance on the quality of attack. Figure 4 shows

the floorplan view of the used FPGA platform (Zynq 7020). In this figure, the placement location of the ILA (integrated logic analyzer) and the VIO (virtual I/O) indicated as the yellow rectangle. These circuits used as an interface between the PC and the FPGA. As said before, detailed information about the experimental setup described in Section 4. As the first experiment, the CUA placed at the bottom right corner of the FPGA chip (red rectangle in Figure 4), and the sensor shifted in three directions; horizontal, vertical and diagonal. Each of the arrows in Figure 4 shows the direction of movement for the related experiments. The colored rectangles (blue, green, and purple rectangles) around the CUA placement block show the initial placement of the TDL part of the sensor for each experiment. The sensor consists of two parts, the TDL part, and the adder tree and ones-counter (Section 4.1). All figures that relate to placement only show the TDL part for simplification. The distances between the sensors and the CUA (X-axis) are expressed as the number of CLBs (configurable logic blocks) between them. It is worth mentioning that the employed FPGA's dimensions are 72 CLBs by 150 CLBs. Similar to Vivado IDE, dedicated memory blocks and DSP blocks assumed one CLB to express distance in horizontal and diagonal movements. For the diagonal movement, distances are calculated by the Pythagorean method. Dimensions of the CUA are 8 CLBs by 4 CLBs, and for the sensor are 17 CLBs by 2 CLBs. Hence maximum distance for horizontal, vertical and diagonal movements can be 62, 119 and 123 CLBs, respectively. Figure 5 shows that increasing

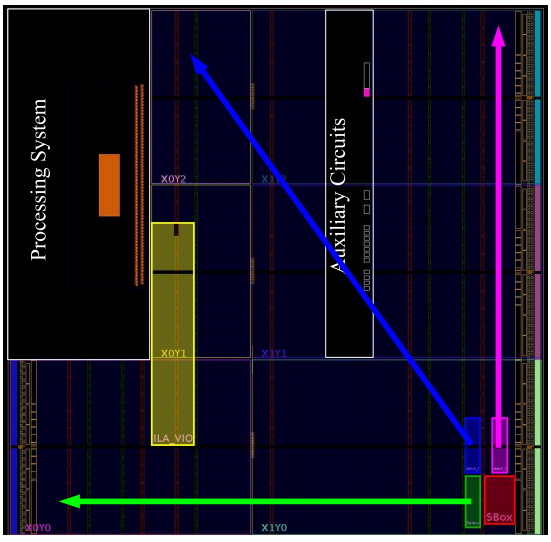


Figure 4. Directions of sensor movements; the CUA location is the red rectangle, and the yellow one shows ILA and VIO placement. Green, Blue, and purple boxes show the sensor's initial placement for horizontal, diagonal, and vertical movements, respectively. The results of this analysis are in Figure 5.

the vertical distance between the sensor and victim

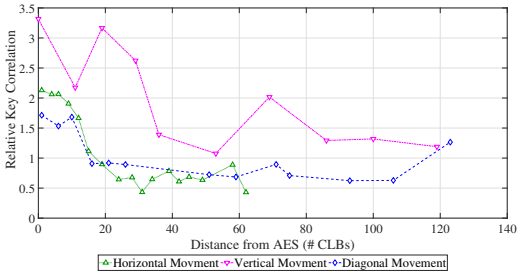


Figure 5. Effect of sensor movement on the RKC, for each direction. Increasing the distance from the CUA reduces the RKC as much as, in horizontal and diagonal movements, this value goes under one (failure in key extraction).

CUA, reduces the leakage that is necessary for a successful attack. However, the correlation of the key is reasonably more significant than the non-key correlations. In other words, even at the maximum vertical distance, information for an attack was still available, and the CPA attack's result was successful. For horizontal and diagonal movements, the value of RKC, which shows the information leakage, decreased as far as the key correlation is no longer greater than non-key correlations. In horizontal movement, minimum values of RKC created when the sensor is near the clock backbone of the chip (the boundary between the left side and the right side regions) and when the sensor sticks to IO blocks (left side of the chip). Similar effects exist in vertical movements when the sensor is near or across HROW (horizontal clock rows - horizontal clock routes in the middle of each clock region). The RKC curve for diagonal movements shows attenuation effect on information leakage, from auxiliary circuits of the chip. Concerning the starting points of each curve, it is noticeable that even at close distances between the sensor and the CUA, when the sensors and the CUA are in identical vertical columns, useful information leakage is more significant than other modes. Comparing the results from different movement directions, especially RKC values for far distances, rises guesses about the power distribution network of the chip. As the results indicate, placing the sensor and the CUA in the same columns of logic slices induces more information leakage. In other words, the sensor sense voltage drop on the PDN better when placed in the same vertical direction as the CUA. Hence we guess main power rails of employed FPGA (or maybe all Xilinx FPGAs) are routed vertically, or the density of horizontal routes of the power grid is limited compare to vertical power rails, or even it may be caused by the thickness of horizontal power routes. These assumptions are not far-fetching; since memory blocks, which are placed vertically inside Xilinx FPGAs, feed power from dedicated power pins.

