J Cryptogr Eng (2016) 6:239–247
DOI 10.1007/s13389-016-0129-2

SPECIAL SECTION ON PROOFS 2014

Method taking into account process dispersion to detect hardware Trojan Horse by side-channel analysis

Xuan Thuy Ngo[1] · **Zakaria Najm**[1] · **Shivam Bhasin**[1,3] · **Sylvain Guilley**[1,2] · **Jean-Luc Danger**[1,2]

Received: 8 February 2015 / Accepted: 12 March 2016 / Published online: 7 April 2016

Abstract Hardware trojans inserted in integrated circuits have received special attention of researchers. Most of the recent researches focus on detecting the presence of hardware trojans through various techniques like reverse engineering, test/verification methods and side-channel analysis (SCA). Previous works using SCA for trojan detection are based on power measurements, or even simulations. When using real silicon, the results are strongly biased by the process variations, the exact size of the trojan, and its location. In this paper, we propose a metric to measure the impact of these parameters. For the first time, we give the detection probability of a trojan as a function of its activity, even if untriggered. Moreover, we use electromagnetic field as side-channel, as it provides a better spatial and temporal resolution than power measurements. We conduct a proof of concept study using an AES-128 cryptographic core running on a set of 10 Virtex-5 FPGA. Our results show that, using this metric, there is a probability superior than 99 % with a false negative rate of 0.017 % to detect a HT bigger than 1 % of the original circuit.

Keywords Hardware Trojan (HT) · Trojan detection · False negative probability · Electromagnetic measurements · Side-channel analysis · Process variation · FPGA

✉ Xuan Thuy Ngo
xngo@enst.fr

[1] Institut MINES-TELECOM, TELECOM-ParisTech, CNRS LTCI (UMR 5141), Paris, France

[2] Secure-IC S.A.S, Cesson-Sévigné, France

[3] Temasek Laboratories, NTU, Singapore, Republic of Singapore

1 Introduction

The trust and security of integrated circuits (IC) design and fabrication is critical for sensitive fields like finance, health, and governmental communications. Due to the complexity and the high cost of IC fabrication cycle, more and more firms outsource their circuit. This trend gives a possibility to introduce malicious circuit, called Hardware Trojan (HT) in any IC. It can either perform a denial of service (DOS), steal sensitive information or deteriorate circuit performance [12, 13].

HT circuit is composed of two parts:

- *Trigger* used to activate the HT.
- *Payload* used to realize the malicious function.

It can be inserted at any point during the design or fabrication process from Register Transfer Level (RTL) to layout and circuit fabrication. In [7], authors show some techniques to insert HT at RTL level. These HT, which are activated with a specific pattern inputs, can leak secret key via RS232 channels. In [4], authors insert their HT at mask level (GDSII). This HT injects a fault in AES cipher to facilitate key recovery through fault attacks.

The HT, unlike software trojans, cannot be removed once it is fabricated. So it is important to detect these HT before they become effective. The detection can be done during the test phase, or at run-time. The amount of efforts required to detect an HT also varies according to the point of insertion. For instance, an HT is easier to detect at RTL or Netlist level if infected source codes are available, which is unlikely. Detection becomes a big challenge at GDSII and fabrication level.

Previous works classify detection methods into two wide categories: destructive and non-destructive. Destruc-

tive methods comprise destructive reverse engineering. This means that we destroy the chip to reconstruct GDSII and Netlist level of the chip using chemical products and optical materials as scanning optical microscopy (SOM), scanning electron microscopy (SEM), pico-second imaging circuit analysis (PICA), etc. The main advantage of this technique is its accuracy with which it can detect all malicious insertions. However, the destructive nature and lengthy times needed to reconstruct the netlist of the chip are a significant drawback. To avoid the destruction of ICs, certain non-destructive methods were introduced, which further can be classified as invasive and non-invasive. Invasive methods consist in modifying IC structures in design phase (RTL, Netlist levels) to prevent the HT insertions or to assist other detection techniques.For example in [2], authors propose a tool for protection of IP sent by a mechanism of key exchanges protecting both the customer and supplier.

Non-invasive methods compare the physical characteristics or logical state of an IC with a genuine circuit, also known as the "golden circuit", at testing time or run-time. In [1], authors propose to add re-configurable DEsign-For-ENabling-SEcurity (DEFENSE) logic to the functional design. In test time, the first approach is using logic testing. It involves applying test patterns at the input and try to detect abnormal behaviors of ICs [3]. But almost all test-time detection techniques are difficult to realize. They cannot ensure that HT will be activated because of the complexity of test patterns.

Side-channel analysis (SCA) can also be deployed to detect HTs by observing and comparing physical traits (example, power consumption, time delay, etc.) of an IC under test against a trusted IC named "golden circuit" [10, 11]. By comparing the detection methods, SCA seems to be one of the best approaches to detect HT because IC physical characteristics will be altered when the internals of an IC are modified. It can detect almost any kind of HT; however, it does need a trusted reference or "golden circuit" which may not always be available. Sometimes, destructive reverse engineering can be applied on a small set of IC to obtain a golden model and then use them to detect HT in other circuits with SCA.

To our knowledge, previous works on SCA-based detection have been limited to either simulations or power consumption measurements on some real circuit. In [8], authors present a practical evaluation of HT detection using SCA on FPGA. But the experiments were performed on a single FPGA, so the process variations were not taken into account. The placement and routing of original circuit in golden model and infected models are not the same in the experiment, which makes it hard to quantify the effect of HTs alone. In this paper, we propose a metric to evaluate the impact of process variations, HT sizes, and HT placements in HT detection. Process variations are studied on 10 different samples of Virtex 5 FPGA (LX30). We keep the same placement and routing of original circuit, using Xilinx FPGA Editor, in all circuit versions (golden and infected) to imitate the real scenario of HT insertion on ICs. Our work on HT detection is done using electromagnetic (EM) measurements.

The rest of this paper is organized as follows: Section 2 gives our metrics rationale. Sect. 3 presents the scenario of our study. It also shows which and how the HT was implemented in FPGA without changing placement and routing. Section 4 shows the results of our study. We give an estimation of the probability of detecting an HT as a function of its size. We also studied the impact of process variations and placements of HT on EM measurements. Section 5 gives a discussion about our results and SCA-based detection methods. We finish with a small conclusion and some perspectives in Sect. 6.

2 Metrics rationale

2.1 How to build a reference model

For a SCA-based detection method, it is essential to have a reference. This reference is constructed keeping certain physical conditions as constant or negligibly changed. These parameters are as follows:

- The temperature.
- Power supply voltage.
- The acquisition setup.

But even with this similar environment, the circuits are always sensitive to process variations (PV), thus deteriorating HT detection efficiency. Indeed two circuits fabricated with the same process, and in the same wafer, have slightly different physical and electrical behaviors. The process variation effect on EM measurement can be modeled by a random noise with a Gaussian distribution [5]. On the other hand, the impact of the HT comes mainly from the trigger block which generates an increase of consumption or EM activity. Therefore, the HT contribution to the side-channel (e.g., the EM field) can be modeled by an activity offset on a net used by the HT. This is illustrated on Fig. 1.

This figure illustrates the activity distribution at a specific sample time and for a set of reference devices. The black Gaussian curve is the activity distribution without any HT, while the gray Gaussian curve is the distribution through the same set of circuits which contains the infected AES (with HT). It is merely an offset which depends on the HT size, placement, and position relative to the probe in case of EM acquisition. Using this model, the probability of detecting a HT can be calculated. Precisely, we can estimate the false

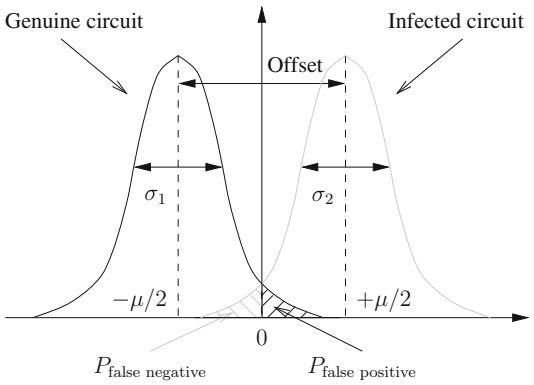

Fig. 1 EM field for genuine and HT infected circuit

positive and false negative probability of HT detection as a function of HT size for a set of ICs.

2.1.1 False negative probability computation theory

By definition, the false positive probability is the probability of rejecting a genuine circuit. It corresponds to the blue area in Fig. 1. The false negative probability is the probability to accept an infected circuit. It corresponds to the gray area in Fig. 1. These two distributions have the same standard deviation ($\sigma_1 \approx \sigma_2 = \sigma$) as the HT impact corresponds to a shift of the mean value. So the false negative probability is almost equal to false positive probability and only one needs to be calculated.

The Gaussian probability curve with a mean of $\frac{\mu}{2}$ and a standard deviation of σ is presented by the following equation:

$$P_0(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \times \exp - \frac{\left(x - \frac{\mu}{2}\right)^2}{2\sigma^2}. \tag{1}$$

Hence the false negative and false positive probability are calculated as follows:

$$P_{\text{false\_negative}} \approx P_{\text{false\_positive}} \tag{2}$$

$$P_{\text{false\_negative}} = \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi\sigma^2}} \times \exp - \frac{\left(x - \frac{\mu}{2}\right)^2}{2\sigma^2} \, \mathrm{d}x. \tag{3}$$

These equations can also be written as

$$P_{\text{false\_negative}} = \frac{1}{2} - \frac{1}{2} \cdot \operatorname{erf}\left(\frac{\mu}{2\sigma\sqrt{2}}\right), \tag{4}$$

In this equation, erf is error function defined as follows:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp\left(-t^2\right) \mathrm{d}t \ . \tag{5}$$

This equation can estimate the false positive and false negative probability of HT detection for each HT size and position. The results in the rest of the paper are computed assuming the aforementioned model for leakage. In [6], the authors also give the same proposition to evaluate the false positive and false negative probability. But they used delay measurements and simulations to perform the result. In this article, we apply this approach to the EM measurement which gives a better spatial and temporal resolution and we do not need to add extra circuit in the test circuit (as delay measurement circuit). Furthermore, we realized tests on a set of 10 FPGAs to evaluate the real impact of process variations on the SCA-based detection method.