Either way, it seems useful power information leakage is significantly more in the vertical direction of the CUA. The RKC value for 5th point of vertical movement shows a significant leakage drop compared to the previous point. We guess it was caused by changing the clock region somehow. The next experiment designed to investigate these guesses on the PDN. Progressive RKC values for minimum and maximum horizontal and vertical distances demonstrate how distance can affect the number of needed traces for a successful attack (Figure 6). The horizontal and

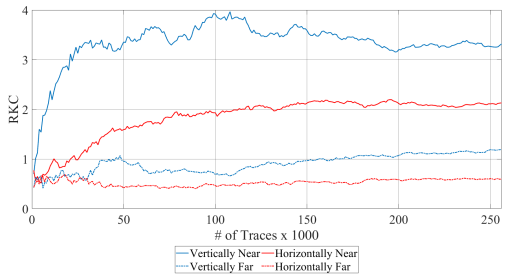


Figure 6. Progress RKC curve for minimum and maximum of horizontal and vertical distances

vertical axes of Figure 6 are the same as Figure 3b. In this figure, blue and red lines show the progressive RKC values for vertical and horizontal displacement, respectively. Also, dashed lines represent the maximum distances. According to Figure 6, the attack needs at least $25K$ traces for the nearest horizontal distance. Nevertheless, for the farthest distance, the attack failed. On the other hand, vertical displacement had a lower impact on attack quality, although attack quality attenuated as the distance increases. In correlation power analysis attacks, the column of the traces matrix that has the highest correlation value with correct intermediate value guesses (hypothesis for correct key) known as the correct time [35]. An interesting observation in our experiments was shifting in the correct time moment (column number in correlation matrix) by changing the distance between the CUA and the sensor (Figure 7). In Figure 7, the vertical axis is absolute values of correlations, and the horizontal axis refers to the column number of traces matrix (also correlation matrix). Each curve shows correlation values for the correct key. In other words, curve show row values of the correlation matrix, which relates to the correct key. The distance between the CUA and the sensor for the blue and red curves is 29 CLBs and 53 CLBs, respectively. For both cases, the attack extracts the key by rank one, successfully. The highest correlation for the blue curve appeared in the 8th column (8th sample point of traces), while for the red curve, which relates to more distance between two circuits, it appeared in the 9th column. As

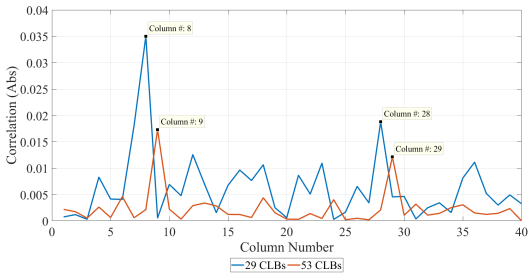


Figure 7. Correct time shift by changing the distance between the CUA and the sensor.

this figure indicates, increasing the distance, besides reducing the quality of attack (lower value of correlation for the correct key), causes a delay in the correct time moment. This behavior was perceptible in both horizontal and vertical displacement. Based on this observation, it can be concluded that the power rails (generally PDN) like transition lines have a delay in distributing voltage drops. Another observed

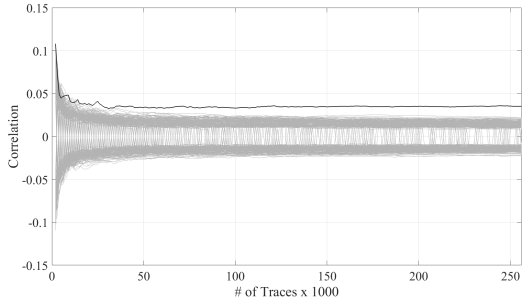
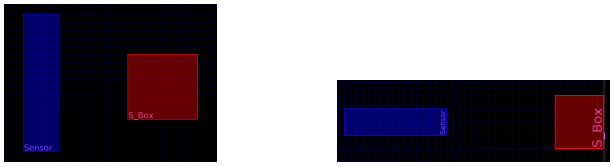


Figure 8. Positive correlation values between traces and correct intermediate hypothesis (correct key guess)

behavior in some horizontal displacement results is the presence of successful attacks with positive correlation values(Figure 8). It is normally expected that by increasing power consumption, the voltage drop will cause an increase in primitives' delay. In other words, sampled values in time intervals with higher power consumption must be smaller and vice versa. Hence, the correlation value between samples and intermediate data (for the correct key guess) must be negative, based on the sensor's operation. We believe that these behaviors originate from the overcharging of adjacent power rails. The critical point is that the overcharge amount in adjacent power rails is correlated enough to extract the correct key. It also should be noted that this behavior never was observed for vertical or diagonal displacements (for sampling in half of the clock cycle).

3.2 Analysis 2: Relative Location

Based on previous experiments results, it can be concluded intuitively that vertical power rails play a more significant role in distributing power inside the FPGA. We used two placement schemes for the CUA and the sensor (Figure 9) to investigate this inference. Figure 9a indicates the situations when the sensor



(a) (b)

Figure 9. (a) Side-by-side placement of the sensor and the CUA. The distance between them kept 4 CLBs for all respected experiments. (b) Modules placed vertically (figure rotated 90 degrees) while the distance between them is 17 CLBs. Only 2 CLB columns of the CUA have overlap with the sensor. In both figures, the red and the blue blocks are placement blocks for the CUA and the sensor, respectively.

and the CUA placed as side-by-side neighbors (generally when both are in the same rows of CLBs), and the distance between them is 4 CLBs. Based on our assumption, if these two blocks move together horizontally while the distance between them remains the same, the *RKC* values should have small changes. In other words, because most of the leaked information is from vertical power rails and horizontal power rails have less impact on total leaked information, the amount of leaked information from the CUA must remain almost the same. On the other hand, more significant changes are expected in *RKC* values by moving them vertically. We swap the location of them in three columns and four rows, as indicated in Figure 10, to check the mentioned opinions. Arrows in Figure 10 show the direction of the sensor movements. Note that the CUA moves with the sensor, while their distance remains unchanged. Figure 11 shows the changes in *RKC* values for these locations. Each curve in Figure 11a represents the *RKC* variations compare to the different distances from the bottom of the chip. In comparison with Figure 11b, which represents changes based on moving in rows, there are higher variations. The maximum value of each curve happened in the first data point, when the CUA and the sensor are at the chip bottom. Changing the locations in the second and third columns of Figure 11a show auxiliary circuits' effects on power traces, which makes *RKC* values smaller. The fourth data point of the second curve is when the sensor is stuck to the right side of the auxiliary circuits, and for the third point, the CUA is stuck to the left side. Figure 11b uses the same datasets without results from the most bottom row, but plot them row by row. This figure shows by changing the locations horizontally, variation in *RKC* is smaller except for the third row. As mentioned before, the last data point in the curve relates to the third row, is when the CUA sticks to auxiliary circuits. Despite similar placement for data points in the second and fourth row, related curves do not represent a similar behavior. Based on our

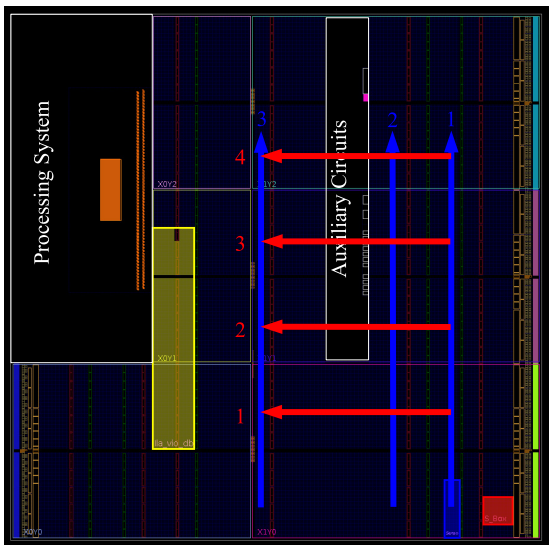
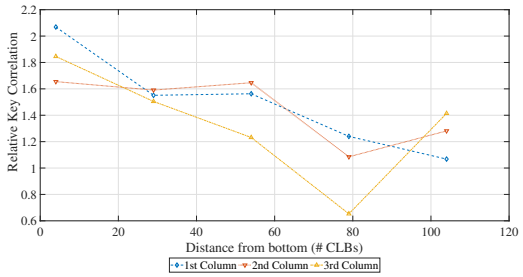
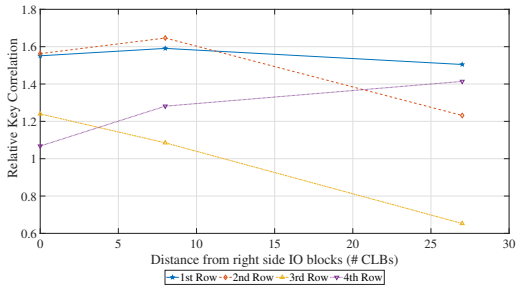


Figure 10. Directions of the sensor movements and the CUA side by side while the distance between them kept the same (4 CLBs). The blue p-block and the red p-block relate to the sensor and the CUA placement constraints, respectively.



(a)



(b)

Figure 11. *RKC* variations based on the location of the sensor and the CUA when moving both of them, as shown in Figure 10. (a) Highlight variations based on moving them vertically and (b) for changing location horizontally.

tests and Vivado IDE floorplan view, it seems the active part of auxiliary circuits is the part in the upper half of the middle region. Perhaps this area of auxiliary circuits is the location of Boundary-SCAN-X0Y0 because, in our design, only this part of the circuit used. This explanation can justify the minimum RKC value on the third column curve of Figure 11a as already mentioned. Figure 12 demonstrates the progressive RKC curves of attack results from 3 different locations of Figure 10, for more detailed analysis on horizontal leakage. In Figure 12, the blue curve re-

Figure 12. Progressive RKC plot of CPA results for three different situation of horizontal leakage. The red and the green RKC curves relate to when the CUA and the sensor stick to auxiliary circuits, respectively. For the blue one, none of them affected by auxiliary circuits.

lates to the second column's first location. Therefore, neither the CUA nor the sensor was affected by the auxiliary circuits. The red curve relates to the second column's fourth location; the CUA sticks to the auxiliary circuits. The last curve (the green one) relates to the fourth data point of the third column, so the sensor sticks to auxiliary circuits. When none of the modules were affected by the auxiliary circuits, the attack needs 36K traces to extract the key. On the other hand, the red curve shows the number of needed traces for a successful attack was more than $218K$ when the sensor sticks to auxiliary circuits. Based on the green curve, $256K$ traces were never enough for extracting the key when the CUA was adjacent to auxiliary circuits (the RKC value never goes higher than 1). As this figure shows, placing the CUA or the sensor near auxiliary circuits can significantly reduce the quality of attack. The next experiments are based on Figure 9b placement scheme. Note that this figure rotated 90 degrees. In this case, the sensor placed on top of the CUA, while the distance between them is 17 CLBs. Because the sensor width is 2 CLBs but for the CUA is 4 CLBs, only two columns of the CUA have overlap with the sensor. Figure 13 shows the directions of moving these blocks in the FPGA floorplan. The color of the blocks is the same as before. Similar to the last experiment, single datasets were used to analyze the changes in RKC values after moving the blocks horizontally and vertically. Analysis of these placements can help us to prove our conjectures about the PDN of the chip. Horizontal and vertical shifting of these blocks might have a small effect on RKC results because the sensor and the CUA placed in the same columns of CLBs, and the vertical distance remained Intact. Figure 14a shows RKC variations in the vertical movement. Therefore, the X-axis of the plot shows the distance from the chip bottom. Figure 14b shows the plot of datasets row by row (horizontal movement of blocks), so the X-axis is the distance from the right side IO-blocks. Relative key correlation values in Figure 14a are mostly higher, compare to previous experiments. However, changes in these values are not small by moving blocks in columns, but in all cases, the maximum correlation of the key is higher than all correlations of non-key guesses. The shape of these curves is interesting and