3 Experimental setup

The proof-of-concept is done on FPGA setup based on Virtex-5 (LX30) FPGA from Xilinx. In the following, we present the measurement setup. Then, we provide details of the HT inserted along with the AES-128 crypto-processor for our study. Next we present the technique used to insert HT in an FPGA, without modifying the placement and routing of the original circuit.

FPGAs are hardware emulation platforms of ASICs. So, we model (at lower cost!) with an FPGA what we would need to do with an ASIC: send the tape-out database to an untrusted off-shore entity like a foundry, and insert a HT in a circuit before fabrication. Even the technology used for fabricating a ASIC is same as that for FPGA (CMOS process). The only difference lies in the tricks and techniques used to insert the HT without disturbing the circuit. In Xilinx, one can use circuit representation in FPGA Editor or XDL to achieve it. On the other hand, the techniques used in IC should be totally different, for instance, "ECO-placement" in Cadence Encounter. (ECO means "Engineering Change Order" and consists in applying small and local modifications to the layout to fix minor timing violations or to improve the yield before going to tape-out.) Nevertheless the result can still be extended from FPGAs to ASICs.

3.1 Measurement setup

The measurement setup is shown in Fig. 2. It is composed of the following:

- FF324 Virtex 5 experimental board with a ZIF socked that allows to change the device under test (DUT).
- Devices Under Test (DUT) are 10 Xilinx FPGA Virtex 5 (LX30) fabricated in 65 nm technology node.
- Langer RFU-5-2 probe that captures the global EM activity of the chip.
- 30 dB Langer EMV power amplifier to amplify EM signal come from probe.
- Agilent 54853A infiniium DSO configured at 5 GS/s.

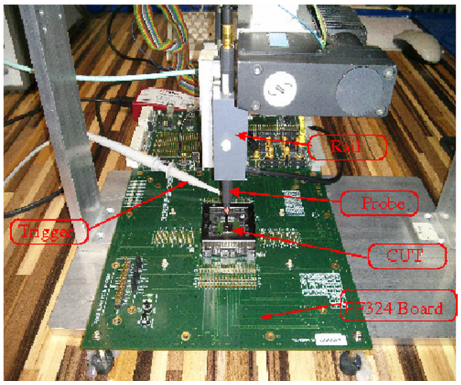

Fig. 2 Platform EM measurement

- A 3-D cartography table, where the probe is attached, that can be controlled by the computer so that to minimize experimental variations.
- Agilent E3631A stabilized power supply for the test board.

This measurement platform ensures the same probe position for all FPGAs under test, thus minimizing the experimental variations. All measurements were carried out with the same temperature condition. A script is used to trigger the FPGA and at the same time capture the trace on the oscilloscope which is then dumped in a file. The acquisition was done by script via computer. The goal of the experiment was to detect any significant abnormal activity without being able to trigger the added HT. During the measurements, the HT is never activated. In our scenario, the designer knows the architecture and the characteristics of the DUT and is able to simulate the behavior of the DUT. Each captured trace contains full activity of the cipher and the communication before and after the cipher. Once the traces are captured, we analyze them to select the interesting points that are used to detect any abnormal activity when a given node of the DUT is stimulated. The traces are acquired for random plaintexts, where each trace is averaged 1000 times by the oscilloscope to minimize the measurement noise. A plaintext which can activate HTs inserted is never used. EM probe was placed on the AES circuit. For each measurement, the position of EM probe is unchanged.

3.2 Trojan description

We synthesized and implemented a AES-128 crypto-processor on the FPGA along with a simple UART controller for communication with the outside world. This AES circuit needs one clock cycle to compute one round or 11 clocks for the whole encryption process. Figure 3 presents the HT inserted in AES. The HT Trigger is composed of 2 activation conditions:

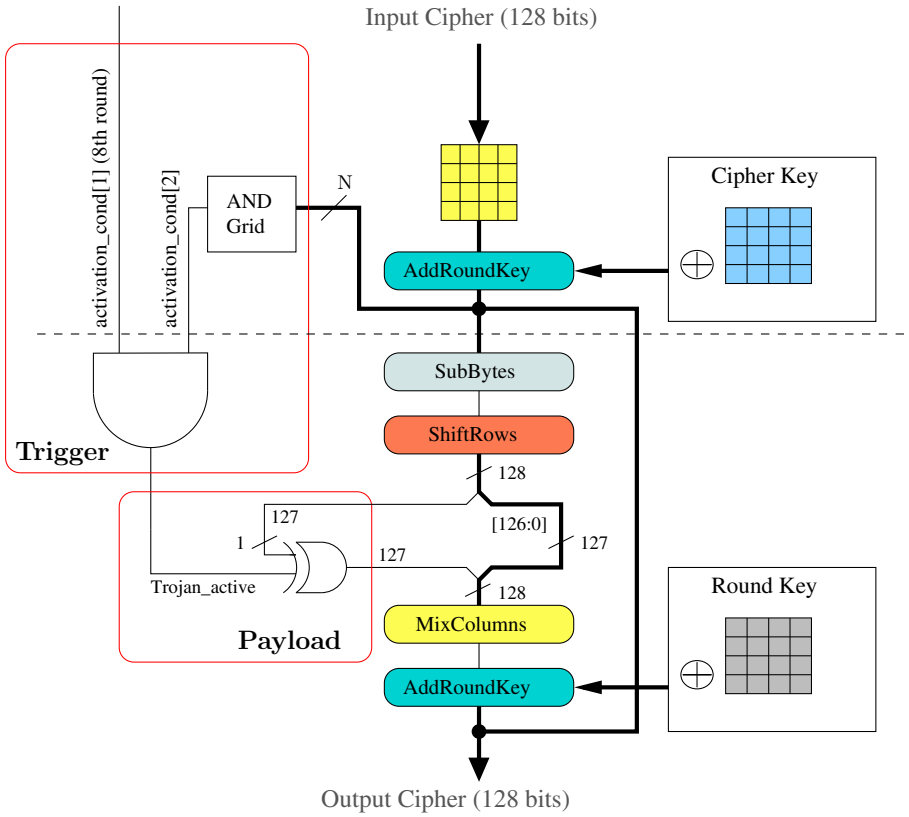

Fig. 3 Trojan inserted on AES 128 bits

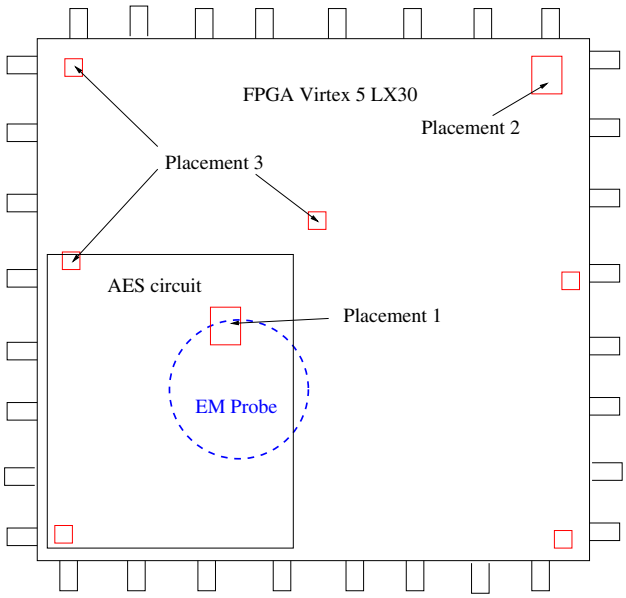

Fig. 4 Hardware trojan placements in FPGA

- First when the 8th computation round is operated, and
- second when N least significant bits (LSB) of 128 bits at the output of AddRoundKey are at "1".

The HT payload is an XOR gate that will inject a fault in the inner eighth round when HT is activated. This HT structure can aid the attacker to perform faults compliant with Piret's DFA [9] if HT is connected to the N least significant bits (LSB) of 128 bits of AddRoundkey output. We choose AddRoundkey output to impact the critical path of the circuit.

To evaluate the impact of HT sizes, we created 3 versions of the same HT, each time varying the generic parameter N as

- *Trojan 1* HT with the parameter $N = 32$, around 0.5 % of the original circuit.
- *Trojan 2* HT with the parameter $N = 64$, around 1 % of the original circuit.
- *Trojan 3* HT with the parameter $N = 128$, around 1.7 % of the original circuit.

We also study the impact of placement of the HT with respect to the original circuit, HT with the size of 1.7 % was placed in three different placements as shown in Fig. 4:

- *Placement 1* Trojan 3 placed within the boundary of AES crypto-processor.
- *Placement 2* Trojan 3 placed outside the boundary of AES crypto-processor in a far-off corner of the FPGA.
- *Placement 3* Trojan 3 placed outside the boundary of AES crypto-processor and dispersed over the FPGA.

In total, we created seven different designs: a HT-free design (golden circuit) and six infected versions with different sizes and placement.

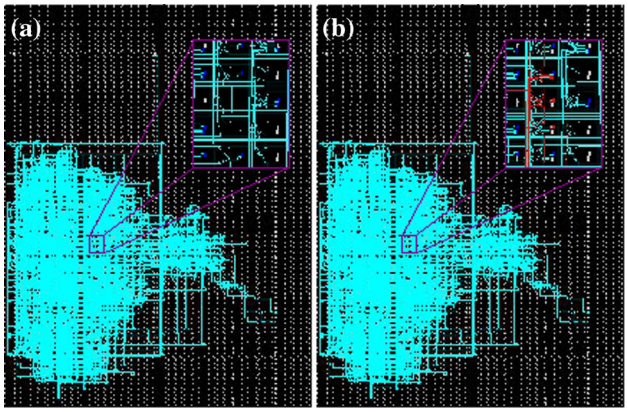

Fig. 5 Placement and routing of circuits on FPGA Virtex 5. for **a** AES 128 bit without HT and **b** AES 128 bit with HT 1.7 %

3.3 Trojan insertion

In order to evaluate the impact of HT in FPGAs, we need to keep the same placement and routing between the golden circuit and HT-infected circuit. Hence the only difference between the two is the logic utilized for implementing the HT logic. To insert a HT in a Virtex 5 FPGA, without modifying the routing, we follow the following steps:

1. Synthesize, Translate, Map, Place & Route the original circuit.
2. Extract the Native Circuit Description (NCD) file which contain the logic, placement & routing information of original circuit for golden model.
3. Open the NCD file using the FPGA Editor tool and insert HT in unused LUTs and Slices of FPGA, manually or by a script.
4. Generate bit files for both original and HT-infected model with FPGA Editor.