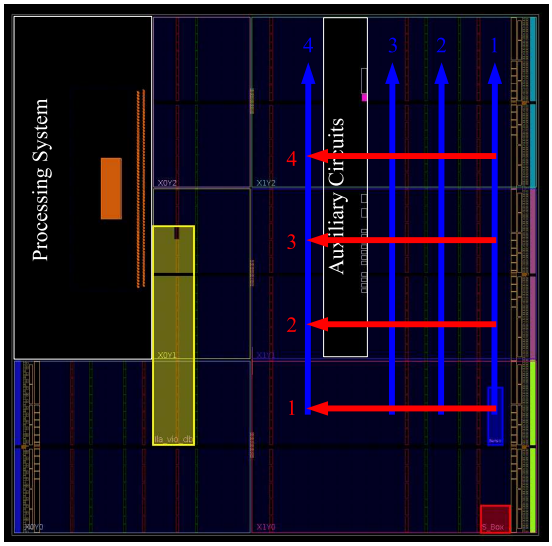
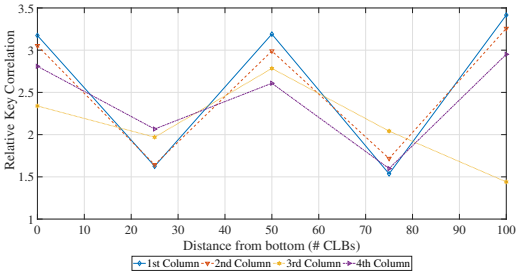
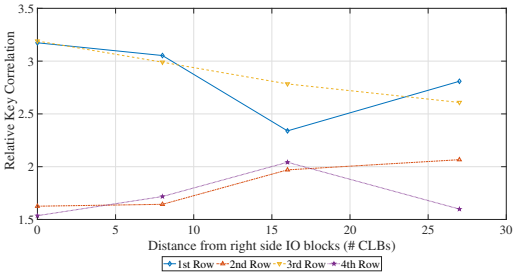


Figure 13. Directions of the sensor movement and the CUA while keeping overlapped columns the same. The blue p-block and the red p-block relate to the sensor and the CUA placement constraints, respectively.

informative. The relation to placements will be maximum when both the sensor and the CUA blocks are in a same clock region, and the minimum points (except for the last point of the third column) relate to when they are in two different vertically adjacent clock regions. It should be noted that the CUA is in the downer clock region, and the sensor is in the upper region. The last point of the third column comes from when the CUA sticks to the active part of the auxiliary circuit. These results show similar effects of clock regions, which mentioned in Section 3.1. We guess vertical power rails inside each of clock regions use the same inner power ring. Hence the RKC values are higher when the CUA and the sensor placed at the same clock region compare to placement in different clock regions. Curves in Figure 14b show mild changes in RKC values when moving the CUA and the sensor inside different CLB rows. RKC Curves



(a)



(b)

Figure 14. *RKC* variations based on the location of the sensor and the CUA when moving both, as shown in Figure 13. (a) Indicates variations based on vertical movements, and (b) for horizontal movements.

of the first and the third rows, when the sensor and the CUA placed in the same clock region, have higher *RKC* values. Note that *RKC* values relate to the last placement point of each column are not used in this plot. The results of these experiments can be considered as evidence for the validity of our assumptions; most internal power leakage of a module under analysis propagates vertically. Similar to the previous analysis (e.g. Figure 12), Figure 15 demonstrates the progressive RKC curves of 2 different locations of Figure 13. In Figure 15, the red plot relates to the third

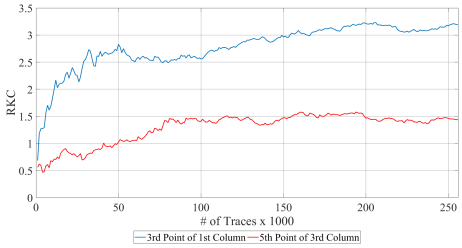
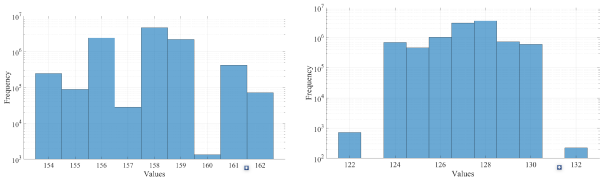
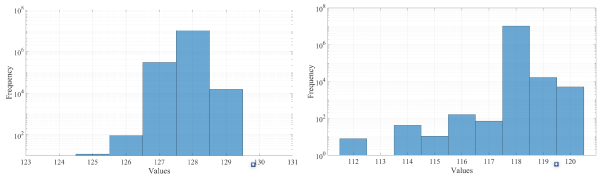


Figure 15. Progressive RKC plot of CPA result for 2 different situation of vertical leakage. The red curve relate to when the CUA and the sensor stick to auxiliary circuits, respectively. For the blue one, none of them affected by auxiliary circuits.