With this method we can ensure that the placement and routing of original circuit is the same in both golden and HT-infected circuit. Figure 5a shows the placement and routing of the original (golden) circuit and Fig. 5b is the placement and routing of infected circuit with a HT of size 1.7 % of the golden circuit. Trojan 1 needs 7 LUTs, Trojan 2 needs 14 LUTs, and Trojan 3 needs 27 LUTs. For each Trojan, the payload part needs only one LUT.

The following section presents the experimental results.

4 Results

In this section, we apply statistical tools to enable HT detection by side-channel measurement. A single EM trace is shown in Fig. 6. The circuit is clocked with a frequency of 24 MHz. It is known that EM measurements generally

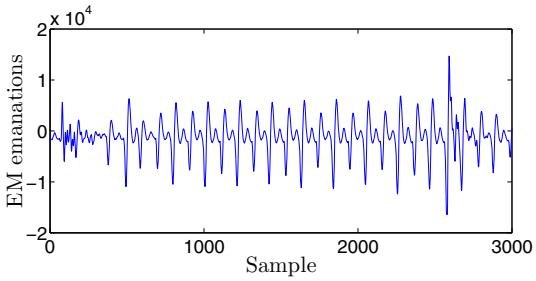

Fig. 6 EM measurement of a single AES-128 encryption

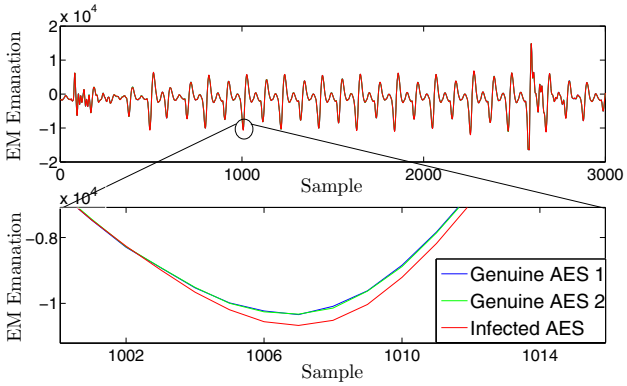

Fig. 7 Hardware trojan detection using averaged EM traces

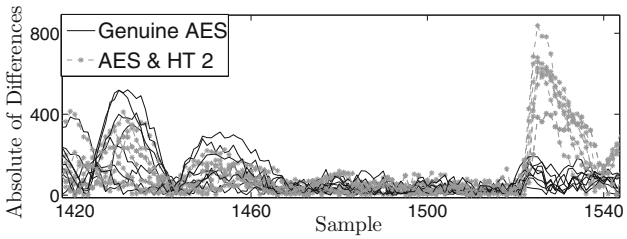

Fig. 8 Impact of process variations on EM measurements

possess a higher SNR than power measurements because of the directivity of the EM probes. EM probes can be spatially more selective and can be made to capture the activity of the stimulated region only.

4.1 HT detection using electromagnetic measurement

The genuine AES and the infected AES with combinational HT presented in Sect. 3.2 are used to evaluate this method. These two designs are implemented on FPGA Virtex 5 ($LX30$). The experiment is performed using the setup described in Sect. 3.1. The circuit is clocked with a frequency of 24 MHz. EM traces are acquired for random plaintexts, where each trace is averaged 1000 times by oscilloscope to minimize the measurement noise. A single EM trace is shown in Fig. 6. We noticed that the SNR seems good thanks to averaging done by oscilloscope. All the ten rounds of encryption can be distinctively seen in this trace.

Figure 7 shows three traces (two traces for the genuine AES design and one traces for infected AES with combinational HT design) for the same plaintext. The traces in black and blue are for genuine AES design taken at different moments with the same plaintext. That means we implement genuine AES design on the FPGA Virtex5 and acquire the first trace with plaintext $P1$. Then we turn off the setup, after we re-implement genuine AES on the same FPGA and we acquire the second trace with the same plaintext $P1$. It allows us to evaluate the measurement noise created by setup installation. Regarding Fig. 7, these two traces are nearly the same by averaging 1000 times with an oscilloscope. Therefore, setup noise is removed. The third trace is the one of infected AES design which is acquired with the same plaintext $P1$ as the two genuine AES traces. We noticed that the trace of infected AES is different compared with the genuine AES traces at some samples. This difference comes from HT insertion. Therefore, the HT can be easily detected by comparing directly the genuine AES traces and infected AES traces with the same plaintext. Notice that the plaintext is fixed but unknown and the HT is never activated during the experiments.