column's last location. Based on Figure 13, the CUA was adjacent to auxiliary circuits in this location. For the blue curve, none of the circuits was affected by auxiliary circuits. As this figure demonstrates, when none of the sensors or victim modules appeared in auxiliary circuits' vicinity, a successful attack needs less than 2k traces. However, when the CUA sticks to auxiliary circuits, the attack needs more than 50K traces. Comparing these results shows that placing the victim circuit or the sensor adjacent to auxiliary circuits can reduce vertical leakage, too. Comparing the dispersion of samples can help to understand how auxiliary circuits influence attack quality. Figure 16 demonstrate the histogram of gathered samples from the sensor for four different attacks. Two cases of this figure relate to the situation that the CUA or the sensor was adjacent to auxiliary circuits. In all plots



(a) (b)



(c) (d)

Figure 16. Histogram of samples for four different attacks: (a) Successful attack, neither the CUA nor the sensor were adjacent to auxiliary circuits. (b) Unsuccessful attack, neither the CUA nor the sensor were adjacent to auxiliary circuits. (c) Successful attack, the CUA or the sensor was adjacent to auxiliary circuits. (d) Unsuccessful attack, the CUA or the sensor was adjacent to auxiliary circuits.

of Figure 16, horizontal and vertical axes relate to the sample values and frequency (logarithmic scale), respectively. Figure 16a is the histogram of samples, which was ultimately successful. Neither the sensor nor the CUA was affected by the auxiliary circuits, in this case. The next histogram relates to samples of attack which was unsuccessful because of the high horizontal distance. In contrast, for two other histograms (Figures 16c and 16d), circuits were adjacent to the auxiliary circuits. For Figure 16c, the attack was successful, but the result has been a failure for the other one. It is worth mentioning that samples' variances were 2.73, 1.98, 0.03, and 0.003, respectively. Based on samples' histograms and variance values, it can be seen that increasing the horizontal distance does not cause a significant change in the dispersion of samples. However, when modules were adjacent to auxiliary circuits, dispersion of samples reduces significantly. Given that, reduction in samples' disper-

sion can be considered as lower voltage fluctuations; a possible conclusion could be the better current flow of the power distribution network in these areas.

4 Experimental Setup

The value of the experimental parameters is significant in the proposed analysis. Therefore, the experimental parameters of the sensors and also circuit under test are described in detail in this section.

4.1 Sensor

The sensor used in our experiments is designed based on [9] with some modifications inspired by [23], to improve its sensitivity. Figure 17 demonstrates the architecture of our sensor. Similar to [9], this sensor used primitive with high delay (e.g., LUT6s and LUT5s) for the initial delay, but we used four parallel Tapped Delay Lines of CARRY4 primitives for the observable delay part to increase quantization of the sensor (similar to [23]). Based on our tests, sensitivity to delay changes in this architecture is higher than the sensor presented in [9]. Because of using 4 TDLs, generating Bubble Error turned to a more serious problem [23]. Generating Bubble Error is the

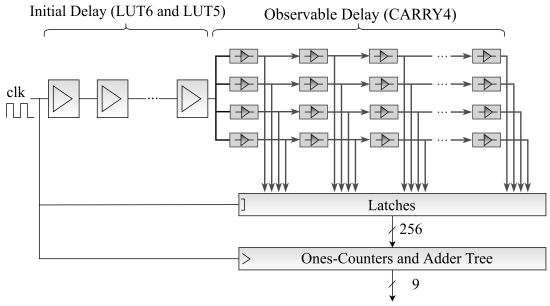


Figure 17. Structure of the employed sensor. The adder tree part is a pipelined structure that consists 6 stages. Structure of this sensor based on [9] and [23].

most undesired behavior of TDLs and is more serious in recent manufacturing technologies. Bubble Errors are wrong logic values in one or some bits of the TDL outputs. For instance, in a tapped string like "1...110100...0", the zero between two ones is Bubble Error, and the string must be in the following pattern "1...111100...0". To clarify the origin of these types of errors, suppose letting a signal with a positive logic value propagate through a TDL and sampling the output of TDL by registers. We expect the signal to propagate in TDL bins one by one. Therefore, the output must be a string like "1...111100...0" in an ideal situation, but the delay of similar logic elements or even wires with the same length is not equal, because of process variation. Consequently, the output of lower value bins may be zero, as regards higher value bins show one (the delay of lower value output wire is greater than the total delay of wire and gate of the higher bin). The other reason is related to the skewness of the clock distribution network. If the clock edge reaches earlier to lower bins' sampling register than higher sampling registers, output samples can contain bubble(s). In conclusion, unbalanced delays of CARRY4 primitives (or any other primitive used in TDL) and wires, and skew of the clock distribution network in the chip caused generating bubble errors. One correction way of bubble errors is priority encoder as used in [9], but since our sensor used 4 TDLs, Ones-Counter encoder employed for bubble error correction, similar to [21, 23]. The operation of this sensor is the same as [9]. The positive level of clocks must have enough time to propagates in all elements of the initial delay and the first elements of the observable delay part of each CARRY4 TDLs. Latches only capture the length of the propagated positive level of the clock in the observable part, when the clock value is equal to one. Then the pipelined adder tree counts the number of existed ones in the binary string captured by Latches. Note that the placement block for ones-counter and adder tree not shown in Figures 4,9,10 and 13, for simplification. Because of using ones-counters and adder tree, our sensor occupies more resources of FPGA than the sensor represented in [9], but for the sake of higher sensitivity, we used the mentioned sensor. Table 1 compare these two sensors. Figure 18 compares the results of attacks with the new sensor and the sensor introduced in [9], by progressive RKC curve. Here, the victim circuit was the DPA contest v2 AES [36] and the goal of the attack was the first key byte of the AES. The blue curve and the red one in Figure 18 represent the progressive RKC values of attack by the new sensor and the sensor represented in [9], respectively. As this figure illustrates, the new sensor needs less than 1K traces for a successful attack. The num-

			The Sensor represented in [9]	The New Sensor
Occupied Resources	Initial Delay	# Latches∗	43	0
		# LUTs∗	43	12
	Observable Delay	# TDLs	1	4
		# CARRY4∗	16	64
		# Registers∗	63	256
	Coding Circuit	Type	Priority Encoder	Ones-Counter
		# Occupied Slices⊥	83	531
		# Registers⊥	79	610

Table 1. Comparison of occupied resources by the sensor represented in [9] and the new sensor. ∗ - Regarding the structure of both sensors, the number of occupied resources for Initial Delay and Observable delay parts are strongly dependent on the time interval of each clock cycle, that the value of the clock signal is one. ⊥ - Amount of occupied slices and registers for coding circuit depend on the length of the observable delay.

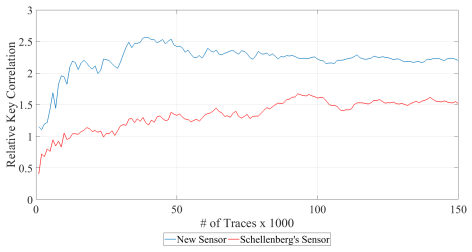
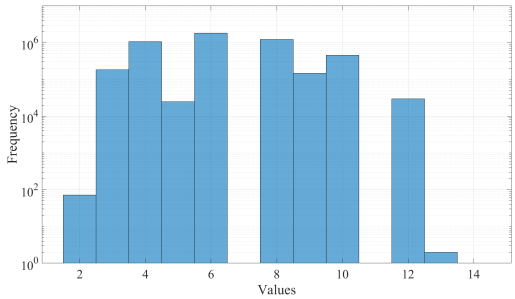
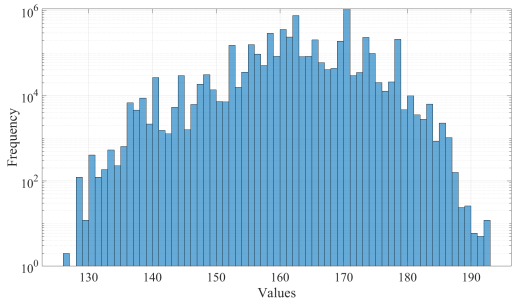


Figure 18. Comparing the impact of new sensor (blue curve) and the one represented in [9] (red curve) on the attack quality (in similar situation), with help of progressive RKC curve.

ber of needed traces from [9] sensor was $13K$. Hence the new sensor reduces the number of needed traces at least ten times. To make this comparison fair, we tried to make the test conditions as uniform as possible. Figure 19 demonstrates histograms of gathered sample for mentioned attacks. Axises of these plots



(a)



(b)

Figure 19. Histogram of samples that used for attacks represented in Figure 18. (a) Histogram of samples gathered by the sensor represented in [9]. (b) Histogram of samples acquired by the new sensor. The frequency (vertical axis) of both histograms are represented on a logarithmic scale.

are similar to the Figure 16. Horizontal and vertical axes relate to the sample values and frequency on a logarithmic scale, respectively. The total number of samples for each histogram is 4,950,000 samples. Figure 19a, demonstrate the histogram of samples from the attack by the sensor represent in [9] (The red curve of Figure 18), and Figure 19b relates to samples gathered by the new sensor (The blue curve of Figure 18). As this figure indicates, samples from the new sensor have a near 6 times higher range of variations. On the other hand, there is no bin with zero frequency in the histogram of samples from the new sensor (As there was only one sample with the value of 127, the frequency of this sample value is zero on logarithmic scale representation). The existence of bin(s) with zero value in Figure 19 stems from the behavior of the priority encoder. Since the priority encoder makes no difference for inputs like "111111000", "111101000" and "111001000" (generate same output value), it is quite normal that priority encoder based sensor to never generate some values. When it comes to an ones-counter coder, this behavior occurs rarely. Hence the new sensor has less quantization error. It is worthwhile to mention that the efficiency of both sensors was examined in various scenarios (different distances and locations). In almost all cases, the new one generates better results (less needed traces and higher RKC values).We also examined other structures like using four sensors of [9] and adding results, using priority encoder for each line and adding output values, and even sensors with more or less number of TDLs, but our tests and comparisons, indicate better results from this one, totally.

4.2 Circuit Under Analysis

We implemented the first two steps of AES for one byte of key and plaintext; AddRound key (bitwise XOR of plaintext and the key) and a single byte Rijndael S-box, as the circuit under analysis (CUA). Figure 20a demonstrates the CUA architecture. Plaintexts are generated by a simple 8-bits counter every 8 clock cycle. Outputs of the S-box overwrite on registers at every rising edge of the clock. The switching activity of the CUA increased for our tests by 16 times amplification on the outputs; instead of using eight registers to capture the S-box output, we used 128 registers (each bit of output stored in 16 registers). Since our goal in this paper was investigation on the influence of the sensor placement and the distance between the CUA and the sensor on CPA results, not only we had to use placement blocks for each of the module instances, but we needed to fix the location of each logic elements of modules inside their placement box, from one experiment to another. Using full AES could make this task more difficult or even impossible. The drawback of the CUA was the small switching activity, which compensated by amplifying the output of the S-box.