4.2 HT detection feasibility with process variations

In Fig. 8 we plotted the difference $|G_{j,1} - \mathbb{E}_{10}(G_1)|$ of all golden circuits inserted in 10 FPGA in black, and $|T_{10,1} - \mathbb{E}_{10}(G_1)|$ of HT test circuit in gray. Our notations are defined below:

- $G_{j,1}$ is the EM trace for the first plaintext of the jth golden circuit,
- $T_{10,1}$ is the EM trace for the first plaintext of the infected circuit inserted in FPGA number 10, and
- $\mathbb{E}_{10}(G_1)$ is the EM trace mean for first plaintext of all 10 golden circuits.

We can notice the static differences between black curves that are due to process variations. We also noticed that the difference of EM measurement in gray for AES with HT 1 (0.5 %) is bigger than the fluctuation of process variations in certain places thus allows the possibility to detect this HT. This clearly shows that the insertion of HT is detectable if we choose specific points of interest.

4.3 Hardware Trojan detection probability

4.3.1 Using the sum of EM differences (1st approach)

For this experiment, we implemented 4 different designs (one with a genuine AES and 3 other with AES infected by HT of size 0.5, 1, 1.7 %) on 10 FPGA Virtex 5. For each

implementation, we take 1 EM trace averaged 1000 times via oscilloscope for 1 random plaintext (note that it is the same plaintext for each acquisition). In total, we took 40 traces. After, we calculate the mean of genuine traces over 10 FPGA. Then we compute the absolute of differences between each trace with this mean trace.

$$Dg_i = \left| G_i - \underset{10}{\mathbb{E}}(G_j) \right|,$$
$$Dt_{s,i} = \left| T_{s,i} - \underset{10}{\mathbb{E}}(G_j) \right|.$$

In these equations, G_i is the traces of genuine design in FPGA i, $T_{s,i}$ is the trace of design infected by HT size s (for $s = 0.5, 1,$ *or* $1.7\,\%$) in FPGA i (for $1 \leq i \leq 10$), and $\underset{10}{\mathbb{E}}(G_j)$ is the mean of genuine traces. Logically, Dt must be bigger than Dg in certain points because of the HT contribution in EM leakage (see Fig. 8). Hence we computed the integral (sum) of these differences Dt & Dg to accumulate HT activity thanks to the following equations:

$$SoDg_i = \sum_{0}^{nb\_sample} Dg_i \tag{6}$$
$$SoDt_{s,i} = \sum_{0}^{nb\_sample} Dt_{s,i} \tag{7}$$

In this equation, $SoDg_i$ is the Sum Of Differences of genuine AES in FPGA i, $SoDt_{s,i}$ is the sum of differences of AES infected by HT size $s\%$ in FPGA i, and nb_sample is the total sample number of the trace (in the case study $nb\_sample = 3005$). These sums of differences are divided into four groups:

- *AES Genuine* contains $SoDg$ of genuine AES for 10 FPGAs.
- *AES & HT1* contains $SoDt_{0.5\,\%}$ of AES infected by HT1 for 10 FPGAs.
- *AES & HT2* contains $SoDt_{1\,\%}$ of AES infected by HT2 for 10 FPGAs.
- *AES & HT3* contains $SoDt_{1.7\,\%}$ of AES infected by HT2 for 10 FPGAs.

Each group has 10 values corresponding to 10 FPGAs.

Figure 9 shows the boxplot for each group. On each box, the central mark is the median and the edges of the box are the 25th and 75th percentiles. The mean and standard deviation of each group are

- *AES Genuine* $\mu = 3.25 \times 10^5$ and $\sigma = 6.9 \times 10^4$.
- *AES & HT1* $\mu = 3.7 \times 10^5$ and $\sigma = 9.1 \times 10^4$.
- *AES & HT2* $\mu = 3.9 \times 10^5$ and $\sigma = 4.1 \times 10^4$.
- *AES & HT3* $\mu = 5.27 \times 10^5$ and $\sigma = 6.27 \times 10^4$.

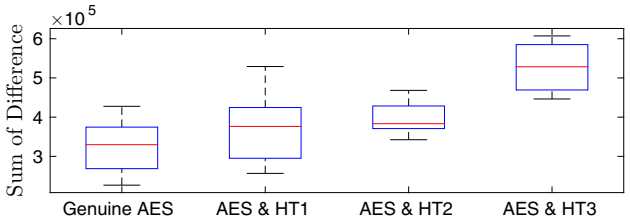

Fig. 9 Sum of absolute differences (SoD) Dg and Dt for the four implementations

Table 1 False negative detection probability

	HT 1 (0.5 %)	HT 2 (1 %)	HT 3 (1.7 %)
1st approach	43	34	9
2nd approach	26	17	5
3rd approach	24	0.017	0.011

Then we apply the Eq. (4) to calculate the false negative probability for different HT sizes using these 4 groups. The result of this first approach is presented in Table 1.

We noticed that the false negative probability decreases when the HT size increase. It is logical because the size of HT has a direct impact on the EM leakage. Bigger HT are easier to detect. We also noticed that there is a 9 % probability that we fail to detect HT that is 1.7 % of the original design, with this first approach. It is because of process variations in EM measurements. In order to improve the detection probability, we use another approach, named "Local Maximas" and "Threshold Technique".