4.3 Whole System Setup

The employed platform was Z-turne board, which is based on Xilinx Zynq XC7Z020. Xilinx Vivado IDE

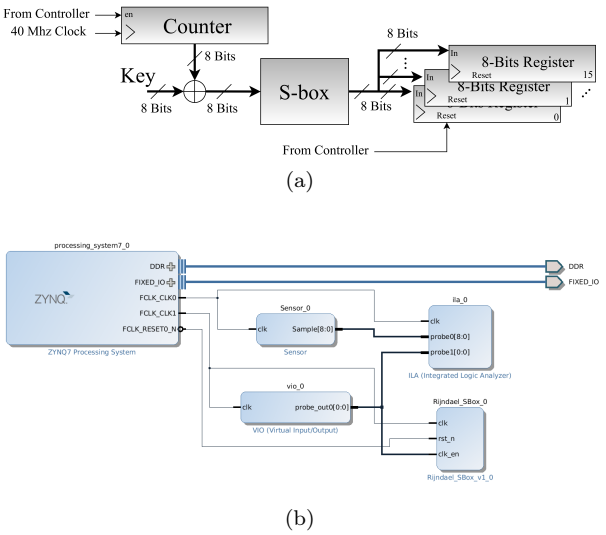


Figure 20. (a) Architecture of used module under analysis, the counter that generates plaintexts and destination registers of S-box not shown. (b) Block diagram of the implemented system on FPGA (This structure remains intact for all tests).

does all synthesis and implementation steps. Samples from the sensor are stored in internal RAM blocks using integrated logic analyzer (ILA) IP, then sent to PC through Xilinx JTAG cable, by running a TCL script under Vivado. Gathered samples used in Matlab-2018a on a laptop with Intel Core-I7 with 16GB of DRAM, which runs CPA scrip on samples. Figure 20b shows the block diagram of our design. The frequency of the CUA and the sensor are $40MHz$ and $200MHz$, respectively. It should be noted that, for more accurate results in comparisons, the sampling process is done on both half clock cycles (by 180°phase shift) for all experiments. The VIO (Virtual input/output) block generates the trigger signal for ILA block and the CUA; this helps us to synchronize our design and make the alignment of traces easier. The other port of the ILA gathers traces from the sensor. For the sake of This structure of the block diagram remains intact for all tests and only placement location changes in different experiments.

5 Conclusion

This paper analyzes the vulnerabilities of multi-tenant FPGA-based systems against non-physical access side-channel attacks. We used an improved version of the TDC-based sensor that presented in [9] to experiment on how sensor location can affect leaked information in samples from the dynamic power consumption of a module. Experimental results indicate that the distance between the victim module and the sensor, as well as their locations, can influence useful leaked information, due to adjacent internal circuits. Based on the results, the best way to split an FPGA between two users is vertical splitting; this lets users minimize internal side-channel leakage by placing critical modules near the IO blocks of their side. Moreover, some of addressed studies like [37] demonstrate that injecting random delay to cryptographic algorithms can harden classic power analysis attacks (with physical access) by creating misalignment in power traces. Since moving the CUA inside the chip can have a similar impact on power traces, using methods like partial reconfiguration to displace security-critical circuits like cryptographic modules can increase the system's security over remote power analysis attacks. Moreover, we showed that most information leaked when both the sensor and the module under analysis placed in the same columns of CLBs. Results also showed auxiliary circuits could have a hiding effect on adjacent circuits, although these impacts are not enough to prevent side-channel leakage, especially for vertical placement of the sensor. Clock backbone and horizontal clock routes (HROW) might have similar effects, but are not significant as auxiliary circuits.

References

[1] Ibm puredata system for analytics architecture, 2014.

[2] Amazon ec2 f1 instances, available on https://aws.amazon.com/ec2/instance-types/f1.

[3] D. R. E. Gnad, F. Oboril, S. Kiamehr, and M. B. Tahoori. Analysis of transient voltage fluctuations in fpgas. In *2016 International Conference on Field-Programmable Technology (FPT)*, pages 12–19, Dec 2016.

[4] Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou. Designing and implementing malicious hardware. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, LEET'08, pages 5:1–5:8, Berkeley, CA, USA, 2008. USENIX Association.

[5] R. S. Chakraborty, I. Saha, A. Palchaudhuri, and G. K. Naik. Hardware trojan insertion by direct modification of fpga configuration bitstream. *IEEE Design Test*, 30(2):45–54, April 2013.

[6] P. Swierczynski, M. Fyrbiak, P. Koppe, and C. Paar. Fpga trojans through detecting and weakening of cryptographic primitives. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(8):1236–1249, Aug 2015.

[7] Paul Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. Introduction to differential power analysis. *Journal of Cryptographic Engineering*, 1(1):5–27, Apr 2011.

[8] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. volume 3156, pages 16–29, 08 2004.

[9] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori. An inside job: Remote power analysis attacks on fpgas. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1111–1116, March 2018.

[10] M. Zhao and G. E. Suh. Fpga-based remote power side-channel attacks. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 229–244, May 2018.

[11] D. R. E. Gnad, F. Oboril, and M. B. Tahoori. Voltage drop-based fault attacks on fpgas using valid bitstreams. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, pages 1–7, Sep. 2017.

[12] Jonas Krautter, Dennis R. E. Gnad, and Mehdi B. Tahoori. Fpgahammer: Remote voltage fault attacks on shared fpgas, suitable for dfa on aes. 08 2018.

[13] Dina Mahmoud and Mirjana Stojilović. Timing violation induced faults in multi-tenant fpgas. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1745–1750. IEEE, 2019.

[14] Md Mahbub Alam, Shahin Tajik, Fatemeh Ganji, Mark Tehranipoor, and Domenic Forte. Ramjam: Remote temperature and voltage fault attack on fpgas using memory collisions. In *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pages 48–55. IEEE, 2019.

[15] Dennis RE Gnad, Cong Dang Khoa Nguyen, Syed Hashim Gillani, and Mehdi Baradaran Tahoori. Voltage-based covert channels in multi-tenant fpgas. *IACR Cryptol. ePrint Arch.*, 2019:1394, 2019.

[16] Ilias Giechaskiel, Kasper Bonne Rasmussen, and Jakub Szefer. C3apsule: Cross-fpga covert-channel attacks through power supply unit leakage. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1728–1741. IEEE, 2020.

[17] Milad Salimian and Ali Jahanian. Analysis of geometrical parameters for remote side-channel attacks on multi-tenant fpgas. In *2020 17th International ISC Conference on Information Security and Cryptology (ISCISC)*, pages 28–35. IEEE, 2020.

[18] Jian Song, Qi An, and Shubin Liu. A high-resolution time-to-digital converter implemented in field-programmable-gate-arrays. *IEEE Transactions on Nuclear Science*, 53(1):236–241, Feb 2006.

[19] C. Liu and Y. Wang. A 128-channel, 710 m samples/second, and less than 10 ps rms resolution time-to-digital converter implemented in a kintex-7 fpga. *IEEE Transactions on Nuclear Science*, 62(3):773–783, June 2015.

[20] Jinyuan Wu and Zonghan Shi. The 10-ps wave union tdc: Improving fpga tdc resolution beyond its cell delay. In *2008 IEEE Nuclear Science Symposium Conference Record*, pages 3440–3446. IEEE, 2008.

[21] E. Bayer and M. Traxler. A high-resolution (< 10 ps rms) 48-channel time-to-digital converter (tdc) implemented in a field programmable gate array (fpga). *IEEE Transactions on Nuclear Science*, 58(4):1547–1552, Aug 2011.

[22] J. Wu. Several key issues on implementing delay line based tdcs using fpgas. *IEEE Transactions on Nuclear Science*, 57(3):1543–1548, June 2010.

[23] Y. Wang, J. Kuang, C. Liu, and Q. Cao. A 3.9-ps rms precision time-to-digital converter using ones-counter encoding scheme in a kintex-7 fpga. *IEEE Transactions on Nuclear Science*, 64(10):2713–2718, Oct 2017.

[24] K. Arabi, R. Saleh, and X. Meng. Power supply noise in socs: Metrics, management, and measurement. *IEEE Design Test of Computers*, 24(3):236–244, May 2007.

[25] Haile Yu, Qiang Xu, and Philip HW Leong. Fine-grained characterization of process variation in fpgas. In *2010 International Conference on Field-Programmable Technology*, pages 138–145. IEEE, 2010.

[26] Christoph Ruething, Andreas Agne, Markus Happe, and Christian Plessl. Exploration of ring oscillator design space for temperature measurements on fpgas. In *22nd International Conference on Field Programmable Logic and Applications (FPL)*, pages 559–562. IEEE, 2012.

[27] John J León Franco, Eduardo Boemo, Encarnación Castillo, and Luis Parrilla. Ring oscillators as thermal sensors in fpgas: Experiments in low voltage. In *2010 VI Southern Programmable Logic Conference (SPL)*, pages 133–137. IEEE, 2010.

[28] Kenneth M Zick and John P Hayes. Low-cost sensing with ring oscillator arrays for healthier reconfigurable systems. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 5(1):1–26, 2012.

[29] Adrien Le Masle and Wayne Luk. Detecting power attacks on reconfigurable hardware. In *22nd International Conference on Field Programmable Logic and Applications (FPL)*, pages 14–19. IEEE, 2012.

[30] Kenneth M. Zick, Meeta Srivastav, Wei Zhang, and Matthew French. Sensing nanosecond-scale voltage attacks and natural transients in fpgas. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '13, page 101–104, New York, NY, USA, 2013. Association for Computing Machinery.

[31] Tamzidul Hoque. *Ring oscillator based hard-*

ware trojan detection. PhD thesis, University of Toledo, 2015.

[32] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori. Remote inter-chip power analysis side-channel attacks at board-level. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–7, Nov 2018.

[33] David Knichel, Thorben Moos, and Amir Moradi. The risk of outsourcing: Hidden sca trojans in third-party ip-cores threaten cryptographic ics. In *2020 IEEE European Test Symposium (ETS)*, pages 1–6. IEEE, 2020.

[34] Ilias Giechaskiel, Kasper Bonne Rasmussen, and Jakub Szefer. Measuring long wire leakage with ring oscillators in cloud fpgas. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*, pages 45–50. IEEE, 2019.

[35] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.

[36] Dpa contest v2.

[37] Marco Bucci, Raimondo Luzzi, Michele Guglielmo, and Alessandro Trifiletti. A countermeasure against differential power analysis based on random delay insertion. In *2005 IEEE International Symposium on Circuits and Systems*, pages 3547–3550. IEEE, 2005.

Milad Salimian received his B.S. degree in computer engineering from PNU University, Tehran, Iran 2013 and the M.S. degrees in computer engineering from Shahid Beheshti University of Technology, Tehran, Iran in 2020. His current research interests are hardware security and FPGA-based embedded system design.

Ali Jahanian received his B.S. degree in computer engineering from University of Tehran, Tehran, Iranin 1996 and the M.S. and Ph.D. degrees in computer engineering from Amirkabir University of Technology,Tehran, Iran in 1998 and 2008, respectively. He joined Shahid BeheshtiUniversity, Tehran, Iran in 2008. His research interests are Hardware security and Biochips design.