4.3.2 Using local maximas (2nd approach)

The principle of this approach is simple: we sum only the local maximas of the absolute differences Dg_i and $Dt_{s,i}$. It allows to reduce the error and improve the false negative results. The computation of the 2nd approach is the following:

- Calculate Dg_i and $Dt_{s,i}$ as presented in Sect. 4.3.1.
- Find the local maximas of Dg_i and $Dt_{s,i}$.
- Sum the local maximas of each Dg_i and $Dt_{s,i}$.
- Compute the false negative on the sum of local maximas.

Table 1 presents the improvement of this approach. The false negative rates of HT 0.5, 1 and 1.7 % are, respectively, 26, 17, and 5 %. So the results are twice better using local maxima values.

4.3.3 Using threshold technique (3rd approach)

This approach comes from the observation in the Sect. 4.2 :The difference of EM Measurement (in gray) for AES with

HT 1 (0.5 %) is bigger than the fluctuation of process variations in certain places. Therefore, it will be easier to detect HT if we choose specific points of interest. By using the threshold technique, we can select these points to improve the detection probability. For this second approach, we first calculate Dt and Dg as in the previous section, after which we apply the threshold technique for theses differences as the Algorithm 1:

Algorithm 1 Threshold technique

```
for k from 1 to Nb_of_FPGA do
  for j from 1 to Nb_of_Sample_point do
    if Dt_{k,j} > max(Dg_{i,j}) with i varies from 1 to Nb_of_FPGA then
      Dt_{k,j} = Dt_{k,j}
    else
      Dt_{k,j} = 0
    end if
  end for
end for
```

The threshold is fixed automatically at each point of the trace using the golden traces. After applying the threshold, we conserve only the interesting points which are the points around of clock edges in our case study. This is logical because most of the activities of AES occur during the clock edges; hence the contribution of HT is maximum at these points. And all uninteresting points will be put to zero. Next, we recalculate the sum of theses new differences using the Eqs. (6), (7) and also the false negative probability using the Eq. (4).

The result of threshold technique is also presented in Table 1.

We notice that the false negative probability of HT 0.5 % is 24 %. But the false negative probability of HT 1 and 1.7 % is 0.017 and 0.011 % respectively. Thus they are easily detected. So this approach improved significantly the HT detection probability.

We also apply this approach to the mean of a 20 traces corresponding to 20 random plaintext of each implementation. For HT with size of 0.5, 1 and 1.7 % are, respectively, the false detection probability is 31, 0.19 and 0.59 %. It is a bit worse than the previous results using only 1 plaintext. The higher false detection probability for HT of size 1.7 % is a statistical artifact as we only use ten samples of FPGA. But it could be helpful to detect sequential HTs which survey a set of plaintext.

4.4 Impact of hardware Trojan placement

One drawback of the global EM measurement is that EM probe is very sensitive. The probe position will directly affect the result. So EM signals can also be different for a different placement of HT. In order to take into account this problem, we try to place the same HT in different positions and show its impacts on EM measurement. For this experiment, we place Trojan 3 in different places on the FPGA number 10 Sect. 3.2. First, HT is placed inside the AES core. Second, it is placed in the right-upper corner of the FPGA. And last, we split Trojan 3 into 6 different parts and disperse them in the FPGA.

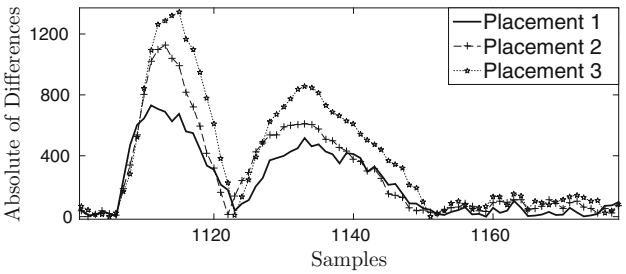

Fig. 10 Impact on EM measurement for different locations of Trojan 1.7 % on FPGA Virtex 5

Figure 10 gives the results of these experiments. Each curve shows the difference between the average of all AES-infected circuit traces and the average of all AES genuine circuit traces. We noticed that EM measurement is practically the same for placement 1 and placement 2. For the placement 3 where HT was dispersed, the results are better. That is because of long routing lines which connect AES to HT and also power buffers to drive the logic region where HT is placed. In ASIC application, we could expect to see the same phenomenon because of long HT routing lines. Theses lines can have an antenna effect and hence amplify the EM leakage. For high-speed application, the long HT lines can create problem for HT working because of a big RC. Hence, the attacker must insert additional buffers to ensure that HT works correctly with the same frequency as the original circuit. These additional buffers will also amplify the EM leakage signal and hence HT can be detected easily. So the placement and routing of HT have a little influence on the result performance.

5 Discussion

This paper shows the probability of detecting a HT as a function of its size. We noticed the influence of process variations in our results. In our case study, the HTs used are combinational in nature and we can easily detect them if its size is bigger than 1 %. For a sequential HT, we estimate to have the similar results because sequential HT is difficult to activate, but its contribution on SCA would be the same as combinational HT.

Our method does not take into account linear dependencies between successive points of side-channel traces.

Therefore, it could lead to a bias in the Gaussian distribution when summing these successive points. But it can be solved by summing only the local maxima detected from absolute of differences on genuine traces.

We also noticed that the trigger of HT is very big compared to its payload. Thus, even if the HT is never activated, there is always the contribution of its trigger to the global consumption. Even if the size of the HT is as small as 1 % of the actual AES circuit, its activity is bigger compared to the activity of AES at few time instants. For HTs, which connect directly to the critical paths (at the outputs of Addroundkey in our case), the activity of HT trigger part could be important even if the payload of the HT is always inactive. In such cases, HTs using the "rare" nets (nets with a very small activity) to activate can reduce the detection probability.

This paper also gives the probability of detecting a HT with the size bigger than 1 % compared to an AES circuit. The proportion of this HT will be smaller for a bigger complex circuit. But we can do a cartography on the circuit and hence detect the HT by repeating the same procedure but only a selected zone of cartography. Regarding the impact of HT placement, we noticed that attacker must make the HT trigger as compact as possible. Also, it is in the interest of the attacker to minimize the use of the routing lines to minimize the contribution of HT in the global or local activity in EM measurement. In any case, we were able to successfully detect a HT bigger than 1 %, irrespective of its placement. Using EM measurement, we can select a local area where the HT is located, thus increasing the contribution of HT which is not the case of global power measurement.

6 Conclusion and perspectives

In this paper, we studied the detection of HT implementation on FPGAs, without changing the placement and routing of the original circuit. This represents a proof of concept study for ASIC circuit where *HTs are added by untrusted chip manufacturers before the final fabrication of the chip. First of all, we demonstrated the power of EM measurements in the detection of HT by side-channel techniques. Next, we tested HT of different sizes to estimate the detection probability as a function of its size. We repeated our experiments on 10 different FPGA of the same reference (Xilinx $LX30$) to study the impact of process variations on this detection probability. The impact of placement of the HT is also studied, which has very little influence of the detection probability. We then introduce a novel metric to detect HT by exploiting side-channel techniques. This metric uses integration of the absolute difference of the captured traces. It predicts the probability of HT detection with a determined false positive and false negative rates. Our results show that, using this metric, there is a probability superior than 99 % with a false negative rate of 0.017 % to detect a HT bigger than 1 % of the original circuit.

Further extension of this work can include a more precise evaluation of the impact of process variations on detection probability. This precision can be achieved by conducting the same experiments on n FPGAs, where $n \gg 10$.

References

1. Abramovici, M., Bradley, P.: Integrated circuit security: new threats and solutions. In: Sheldon, F.T., Peterson, G., Krings, A.W., Abercrombie, R.K., Mili, A. (eds) CSIIRW, pp. 55. ACM (2009)
2. Alkabani, Y., Koushanfar, F.: Active hardware metering for intellectual property protection and security. In: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, SS'07, pp. 20:1–20:16. USENIX Association, Berkeley (2007)
3. Banga, M., Hsiao, M.S.: ODETTE: a non-scan design-for-test methodology for Trojan detection in ICs. In: International Workshop on Hardware-Oriented Security and Trust (HOST), IEEE, pp. 18–23 (2011)
4. Bhasin, S., Danger, J.L., Guilley, S., Ngo, T., Sauvage, L.: Hardware Trojan horses in cryptographic IP cores. In: FDTC, pp. 15–29, August 20, Santa Barbara, CA, USA (2013)
5. Bowman, K.A., Duvall, S.G., Meindl, J.D.: Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration. IEEE J. Solid-State Circuits **37**(2), 183–190 (2002)
6. Cha, B., Gupta, S.K.: Efficient trojan detection via calibration of process variations. In: 2012 IEEE 21st Asian Test Symposium (ATS), pp. 355–361 (2012)
7. Jin, Y., Kupp, N., Makris, Y.: Experiences in hardware Trojan design and implementation. In: Proceedings of the 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, HOST '09, pp. 50–57. IEEE Computer Society, Washington DC (2009)
8. Kutzner, S., Poschmann, A.Y., Stöttinger, M.: Hardware trojan design and detection: a practical evaluation. In: Proceedings of the Workshop on Embedded Systems Security, WESS '13, pp. 1:1–1:9. ACM, New York (2013)
9. Piret, G., Quisquater, J.J.: A differential fault attack technique against SPN structures, with application to the AES and KHAZAD. In: CHES, LNCS, vol. 2779, pp. 77–88. Springer, Cologne (2003)
10. Potkonjak, M., Nahapetian, A., Nelson, M., Massey, T.: Hardware trojan horse detection using gate-level characterization. In: DAC, pp. 688–693. ACM (2009)
11. Rad, R., Plusquellic, J., Tehranipoor, M.: Sensitivity analysis to hardware Trojans using power supply transient signals. In: Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, HST '08, pp. 3–7. IEEE Computer Society, Washington, DC (2008)
12. Skorobogatov, S., Woods, C.: Breakthrough silicon scanning discovers backdoor in military chip. In: Proceedings of the 14th International Conference on Cryptographic Hardware and Embedded Systems, CHES'12, pp. 23–40. Springer, Berlin, Heidelberg (2012)
13. U.S. Department Of Defense. Defense science board task force on high performance microchip supply. http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf