An On-Chip Technique to Detect Hardware Trojans and Assist Counterfeit Identification

Maxime Lecomte, Jacques Jean-Alain Fournier, Philippe Maurine

An On-Chip Technique to Detect Hardware Trojans and Assist Counterfeit Identification

Maxime Lecomte, Jacques Fournier, and Philippe Maurine

***Abstract*— This paper introduces an embedded solution for the detection of hardware trojans (HTs) and counterfeits. The proposed method, which considers that HTs are necessarily inserted on production lots and not on a single device, is based on the fingerprinting of the static distribution of the supply voltage (V_{dd}) over the whole surface of an integrated circuit. The measurement of this fingerprint is done through an array of sensors sensitive to the local V_{dd} value and fingerprint extraction is based on a novel variation model of CMOS logic performance. This model takes into account not only process variations but also the impact of the design (layout, supply routing, and so on). In addition to the fingerprinting process, this paper introduces an adaptive distinguisher to deal with the difficult problem of fixing the p-value on large sets of statistical tests. The efficiency of the whole detection methodology is experimentally demonstrated on a set of 24 FPGA boards.**

***Index Terms*— Electromagnetic measurements, field programmable gate arrays, hardware Trojan (HT), process variation, rings oscillators, side-channel analysis.**

I. Introduction

BECAUSE of the globalization of their manufacturing process, integrated circuits (ICs) have become increasingly vulnerable to malicious alterations. An adversary can modify an IC from the specification step up to the packaging stage. This threat raises concerns as ICs are used in a wide variety of critical applications. This kind of malicious alteration, called a hardware trojan (HT) insertion, can have different effects, which can be parametric (which, for example, reduces the IC's performances) or functional (which can leak sensitive data or cause a denial of service) [1].

An HT is composed of two parts: 1) a trigger and 2) a payload. The trigger is the mechanism that scans a few signals within the IC until a specific condition is met. When this condition is met, the payload is activated. The trigger can either be generated externally (e.g., external signals or a physical condition) or internally (a special internal state, data, etc). Moreover, the trigger can either be combinational (result of a logical operation) or sequential (related to a succession of states). The payload is the "malicious" effect of the HT. The payload can be explicit when signals are directly added, removed, or deactivated. The payload can also be implicit when the effect cannot be directly observed like, for example, leaking sensitive information through side channels like the power consumption. The detection of an HT before its activation is a difficult task and it still remains a challenging problem even after its activation when the payload is implicit.

Manuscript received July 8, 2016; revised September 16, 2016 and November 2, 2016; accepted November 5, 2016.

M. Lecomte is with CEA-Tech, 13541 Gardanne, France (e-mail: maxime.lecomte@cea.fr).

J. Fournier is with CEA Leti, Grenoble 38054, France (e-mail: jacques.fournier@cea.fr).

P. Maurine is with LIRMM, 34090 Montpellier, France (e-mail: philippe.maurine@lirmm.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2016.2627525

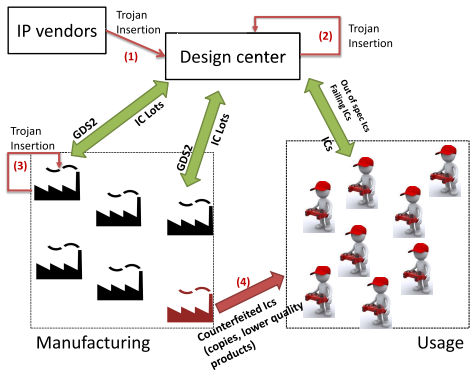

Fig. 1. Main threats to IC integrity.

A. Threats to IC Integrity

Fig. 1 summarizes the different steps, from design to exploitation, for manufacturing an IC and the associated threats on IC integrity.

The first vulnerabilities are at the design stage. A corrupted piece of hardware can be introduced into the product (Threat 1) or a rogue designer can introduce an HT into the HDL description (Threat 2). It is difficult to protect against such threats, but some solutions based on ad hoc design and verification methods have been proposed [2], [3].

The second vulnerable stage is the manufacturing (Threat 3). For example, filler cells can be substituted by logic gates inducing a denial of service or more complex functionalities, or a fuse can be disabled, and so on. A last threat is that of counterfeit. A taxonomy on counterfeits is given in [4]. It consists in selling second hand products, lower quality devices, or functional copies directly onto the market causing potential financial losses (Threat 4) for both the device manufacturer and the end user. Some can even be almost perfect copies that are extremely difficult to detect.

B. Background

The probability of triggering and thus detecting an HT during functional tests is low. As a result, testing is

an expensive approach to that end with no final guarantee about the integrity of the devices under test. The sole technique offering a high confidence level is reverse engineering. However, inspecting the circuit through reverse engineering is an expensive process in terms of cost and time and can be destructive. This solution can therefore be applied to only a few devices, even though some latest imaging-based methods could offer simpler and faster promising alternatives [5].

Several nondestructive methods for HT detection have been proposed recently. The first proposed approaches analyze, using statistical means, the overall power consumption of an IC to detect HTs. In [6], a detection technique based on the Karhunen–Loève theorem is proposed in order to detect the power consumption of the HT within process variations. However, this paper reports only validations obtained by simulations, omitting things like the measurement noise. In order to enhance the detection capabilities, other techniques have been proposed in [7] to locally analyze the propagation delays of logical paths with embedded monitors. However, once again, only simulation results are provided.

Later, [8] proposed to integrate a hardware system to monitor the critical wires of an IC. However, little information is given about the efficiency of this technique. In parallel, a test solution was proposed in [9]. It aims at easing the triggering of an HT or at least increasing its electrical activity.

In [10], an attempt to suppress process variations has been proposed based on the strong correlation among the maximum operating frequency of ICs, $F_{\max}$, and their dynamic power consumption. This approach, however, faces the difficult problem of measuring $F_{\max}$ [11].

Then in 2011, the use of ring oscillators (ROs) has been proposed to detect HTs. Lamech *et al.* [12] conducted an analysis of RO sensitivity to the presence of an HT and concluded that it was difficult to detect really small HTs. At the same time, Zhang and Tehranipoor [13] proposed the use of an array of RO, used in conjunction with a principal component analysis [14], to distinguish infected ICs from genuine ones. This proposal has been validated on FPGAs using a digital sampling oscilloscope (DSO) to measure the oscillation frequency of the ROs. This idea has then be applied to design an ASIC [15], but the results formerly obtained with FPGAs have only been partly validated. This could be explained by the fact that an embedded 8-bit counter was used to measure the oscillation frequency of the RO, which significantly lowered the measurement accuracy.

Later, Cao *et al.* [16] proposed to cluster, during the design step, the power grid in several voltage islands embedding each a dedicated sensor to enhance the detection capability. However, no experimental result is given. Soll *et al.* [17] described a method based on the use of near-field electromagnetic (EM) cartography. They concluded that it seems difficult to detect all HTs. However, Balasch *et al.* [18] proposed a more efficient technique to interpret the EM traces. As a result, they concluded that it is possible to detect really small HTs but with special care to control the temperature during measurements. Finally, Thuy *et al.* [19] analyzed EM emanations from FPGAs and succeeded in differentiating a genuine population from an infected one.

C. Contributions

Based on the above considerations, on-chip monitoring solutions seem relevant in terms of efficiency since the resulting detections rates are higher than for off-chip methods. Furthermore, these solutions seem industrially viable since the cost of the equipment, dedicated to data acquisition, is reduced as the tests can be done in parallel. For those reasons, this paper introduces an on-chip detection method.

In addition, we consider that the infection of a single device is not realistic because of the current life cycle of ICs. That is why the HT detection methodology proposed in this paper does not aim at establishing if an IC is infected but aims at checking the integrity of a whole production lot. Moreover, it should be noted that the proposed approach also allows determining if an IC is a rough counterfeit or if it is new or old, i.e., if it is potentially a reused IC.

The principle behind this methodology is to detect, thanks to an embedded sensor network, an eventual alteration of the inner structure (the presence of an HT), a modification of its floorplan (rough counterfeit), or a degradation induced by the aging effect (reused IC). These alterations modify the IC power distribution and in particular the static voltage drops [20] in the glue logic and hence that in the sensor array.

The proposed method also exploits a novel variation model of the performance of CMOS structures in real designs (not in test chips dedicated to the fine measurement of the intradie and interdie variations), a model that is introduced and validated in this paper.

Finally, this paper aims at introducing an adaptive distinguisher with a heavily reduced false positive rate and a high detection capability. This distinguisher is dedicated to the detection of stealthy HTs, i.e., HTs having a reduced spatial impact (small power consumption and/ or small size) or a reduced impact in time (their trigger consumes power only a short time) or both. This adaptive distinguisher can be applied to results from time-domain side channel analysis (e.g., EM analysis) or spatial analysis based on an embedded sensor array.

The rest of this paper is organized as follows. Section II details our model of HT infection and characterizes the passive and dynamic impacts of HTs on the supply voltage. Based on these results, Section III details the proposed methodology to detect HTs and potential counterfeits. Section IV describes the experimental results validating the proposed approach as well as the proposed variation model on a set of 24 FPGAs. Finally, the perspectives generated by our approach are discussed and a conclusion is given in Sections V and VI, respectively.

II. Hardware Trojan Emulation and Impacts

The insertion of additional logic into an IC and therefore that of an HT modify its inner structure and thus the static distribution of V_{dd} in the power/ground networks even if this additional logic remains at rest. The first consequence of this constitutes the static effect of an HT insertion. As a second effect is that when activated, this malicious logic increases the dynamic power consumption depending on the number of switching bits. This is the dynamic effect of an HT insertion.

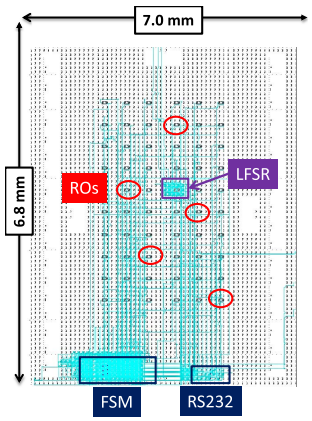

Fig. 2. Floorplan of the characterization test chip.

A. Measurement Setup

In order to precisely analyze the static and dynamic impacts of an HT insertion, a design was implemented on a Xilinx Spartan-3E 1600E FPGA [21] using Xilinx tools. This design includes the following:

1) a finite-state machine (FSM);
2) a serial communication block (RS232) that handles the communication between the computer and the chip;
3) a network of 60 ROs.

The ROs are used as on-chip sensors to locally monitor the internal supply voltage across the IC's surface. The FSM and the RS232 are placed far enough from the ROs so as not to influence them. The floorplan of the resulting design is shown in Fig. 2. One could observe that the 60 ROs form a 6×10 matrix. This placement was adopted to allow a spatial analysis of the static and dynamic impacts of an HT on the inner distribution of V_{dd}. The 60 ROs have exactly the same design. They are composed of four inverters and of a NAND2 gate in order to enable/disable each RO separately. The oscillation period ($\mathrm{T_i}$) of ROs i is around 150 MHz (6.667 ns) depending on the local quality of the process.

To improve the quality of the mesurement of the oscillation frequency (done through an output pin of the FPGA), each RO is connected to a clock divider that allows a cleaner measurement of $\mathrm{T_i}/2$ and hence $\mathrm{T_i}$. These measurements are done using a DSO with a sample rate of 40GS/s (25 ps between two samples). In order to get an accuracy of ± 0.25 ps on the $\mathrm{F_{RO}}/2$ measurements, we measured 100 oscillations of $\mathrm{T_i}/2$ and we repeated the process 1000 times to get, by averaging, a precise estimate of $\mathrm{T_i}$ but also of the standard deviation σ_{Ti}. To ensure a stabilized voltage, a dc power supply with a 0.05 % accuracy was used to directly power the FPGA core.

B. Measurement Accuracy

Prior to any analysis, we quantified the accuracy (including all sources of measurement noise: circuit, environment, power supply, etc.) of our measurements and the impact of intradie and interdie variations. To that end, we measured, on several boards, the mean value $\mathrm{T_i}$ and the standard deviation σ_{Ti} of each RO. Fig. 3 illustrates the precision of our measurements. It gives the 60 standard deviations obtained on two boards. As shown, σ_{Ti} values range between 12.2 and 13.4 ps.

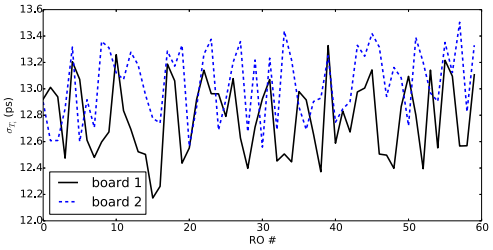

Fig. 3. Standard deviations $\sigma_{\mathrm{T_i}}$ of the 60 ROs for two different boards.

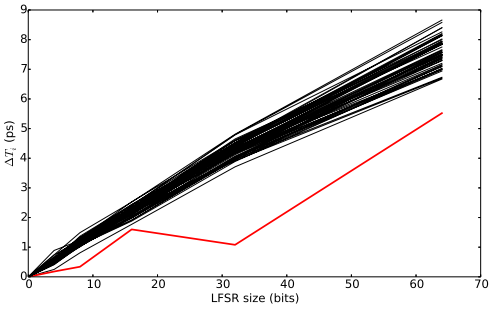

Fig. 4. Evolution of the ΔT_i with respect to the amplitude of the parasitic switching activity.

C. Hardware Trojan Implementation

To emulate an HT, a linear feedback shift register (LFSR) was placed all around the RO27 as shown in Fig. 12. This LFSR was designed in order to be able to control its word size that ranges between 4 and 64 bits. This was done to mimic a dynamic impact equivalent to the flipping of 2, 8, 16, and 32 D flip-flops (DFFs) (bits). Moreover, the LFSR was clock gated in order to be able to only observe the impact of the additional leaves of the clock tree on the IC behavior, i.e., while there is no switching activity of the LFSR itself.

With such features, this LFSR thus allows us to simulate the dynamic impact of any sequential HT as well as the static effect of any combinational or sequential HT. Of course, such a structure does not cover all potential HTs that an adversary can imagine, but at least all those made up of additional logic gates.

D. Dynamic Impact of an HT

Fig. 4 shows the difference (with and without activity), ΔT_i, between the periods of the 60 ROs with respect to the amplitude of the parasitic (LFSR) switching activity. The periods linearly increase with the number of switching bits in the LFSR. However, among those 60 curves, the red one in Fig. 4 shows a specific and unexplained behavior. It corresponds to the ΔT_{27} observed for RO27, which is located inside the LFSR.

The observed increase ranges between 0 and 7 ps for most ROs. This corresponds to only 0.048% of T_i. This value is

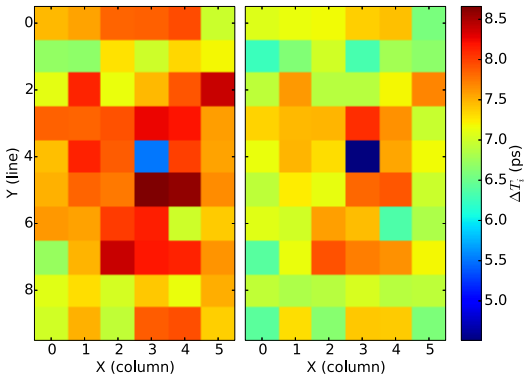

Fig. 5. Map of ΔT_i with a 64-bit LFSR for the two boards.

a very small compared with the effect of process variations, even intradie ones, whose effect can reach 500 ps on these boards.

E. Spatial Spread of the Dynamic Impact

In order to analyze what happens on RO's period regarding the distance from the parasitic switching activity, the spatial distribution of the dynamic impact was also characterized. To that end, the LFSR, configured to work with 64-bit words, was mapped around RO27 placed at $(X, Y) = (3, 4)$. Then all ΔT_i were measured. This was done for several boards.

Fig. 5 shows two cartographies of ΔT_i obtained with two different Spartan3E-1600 devices. Each point on these cartographies is an RO with 24 slices between two ROs. It was estimated that each slice has a height equal to $\sim 120 \mu$m according to the number of slices (along X and Y) embedded in these devices and to the device dimensions (measured using X-rays without removal of the package).

Fig. 5 shows that the parasitic switching activity induces an increase of RO27 period of 5 ps. This value is the minimal one over the whole IC's surface for both cartographies. This is a surprising and unexplained result. Indeed, the RO27 is surrounded by the LFSR and one would have expected a maximal dynamic impact at this location. This latter, 8 ps is rather observed for the ROs really close to RO27, while for the furthest ones, the period increase is equal to 6 ps.

Nevertheless, these results also suggest that the effect of the switching of 32 DFFs on the frequencies is extremely small for the ROs (5 ps) furthest from the parasitic switching activity compared with that of the closest ones (8 ps). Thus, one may conclude that the induced voltage perturbation is global and with a very small amplitude. This is reassuring since these results reflect the fact that the power/ground networks are designed to be as little resistive as possible in order to avoid the occurrence of important voltage drops that can compromise the timing constraints.

Another interesting point highlighted by these experimental results is that the influence of the parasitic switching activity on the periods (ΔT_i) is relatively constant at a given distance from the parasitic activity. This means that its effect seems to be *relatively* independent of the intradie variations.

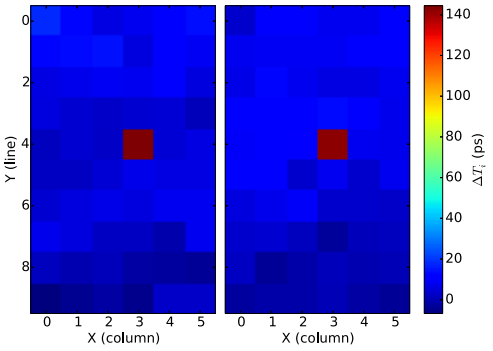

Fig. 6. Static impact of an HT insertion for boards 1 and 2.

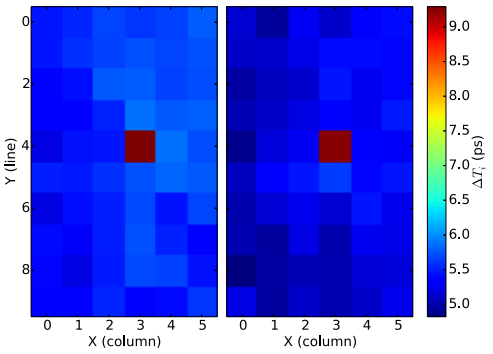

Fig. 7. Impact of activating the LFSR for boards 1 and 2.

F. Static Impact of an HT

Until now, we studied the impact of the parasitic switching activity of an HT, i.e., we observed the effect on ROs of an LFSR that was mapped into the FPGA, according to the word size it manipulates. To characterize the static impact of an HT insertion, we measured the impact of its implementation itself, i.e., of the procedure of mapping or not the LFSR, the LFSR being at rest (clock off). We then measured the effect of activating or not the LFSR's clock.

Fig. 6 shows the impact of integrating or not the LFSR on the RO period for two FPGA boards, while Fig. 7 shows the impact of activating or not the LFSR.

As shown in Fig. 7, activating the 64-bits LFSR induces a global increase of T_i of 5 ps and a local increase (only the infection point is affected) of 9 ps. This is coherent with what was previously observed.

Fig. 6 shows that the LFSR implementation has a global static impact of 20 ps and a static impact of 150 ps at the infection's location. The implementation of the HT is therefore the main source of modification in the V_{dd} distribution within the IC, but its impact is very local. Indeed, only the behavior of one RO has been strongly modified by the implementation.

This is an important result. Indeed, it suggests that implementing an HT induces significant modifications of the local characteristics of the power/ground networks (R, C, and local static current), modifications that could result in a modification of the local V_{dd} value and thus of timing performances of the surrounding logic. In addition, these results suggest that transient switching currents have less impact at a clock

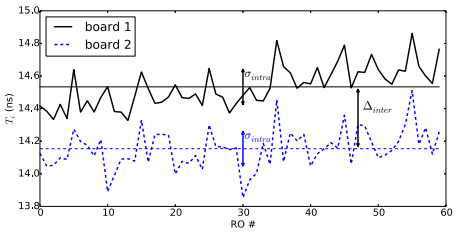

Fig. 8. 60 period values measured on two boards.

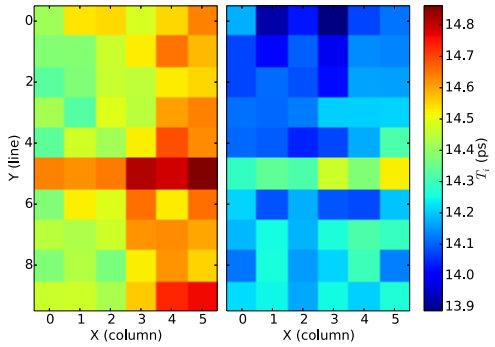

Fig. 9. Intradie variation map board 1.

frequency of 50 MHz. This latter observation should be confirmed for higher clock frequencies. These observations are the basis of the detection methodology introduced in this paper.

G. Process Variations

After having studied the static impact of an HT implementation and that of its parasitic switching activity, the influence of the process variations on the RO periods was studied.

The classical model of process variations classifies them into two categories: interdie variations and intradie variations. The interdie variations are the differences in terms of process quality between ICs and thus between their performance. Intradie variations denote the physical and electrical differences between elementary structures (interconnects, transistors, and so on) within a same circuit.

We estimated the impact of interdie and intradie variations from our lot of 24 FPGA boards. Fig. 8 gives the periods of the 60 ROs for two different boards. Therefore, it simultaneously shows the impacts of interdie and intradie variations. One can observe that there are significant interdie variations. In this case, they are responsible of changes of more than 500 ps between the two ICs. It also indicates that intradie variations are responsible for changes larger than 200 ps within each IC.

Fig. 9 gathers two maps giving the oscillation period of each RO for the two considered boards. A link between the location of an RO over the IC surface and its oscillation period clearly appears. This link could be explained by the impact of the power grid, i.e., on how the IC is designed. However, process variations have a larger impact. Indeed, the drawing of the two cumulative density distributions of the period and Fig. 8 show that they are nearly fully disjointed.

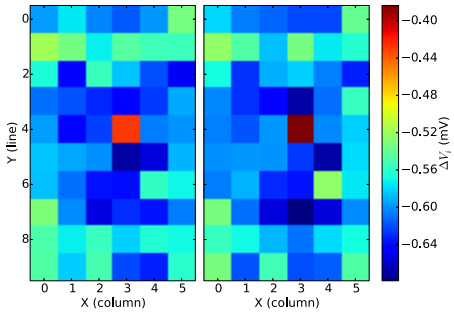

Fig. 10. Maps of the voltage drops induced by the activation of a 64-bit LFSR.

H. Discussion

In the preceding paragraphs, the characterization results have indicated the importance of the power/ground distribution. First, these experimental results have suggested that the main impact of an HT insertion is a local and static alteration of V_{dd}. Second, during the characterization of the impact of process variations, it has been observed that the placement of the RO with respect to the power/ground distribution could have an impact on their performance. All these observations encouraged us to finely analyze the impact of an HT insertion on the inner V_{dd} distribution.

1) IR Drop Cartographies: In order to sketch cartographies of the inner V_{dd}, in the presence or not of an HT, we characterized the sensitivity of the oscillation frequency to V_{dd} (the supply voltage value) of each RO in the design. This was done with a dc power supply for a voltage range of [1.19 V, 1.21 V], considering that the circuit's nominal voltage supply is 1.2 V.

As expected, it has been found that the period decreases linearly with V_{dd} on this small voltage range. From there, we applied a linear regression on the results from each RO to obtain the function $T_i = f(V_{\mathrm{dd}})$ for each RO. It was found that $(\Delta F/\Delta V_{\mathrm{dd}})$ is nearly the same for all the RO and equal to $(-1.2\cdot\ \mathrm{ps\,mV^{-1}})$. From there, we translate all formerly observed oscillation period changes into voltage drops.

Fig. 10 gives, for two FPGA boards, the maps of the V_{dd} drops induced by the activation of an implemented LFSR (working with words of 64 bits). It shows that the maximum voltage drop induced by the LFSR switching activity appears around the LFSR location. At this location, the voltage drop reaches 0.66 mV. This voltage drop propagates while decreasing to reach 0.54 mV at the farthest locations from the infection. As expected, these IR drop values are far below the values considered during design stages that follow a corner-based approach, V_{dd} corners being usually set to $\pm 10\%$ of the nominal V_{dd} value (here 1.2V).

Fig. 11 shows, for two different devices, the voltage drops created by the implementation itself of the LFSR. A global drop of 1 mV appears as well as a really local drop of 12 mV. This confirms that the implementation has a significant impact on the inner supply voltage of the IC. This impact is 20 times larger than the impact of the LFSR activation, i.e., 20 times larger than the impact of the LFSR switching

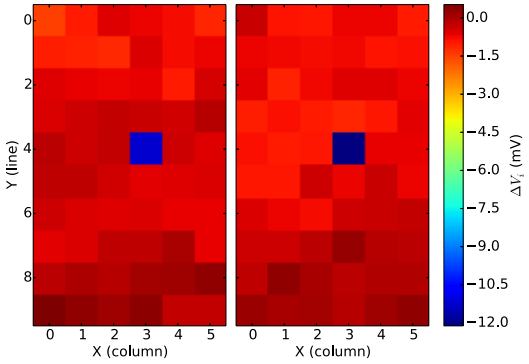

Fig. 11. Maps of the voltage drops induced by the fact of implementing the LFSR.

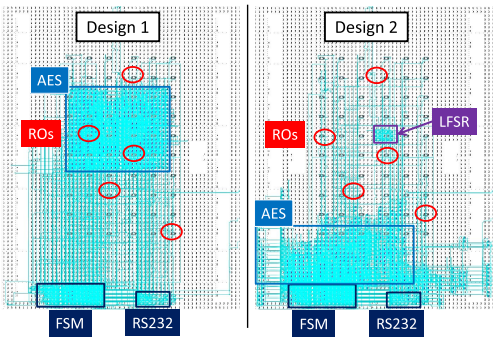

Fig. 12. Two implementations (floorplans) of the considered design.

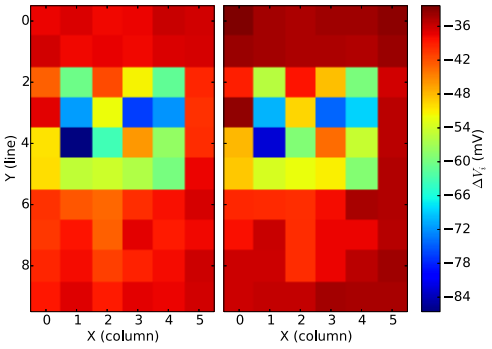

Fig. 13. Voltage drop maps obtained, on two boards, for the first implementation (Design 1) of the AES.

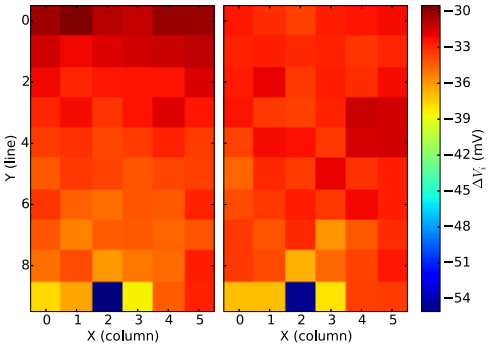

Fig. 14. Voltage drop maps obtained, on two boards, for the second implementation (Design 2) of the AES.

activity. We obtained similar maps for the two boards showing that the phenomenon is relatively independent of process variations.

2) Influence of the Design: The previous results show that an HT insertion modifies the V_{dd} distribution in the IC. At the same time, Fig. 8 indicates that the IC flooplan and thus the way the supply is distributed have an impact on RO performances and more generally on CMOS logic performances. In order to better understand and quantify the design influence on the voltage distribution and thus on CMOS logic performances, we implemented in two different ways the same design into our FPGAs.

More precisely, two different sets of place and route constraints were adopted to integrate an Advanced Encryption Standard (AES) as illustrated in Fig. 12. The AES used is a hardware implementation of the NIST encryption standard specified in [22]. The design also embeds an LFSR to emulate an HT. This LFSR is clocked with a clock net of an AES register. Figs. 13 and 14 show the voltage drops induced by the two implementations of this design. The two maps allow redrawing the floorplan of the design under the RO matrix, as the ROs located in the implemented area are deeply impacted with an impact that goes locally over 80 mV. We also observe a global impact of 30 mV. Despite these observations, the main result is that the voltage distribution significantly depends on how the design is physically implemented.

At this stage of this paper, we can now assume with a high level of confidence that the insertion of an HT modifies the distribution of the supply of an IC at rest. Of course, the amplitude of the related modification depends of the size of the HT and could be really small. This assumption constitutes the basis of the HT detection method described in the next section.

III. Detection Method

This section describes a method for detecting HTs and rough counterfeits. It is based on the results of the previous section and on three contributions with respect to the state of the art:

1) a new infection paradigm;
2) a model of CMOS logic performance variations at design level;
3) a novel distinguisher for the decision making, i.e., to determine if an IC is genuine, counterfeited, or infected.

A. Features of Infected Circuits or Counterfeits

Many methods have been proposed to detect HTs. Among them, a large majority aims at detecting the parasitic switching activity generated by their trigger. However, as shown in the previous section, this parasitic switching activity is not the only measurable trace left by HTs. Another one is the alteration of the inner structure of the IC. For example, the HT insertion modifies the local and global capacitance and resistance of the power and ground networks. This modification induces a different current flow in the IC and thus a different static or dynamic voltage distribution (static or dynamic voltage drops).

Based on the taxonomy given in [4], the considered counterfeits are specific cases of recycled or remarked components. These counterfeits are characterized by a different physical structure (remarked IC) or different electrical characteristics due to the aging effect (reused ICs) and therefore by a different repartition of V_{dd} across the IC.

B. Principle of HT and Counterfeit Detection

Our detection method is based on a *simple principle: the fingerprinting of the static supply voltage distribution over the whole surface of ICs at rest* (i.e., just powered ON with the clock active). In order to do this, a network of sensors is uniformly spread over the whole IC surface to get a cartography of the V_{dd}. Any sensor sensitive to the supply voltage V_{dd} can be used. In the experiments reported in Section IV, the same ROs as the ones considered in Section II are used. Given that the frequency f of an RO is sensitive to the local V_{dd} value, the distribution of f across the IC surface, in the absence of any process variation, is a direct picture of the V_{dd} distribution. Hence, in our approach, we have to get rid of the effect of intradie and interdie process variations. With such an approach, we shall be able to mitigate risks linked to the introduction of HTs at the manufacturing stage.

C. Process Variation Model and Performance Variation Model of CMOS Structures

Given p, an inherent parameter of the fabrication technology, the impact of the process variations is generally described as follows:

$$p = \bar{p} + \Delta p_{\mathrm{inter}} + \Delta p_{\mathrm{intra}} \tag{1}$$

with $\bar{p}$ being the mean (or typical) value of the parameter on a whole lot of a production, $\Delta p_{\mathrm{inter}} \sim N(0, \sigma^2_{\mathrm{inter}})$ the effect of the interdie variations assumed normal, and $\Delta p_{\mathrm{intra}} \sim N(0, \sigma^2_{\mathrm{intra}})$ the impact of the intradie process variations also assumed normal.

This process variation model is well known and widely adopted to simulate the effect of process variations on the parameter p of an IC (a transistor parameter, a resistance, a pn junction, etc.). However, the extraction of the standard deviation values σ_{intra} and σ_{inter} is generally performed on dedicated ICs (regular arrays of MOS transistors [23] or SRAM cells [24]), which are quite uniform relative to their physical structures and under controlled voltage and temperature. Thus, this process variation model does not take into account the impact of the physical structure of ICs (power supply routing, local transistor density, etc.) on the CMOS gate performance or on that of an embedded sensor, which, of course, depends on all process variations through (1). Hence in our case, we shall use the following variation model for the output values $T(x_i, y_i)$ of a sensor i located at (x_i, y_i) in the IC:

$$T(x_i, y_i) = \bar{T} + \Delta T_{\mathrm{inter}} + \Delta T_{\mathrm{intra}} + \Delta T(x_i, y_i) \tag{2}$$

where $\Delta T(x, y)$ is a deterministic value that depends on the position of the sensor in the IC and that models the impact of the IC structure on the sensor performance. To ease the reading, $T(x_i, y_i)$ and $\Delta T(x_i, y_i)$ shall be denoted by (since the beginning of this paper) T_i and ΔT_i, respectively. This temporal notation was adopted to emphasize that the variation model considered in this paper is a spatial model.

It should be noted that this deterministic term must be considered as time varying when dealing with an operating circuit because the power consumption varies with its activity. Here, for sake of simplicity, this dependence is omitted since we aim at fingerprinting the V_{dd} distribution of ICs at rest.

D. Fingerprinting the IC's Structure

Considering the variation model given by 2, fingerprinting the structure of a design featuring a network of q sensors regularly spread on its surface is relatively simple for the same manufacturing lot of ICs. The q values of ΔT_i are calculated by averaging the impact of the process variations on m_{lot} devices of the same lot

$$\Delta T_i = \frac{1}{m_{\mathrm{lot}}} \cdot \sum_{j=1}^{m_{\mathrm{lot}}} T_i^j - \bar{T} = \frac{1}{m_{\mathrm{lot}}} \cdot \sum_{j=1}^{m_{\mathrm{lot}}} \Delta T_i^j \tag{3}$$

$$\sigma_{\Delta T_i} = \sqrt{\frac{1}{m_{\mathrm{lot}}} \cdot \sum_{j=1}^{m_{\mathrm{lot}}} (\Delta T_i^j - \Delta T_i)^2} \tag{4}$$

where

$$\bar{T} = \frac{1}{m_{\mathrm{lot}} \cdot q} \cdot \sum_{j=1}^{m_{\mathrm{lot}}} \sum_{i=1}^{q} T_i^j \tag{5}$$

where T_i^j is the measurement of the output of the sensor i of the device $j \in \{1, \ldots, m_{\mathrm{lot}}\}$ of the considered lot.

With these notations, the vector S^{Design} can be defined as follows:

$$S^{\mathrm{Design}} = [\Delta T_1, \ldots, \Delta T_q, \sigma_{\Delta T_1}, \ldots, \sigma_{\Delta T_q}] \tag{6}$$

where S^{Design} represents the fingerprint of the physical structure of an IC called "design" and is by construction independent of the process variations. This fingerprint is the base of the HT and counterfeit detection methods proposed later in this section.

In the case of this paper, the fingerprint is based on a network of q sensors. However, it can be generalized to off-chip measurements such as EM emanation analysis of ICs. In this case, we note M^j (7) the set of the q values (samples) provided by a DSO during an EM analysis of the jth IC. As a result, we obtain one vector M^j for each IC and then one set of vectors by lot

$$M^j = \left[m_1^j, \ldots, m_k^j, \ldots, m_q^j\right]. \tag{7}$$

By replacing T_i^j by m_i^j in (3) and (4), S^{Design} is now defined as

$$S^{\mathrm{Design}} = [\Delta M_1, \ldots, \Delta M_n, \sigma_{\Delta M_1}, \ldots, \sigma_{\Delta M_q}]. \tag{8}$$

E. Detection Methodology

The starting point of our methodology is the addition of a network of sensors sensitive to V_{dd}. Those sensors are placed so as to cover the whole IC surface. The granularity, i.e., the distance between two sensors, is chosen by designers depending on the tradeoff between detection capabilities and costs.

When the first run or the test run (which are less likely to be infected, as this run is dedicated to the characterization that allows HT detection) is received, the integrity of some devices is verified to qualify the whole lot. This could be done by applying or using optical means [25]. Once the first production lot is qualified, the signature (6) of the design is calculated using (3) and (4). This fingerprint constitutes the reference fingerprint for the considered design. It should be noted that if large design modifications have been done between the test run and the first production lot, the latter should be qualified HT free using reverse engineering methods. The designer will then usually order other runs (production runs) from the same foundry or from another one that offers the same technology node. Once those new production lots are received, their corresponding fingerprints are calculated and are compared with the reference one to verify that the newly received lots have not been corrupted.

In the same way, at some (later) points in time, the designer can have "field returns," which could contain counterfeits. Despite the hard problem of aging, with the reference fingerprint, some preliminary tests can be done to check the origin of these devices before the application of expensive, complex, and destructive reverse engineering methods. To do that, the designer extracts the fingerprint of the suspected device and compares it with the reference fingerprint to finally get a probability that the device has been recently fabricated and is genuine. If the probability is too low, complementary analyses (like reverse engineering) can be applied.

The above procedures require the comparison of the reference fingerprint with that of a new production lot in order to detect the eventual presence of an HT (case 1). The procedures also require the comparison of the reference fingerprint S^{REF} with the fingerprint of a single device in order to detect counterfeits (case 2).

1) Case 1 (HT Detection): When the integrity of a new lot of devices has to be checked, the first step is to calculate its fingerprint $\mathrm{S}^{\mathrm{NewRun}}$. Since this signature shall be calculated using a high number of devices (>100), the estimate of means can be considered as reliable. It is therefore possible to apply a statistical tool working on the means such as the T-test and more precisely on Welch's test

$$w_i = \frac{S_i^{\mathrm{Ref}} - S_i^{\mathrm{NewRun}}}{\sqrt{\frac{\left(s_{n+i}^{\mathrm{Ref}}\right)^2}{m_{\mathrm{lot}}} - \frac{\left(s_{n+i}^{\mathrm{NewRun}}\right)^2}{m_{\mathrm{lot}}}}}. \tag{9}$$

where W is the vector composed of the q T-values

$$W = [w_1, \ldots, w_k, \ldots, w_q]. \tag{10}$$

If a value of W is over the critical value, the lot is considered infected.

2) Case 2 (Counterfeit Detection): The case of a suspected "field return" is more difficult to treat because the suspected device could be a genuine but old circuit whose characteristics have been modified by the aging process. In the latter case, the aging may have altered its signature so that it cannot be recognized as genuine using the proposed technique. Therefore, in this case, the proposed procedure only allows declaring a suspected device as genuine if its fingerprint matches with the reference one. However, it does not allow declaring the device as a counterfeit if its fingerprint does not match. Further tests are necessary to prove that it is effectively a counterfeit.

The proposed technique thus works in that case as an authentication technique. However, there is room for enhancements allowing one to take into account aging process in the procedure. A starting point could be to collect reference fingerprints of aged circuits.

Treating "field returns" is also more difficult because the fingerprint described so far for HT detection cannot be calculated on a single device. Indeed, we only have at disposal the $T^{\mathrm{Suspected}}$ of the considered device. In this case, we first calculate $T_i^{\mathrm{Suspected}} - \bar{T}^{\mathrm{Suspected}}$. This is done for mitigating the impact of interdie variations (which have, among all variations, the greatest impact on the RO performance) and that of temperature. This step can thus be viewed as a process centering. This has done the probabilities that all $T_i^{\mathrm{Suspected}} - \bar{T}^{\mathrm{Suspected}}$ values come from the normal distributions below are computed. To that end, the probabilities for all sensors are combined (a multinormal distribution is defined with all $\sigma_{\Delta T_i}$) to obtain the probability that the considered device is a genuine one

$$N\left(0, \left(s_{q+i}^{\mathrm{Suspected}}\right)^2\right) = N\left(0, \sigma_{\Delta T_i}^2\right). \tag{11}$$

This is a way to decide if the remaining intradie process variations could explain or not the observed differences between the first part of the reference signature and the RO performances of the suspected IC. If this is the case, it is really unlikely that the device is a counterfeit. If this is not the case, it could be a counterfeit, but additional tests are mandatory because of the aging effect.

F. Adaptive T-Test

The T-test is a statistical test that allows deciding if two distributions have the same mean or not. The null hypothesis, the equality of the means, is rejected if the T-value is above a critical value, t_{critic}. Usually, t_{critic} is chosen by the user by setting the p-value (α). Generally, an α of 0.05 is chosen. This means that a false positive rate of 5% is accepted. In practice, rejecting the null hypothesis is equivalent to declare the IC lot infected. Thus, setting the α to 5% means that we accept to throw to the bin 5% of noninfected lots. In a context of production, this is huge. On the other hand, choosing a lower p-value resumes in lowering the detection capability and thus accepting to have a significant probability to let infected ICs or lots pass the test.

To overcome this problem, we propose an adaptive solution to set t_{critic}. The idea is to consider that an HT alters the T-values of only a minority of T-tests because stealthy HTs have reduced temporal and/or spatial impacts on the

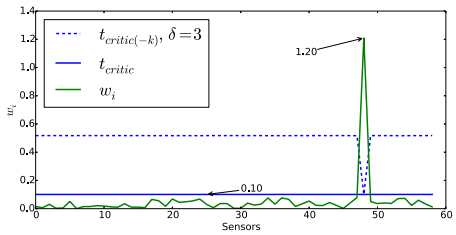

Fig. 15. Adaptive t_{critic} value.

IC behavior. Indeed, depending on the source of information, the values can be considered related to a spatial location or related to a temporal point. Thus, if the sensor array (or the EM traces) features a sufficiently large number of sensors (or time samples), only a few of them will be affected by the presence of an HT. As a result, we do assume that if many T-tests are applied to compare two signatures, most of them must pass the test. Therefore, we propose to compute t_{critic} as follows. The calculus of t_{critic} starts by the one of $t_{\text{critic}_{(-k)}}$, the critical T-value associated to the kth sensor or sample

$$t_{\text{critic}(-k)} = \mathcal{W}_{(-k)} + \delta \cdot \sigma_{(-k)} \tag{12}$$

where the $t_{\text{critic}_{(-k)}}$ is the sum of two terms: the mean $\mathcal{W}_{(-k)}$ given (13) and the standard deviation $\sigma_{(-k)}$ given (14). The parameter δ is the threshold that has to be adapted to the measurement source; this point will be discussed later

$$\mathcal{W}_{(-k)} = \frac{1}{q-1} \sum_{i=1, i\neq k}^{q} \mid w_i \mid \tag{13}$$

$$\sigma_{(-k)} = \frac{1}{q-1} \sum_{i=1, i\neq k}^{q} (w_i - \mathcal{W}_{(-k)})^2. \tag{14}$$

As shown 13, w_k is rejected from the calculation of $t_{\text{critic}_{(-k)}}$ because it must not be taken into account for a proper setting of $t_{\text{critic}_{(-k)}}$. Indeed, if an HT impacts the kth sensor (alsp the kth sample), the kth T-value, w_k, could become really large. As a result, taking it into account the calculus of $\mathcal{W}_{(-k)}$ would imply an undesired overestimate of $\mathcal{W}_{(-k)}$ and $\sigma_{(-k)}$ with respect to the case in which the kth sensor is not impacted by an HT.

After the computation of the q $t_{\text{critic}_{(-k)}}$ values, one can finally obtain t_{critic} by picking up the minimal $t_{\text{critic}_{(-k)}}$ value

$$t_{\text{critic}} = \underset{k=\{1,\ldots,q\}}{\text{argmin}} \{t_{\text{critic}(-k)}\}. \tag{15}$$

As for an illustration, Fig. 15 shows all $t_{\text{critic}_{(-k)}}$ values and t_{critic} in the case of an infection around RO48. The green curve corresponds to the vector W obtained by comparing an infected lot (IL) with a genuine one through a network of 60 sensors. It shows that the T-value, w_{48}, is significantly higher than all other values. The blue dotted curve represents the 60 $t_{\text{critic}_{(-k)}}$ values computed with $\delta = 3$. As one can observe, the minimal $t_{\text{critic}_{(-k)}}$ value is obtained when $k = 48$, and thus $t_{\text{critic}} = t_{\text{critic}(-48)}$.

With such a definition of t_{critic}, we are now able to adaptively set a decision making threshold for detecting HT despite the influence of process variations and measurement noise.

IV. Experimental Results

A. Experimental Setup

The measurement setup is similar to the setup presented in Section II. On each Spartan-3E-1600 FPGA, a 128-bit-key AES, an RS232 communication block and an FSM have been placed and routed. An array of 60 ROs has been added to the design. Each RO is coupled with a clock divider by two so as to be able to measure the 60 frequencies on an IO pad through a multiplexer. The area overhead incurred by the addition of our on-chip detection hardware is about 3.2% of the FPGA resources. The frequency measurements are performed with an oscilloscope from Lecroy featuring a 4-GHz bandwidth and a 40-Gsamples/s sampling rate.

In order to obtain accurate measurements (accuracy of ± 0.025 ps), each frequency estimation is done by measuring the duration equivalent to 100 periods and by repeating this experiment 100 times to obtain a mean value of each RO period: T_i^j. During these measurements, the IC is kept inactive, i.e., just powered ON and with the clock running.

The time spent to measure the 60 values T_i on a board is lower than 2 min, which is short enough to consider the temperature as constant in our laboratory (equipped with air conditioning) environment. The key point is to ensure a constant temperature during the fingerprint extraction of each lot so as to limit the impact of temperature variations on the fingerprint. However, keeping the temperature identical to extract the signature of all considered lots is not a tight constraint. Indeed, during the computation of each fingerprint, the first step is to center the T_i distributions. This results in suppressing the global shift of all T_i^j with temperature, provided the latter is kept constant during the fingerprint extraction of each lot. Of course, despite this observation, it remains preferable to control temperature and work in a controlled environment. In order to guarantee a good stability of the supply voltage, the FPGA is powered by a stabilized dc supply source with an accuracy of 0.05%.

To emulate the effect of an HT, a 64-bit LFSR is used. It occupies an area of 48 slices, which represents 0.32% of the FPGA's surface and 2.7% of the AES. Note that the AES alone is mapped onto 1778 slices. The LFSR is clocked at 50 MHz by taking the clock input of a DFF of the AES. This HT can therefore be considered as a sequential HT.

To emulate counterfeits, several constrained place and route steps of the design are performed. Three different floorplans of the same HDL code (three leftmost pictures) have been implemented. One of them (Design 1) is considered as the original/genuine design and the other two (Designs 2 and 3) are considered as counterfeits.

B. Validation of the On-Chip Detection

1) Counterfeit Detection: In Section III, we introduced a variation model of CMOS logic performance and thus of our sensors. This novel model introduces a deterministic term that expresses the impact of the design structure on the sensor performance and particularly the impact of the power distribution. For this novel model being the base of the proposed detection method, we start by evaluating its relevance.

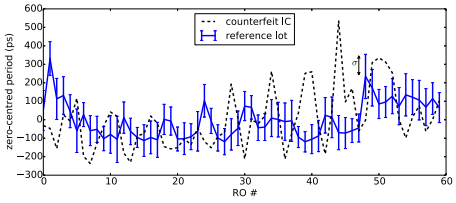

Fig. 16. S^{Design1} and signature of a counterfeit.

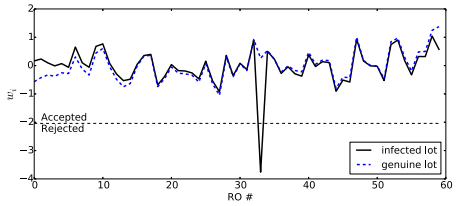

Fig. 17. T-test between the fingerprints of 15 infected and 15 genuine devices with the signature of the 15 reference devices.

To show that our method can determine whether a suspected IC is a counterfeit or not (compared with a reference lot) using our model, we implemented different designs, functionally equivalent but with different place and route constraints. The frequencies of the 60 ROs from the first design have been measured on 15 ICs. Then, the frequencies of the 60 ROs from the second design have been measured on one IC. Fig. 16 shows the complete fingerprint (calculated from 15 devices) of the design 1 (dark curve), i.e., the values $\Delta T_i \pm \sigma_{\Delta T_i}$, and the fingerprint of a suspected device (dotted line). In this case, there is visually no doubt that the considered device is a counterfeit. For example, the ROs 30, 39, and 40 are out of the $\pm 3 \cdot \sigma_{\Delta T_i}$ measured on the reference lot. This result validates the proposed technique to detect rough counterfeits and thus the considered variation model.

2) HT Detection: The detection method of an IL is similar to that of a counterfeit lot, although the alteration of the physical structure is expected to be significantly smaller and localized. Fig. 17 shows the results obtained by applying the T-test (lower picture) in order to verify the integrity of lots of 15 infected and 15 genuine ICs with the reference lot of 15 ICs. 30 boards were used. To emulate the infection (the presence of a sequential HT), a 64-bit LFSR (48 slices) has been added to Design 1. Both the difference of means (DoM) and the T-test allow detecting an anomaly located around RO33, which is effectively close to the LFSR. Moreover, the DoM remains low between the reference and the genuine lots (GLs). In this case, the absolute T-values ($|w_i|$) do not exceed 2.04 for $i \in \{1, \ldots, 60\}$. GLs are therefore recognized as uninfected lots. These results validate the proposed detection methodology and above all the proposed variation model of the performance of a CMOS structure in a real design, which strongly depends on the power distribution in advanced CMOS technologies.

C. Validation of the Off-Chip Detection Technique

For the off-chip detection validation, the experimental setup is similar to the previous one, except that the measurements were done while the ICs were operating, and more precisely during AES encryptions. For each IC, 10 000 traces were acquired in order to reduce measurement noise by averaging. A set of 100 000 random plaintexts has been generated and applied identically for each IC characterization. The traces from each IC needed to be resynchronized with the traces from the other boards. To that end, the EM peaks related to the execution of the AES rounds were considered as time references. Each acquisition lasted 250 ns, and the obtained traces were composed of 10 000 samples, and thus M was made up of $q = 10\,000$ values; this is extremely large compared with the case of the 60 embedded sensors.

In a similar way to the previous case, a 64-bit LFSR was used to emulate the effect of an HT. Contrary to the previous case, we did not look for the passive and continuous impact of the HT on the measurements. This is impossible with EM measurements that reveal variations of the current flowing in ICs. This setup was thus elaborated in order to detect the switching activity of HTs that have a limited impact in time. For this purpose, the LFSR is clocked by the global clock and synchronized with the eighth round of the AES. Thus, the LFSR has a switching activity one clock cycle during each encryption. It was placed in the AES as it should be the worst case due to the influence of the AES activity.

1) Distinguisher Adaptation: In the previous case, we considered that only one sensor was impacted by the HT. Actually, the higher the resolution of the measurement technique is, the larger the number of impacted samples will be. Thus, we need to consider excluding several samples instead of only one. Given r the number of samples excluded, (13) and (14) are modified as follows:

$$\mathcal{W}_{(-k)} = \frac{1}{n-1} \sum_{\substack{i=1 \\ i \neq \left[k-\frac{r}{2}, k+\frac{r}{2}\right]}}^{n} w_i \tag{16}$$

$$\sigma_{(-k)} = \frac{1}{n-1} \sum_{\substack{i=1 \\ i \neq \left[k-\frac{r}{2}, k+\frac{r}{2}\right]}}^{n} (w_i - \mathcal{W}_{(-k)})^2. \tag{17}$$

To determine the size of the exclusion window, two parameters have to be taken into account: 1) the impact of the HT on our measurements in terms of surface (also duration) for a spatial (also temporal) analysis and 2) the resolution (spatial or temporal) of the measurement setup. Using these two parameters, we can determine the maximum number of impacted samples in the vector M and determine the exclusion window r.

Having the clock period of the IC under EM analysis being equal to 20 ns and the sampling rate of the oscilloscope being equal to 40 Gsamples/s (25 ps a sample), the duration of the exclusion window could be set to (50 ns/25 ps) = 2000 samples if one aims at verifying the integrity of the DUT with a resolution of roughly a clock cycle. This is because the impact of an electrical activity of duration equal to one clock cycle can last more than one clock period. It should be observed that setting r to a too low value could lead to enlarging t_{critic} and thus increasing the false negative rate. On the other hand, taking a too large window has no real influence on t_{critic} as long as the number of remaining samples allows accurately

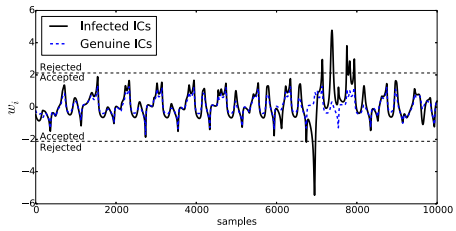

Fig. 18. T-test between the fingerprints of 15 infected and 15 genuine devices with the signature of the 15 reference devices when considering the EM measurements.

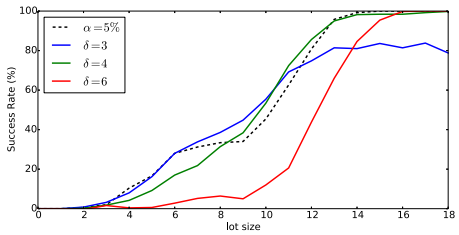

Fig. 19. SR obtained with the embedded sensor approach (spatial detection method): the case of a 64-bit HT.

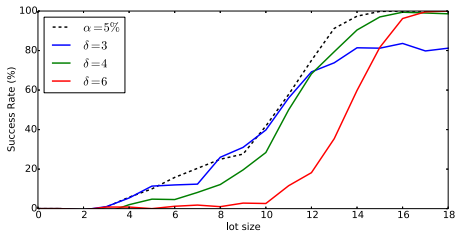

Fig. 20. SR obtained with the embedded sensor approach (spatial detection method): the case of a 32-bit HT.

computing the mean and the standard deviation involved in the previous equations.

2) Results: Fig. 18 gives the T-values obtained when considering EM measurements. The green curve corresponds to the vector W obtained by comparing the reference lot with a genuine one. The red curve corresponds to the vector W obtained by comparing the reference lot with an IL. As expected, the two curves are superposed over the large part of the vector. However, from sample 6300 to sample 8200, the two curves are significantly different. The HT is therefore detected. As a result, an exclusion window of 2000 points seems appropriate to detect an HT that is operating only during one clock cycle. The dotted curves in Fig. 18 correspond to t_{critic}. It shows that t_{critic} is not impacted by the HT, but only by the process variations and the measurement noise.

D. Success Rate

One may question on the efficiency of the proposed distinguisher, which is highly dependent on the number of ICs in each lot. To ease this efficiency analysis of our detection technique and of other solutions, we introduced herein the idea of success rate (SR). The SR is defined with regard to the number of ICs contained in each lot. More precisely, the SR is the percentage of ILs classified as infected (true positives) minus the percentage of GLs classified as infected (false positives)

$$\text{SR} = \frac{\#\ \text{IL} -> \ \text{infected}}{\#\ \text{IL}} - \frac{\#\ \ \text{GL} -> \ \text{infected}}{\#\ \text{GL}}. \quad (18)$$

During our efficiency analysis, we did consider the reference lot and lots under test of the same size. Actually, 19 FPGA boards were used to treat infected lots and GLs. To compute the SR for a given lot size and noted l, l boards were randomly drawn to build the reference lot and l boards were randomly drawn to create tested lots. The draw was done in a way that minimizes the intersection between the two sets of boards. Therefore, there is no intersection for $l < 10$ and the ICs are not fully independent for $l \geq 10$. Thus, the impact of the process variation on the result decreases faster with the lot size than if fully independent lots were used.

For each draw, the adaptive T-test was applied twice between the two lots according to the two considered cases. In the first case, the reference lot and the tested lots are GLs (without LFSR/HT). This case allowed computing the false positive rate. In the second case, the reference lot is of course a GL, while the tested lots are ILs. This allowed computing the true positive rate. 500 draws were done by the lot size.

E. Success Rate With Embedded Sensors

Fig. 19 shows the evolution of the SR with regard to the lot size, for a 64-bit HT. The dotted curve in Fig. 19 represents the SR obtained with the T-test with the p-value set to the classical 5 %. The other curves represent the SR obtained with different values of δ used to determine t_{critic}.

As shown, the results significantly change with δ. If the value is too low, the false positive rate (the second term of 18) increases and the SR does not reach 100% (with $\delta = 3$, it never exceeds 80%) because of some fluctuations due to process variations or measurement noises leading to T-values (outliers) exceeding t_{critic}.

In the contrary case, if δ is too high, the distinguisher accepts most of the infected ICs as genuine. As a result, the true positive rate (the first term of 18) decreases toward 0. In our example, Fig. 19, this does not occur since we reach 100% of success with the highest threshold $\delta = 6$.

We also analyzed the impact of the HT size (in bits) on the SR evolution. Figs. 19–22 give the obtained SR for infections by 64-, 32-, 16-, and 8-bit LFSRs, respectively. As expected, it appears that the SR decreases with the HT size. For the smallest HT, the SR never reaches 100% even if it remains high (>80% for $\delta = 4$) when using the adaptive distinguisher.

However, observing Figs. 19–22 discloses a shift of the SR curves to the right as the HT size decreases. This means that the lot size needed to obtain a high SR grows as the HT size decreases. For HT sizes of 8 and 16 bits, we do not have lots enough large to obtain an SR of 100%. However, as the curves corresponding to 8- and 16-bit infection and the curves corresponding to 32- and 64-bit infections have the same shape, it seems that larger lots would allow distinguishing the infection through process variations with a higher SR.

In addition to this observation, one may observe that the classic T-test provides similar SR values for 64-bit and 32-bit infections than with our adaptive technique.

TABLE I
T-STATISTIC WITH RESPECT TO THE LFSR SIZE AND DISTANCE

Size	8 bits			16 bits			32 bits			64 bits		
	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}
0,0	0.08	1.26	48	0.11	1.6	48	0.08	3.85	48	0.08	4.79	48
x-2	0.11	0.62	48	0.13	0.09	6	0.14	1.06	48	0.07	2.59	48
x-4	0.10	0.07	6	0.10	0.11	26	0.09	2.01	48	0.13	2.58	48
y+2	0.13	0.08	6	0.12	0.6	48	0.08	3.01	48	0.08	2.56	48
y+4	0.10	0.38	48	0.11	0.65	48	0.08	2.92	48	0.12	2.86	48

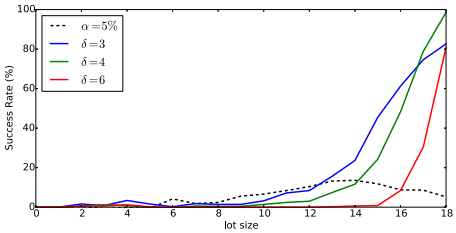

Fig. 21. SR obtained with the embedded sensor approach (spatial detection method): the case of a 16-bit HT.

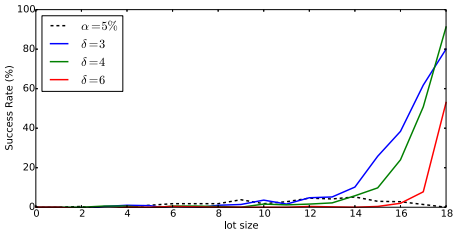

Fig. 22. SR obtained with the embedded sensor approach (spatial detection method): the case of an 8-bit HT.

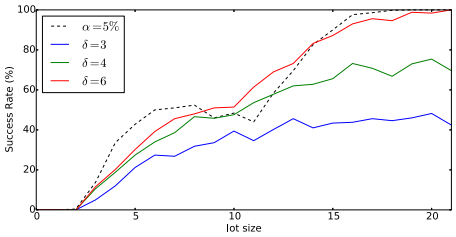

Fig. 23. SR obtained with EM analysis (time-domain detection method): the case of a 64-bit HT.

However, the SR remains below 20 % for 16- and 8-bit infections using this classical approach, while it is higher than (>80% for $\delta = 4$) with the adaptive distinguisher. This shows that our distinguisher is relevant for both large and small (stealthy) infections.

Finally, regarding the impact of the thresholds, $\delta = 4$ appears to be a good choice to optimize the false positive and the false negative rates with a minimal lot size. However, the differences observed between the SR values obtained with the different δ values are moderate.

F. Success Rate With EM Analysis

Fig. 23 shows the evolutions of the SR with respect to the lot size when EM analysis is preferred to the embedded approach. Each curve represents the SR obtained with a different value for δ. The dotted curve in Fig. 23 represents the SR obtained with the classical T-test, i.e., α equal to 5%. In the same way as in the previous experiments, it has been observed that the choice of threshold fixes the SR evolution by changing the false positive and the true positive rates.

However, these results are significantly different from those obtained with embedded sensors. Indeed, one may observe that with $\delta = 3$, the SR does not exceed 50%. This means that we are highly impacted by false positives. This is may be due to a higher measurement noise in EM traces or to realignment errors. Fig. 23 also shows that $\delta = 6$ seems to be a valid choice for EM measurements as it offers the best SR for all lot sizes. One may also observe that the classic T-test ($\alpha = 5\%$) allows obtaining a similar SR than with $\delta = 6$ and that both tests allow reaching 98% of SR within the available lot sizes in the case of a 64-bit HT. However, because we did not succeed in achieving high enough SR values for smaller HTs, it appears that EM analysis seems less efficient than an embedded-sensor-based detection technique. Nevertheless, this result confirms that the adaptive T-test is also efficient when applied on off-chip EM measurements.

G. Impact of the HT Size and Distance to HT

In order to determinate the density of sensors required to obtain a high coverage of the IC surface, the impact on the detection capability with respect to the distance separating the RO and the trojan was studied as well as the impact of the size (in bits) of the infection. We analyzed these two parameters by routing many designs with different LFSR locations. For each location, four different LFSR sizes were considered: 8, 16, 32 and 64 bits. The first considered LFSR location is that of an RO, where the distance measured in slices is thus equal to zero. The other locations correspond to shift of the LFSR to the left (or up) by two and four slices. These locations are denoted by $x - 2$ and $x - 4$ (also $y + 2$ and $y + 4$).

The detection results obtained with the T-test are given in Table I for an LFSR placed close to RO48. In Table I, $t_{\max}$ is the greatest T-value (in absolute value) obtained over the 60 ROs, t_{argmax} is the RO index for which $t_{\max}$ is obtained, and t_{critic} is the adaptive critical value computed with 15.

As shown, the obtained T-values quickly decrease with the size of the HT. More precisely, one may observe that as soon as the HT size becomes smaller than 32 bits and the HT is placed far from RO48, it becomes difficult to detect it. In such conditions, $t_{\max}$ becomes lower than t_{critic} and is not obtained for RO48. However, one may also observe that small HTs are still correctly detected when placed close to R048. Considering these results, we may conclude that the range of detection of each RO is too narrow and that a significant

TABLE II
T-STATISTIC WITH RESPECT TO THE LFSR SIZE AND DISTANCE WITH EXTENDED RO

Size	8 bits			16 bits			32 bits			64 bits		
	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}
0,0	0.21	0.55	35	0.47	0.87	35	0.48	1.49	35	0.39	4.79	35
x-2	0.17	0.45	35	0.47	0.67	35	0.44	1.70	35	0.39	3.86	35
x-4	0.11	0.90	35	0.13	0.78	35	1.16	2.27	35	0.4	3.41	35
y+2	0.14	0.34	35	0.46	0.90	35	0.43	2.06	35	0.42	4.41	35
y+4	0.15	0.72	35	1.18	1.57	35	0.44	1.94	35	0.71	5.02	35

TABLE III
T-STATISTIC WITH RESPECT TO THE LFSR SIZE AND POSITION WITH COMPACT RO

Size	8 bits			16 bits			32 bits			64 bits		
	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}	t_{critic}	t_{max}	t_{argmax}
outside	0.08	1.26	48	0.11	1.6	48	0.08	3.85	48	0.08	4.79	48
near	0.10	3.4	19	0.15	3.52	19	0.23	2.46	19	0.06	2.88	19
insisde	0.08	1.52	9	0.06	6.7	9	0.09	5.2	9	0.08	3.94	9

number of sensors are needed to cover the whole IC surface. This could be not acceptable according to design constraints.

H. RO Improvement

Previous experiments done with *compact* ROs of five stages showed that their detection range is insufficient to detect small HTs. The detection capability of *extended* ROs covering each a larger surface of the IC was therefore analyzed. *Extended* RO features 13 stages (rather than 5) uniformly spread over a square of nine slices. These *extended* ROs have a mean period of around 21 ns when *compact* RO have a period of 13ns.

Results given in this section are computed from a set of 20 boards. They were obtained with exactly the same setup as the previous one except that only 50 of them were implemented to cover the IC surface instead of 60. The area overhead incurred by the addition of our on-chip detection hardware is about 4.7% of the FPGA resources.

Table II shows detection results obtained with *extended* ROs considering different distances between the HT and the ROs. Comparing Table II with Table I, one can observe that despite the obtained T-test values are lower than those obtained with the *compact* RO, *extended* ROs have a greater detection range. Indeed, while cases (8 bits: $x-4$, $y+2$) and (16 bits: $x-2$, $x-4$, $y+4$) were not detected using the compact ROs, they are now detected using the extended ones. Therefore, the design of RO have a significant influence on the spatial coverage for a given sensor density. It seems interesting to route RO so that they occupy a large area rather than route them on the smallest possible surface area.

I. Impact of the Surrounding Design

At this stage of this paper, nearly all reported experiments were performed with an array of embedded sensors and the HT far from the AES, i.e., without any immediate surrounding logic. One may wonder about the influence of surrounding logic on the detection capability. To get insight into this influence, Table III shows the T-values obtained for the extended RO, the closest from the infection, while the latter is *outside, near, or inside* the AES added to our test chip. For these three positions, the impact of the LFSR size in bits (8, 16, 32, and 64 bits) has also been studied. As shown, in all cases, T-values are greater than t_{critic} and thus the HT detected. From these results, we may suggest that the surrounding logic does not seem to have any significant impact on the detection capability of our detection technique.

V. DISCUSSION

At this stage of this paper, we introduced and experimentally validated a technique to detect HT and rough counterfeits. The method works at a lot level and is based on the cartography, done with an array of sensors sensitive to V_{dd}, of the static distribution of V_{dd} of ICs that are at rest (powered and the clock active). It aims at detecting changes in the V_{dd} distribution induced by the insertion of malicious additional logic. Because the fingerprinting is done on ICs at rest, one may expect to detect any type of HT whose insertion requires extra logic.

Regarding the size of HTs that can be detected, it clearly appears from the obtained results that it depends on the number of ICs in the reference and tested lots. With lots with only 19 devices, we were able to detect the insertion of an 8-bit LFSR used to emulate HTs after the rough optimization of RO used as sensors. One can expect detecting smaller HTs with larger lots. There is also room to increase the detection capability by designing sensor with a higher sensitivity to V_{dd} and a lower sensitivity to process variations than the basic ROs considered in this paper.

Concerning counterfeit, the proposed technique can be used to detect rough counterfeits characterized by a different floorplan or a different physical structure but functionally identical, namely, remarked ICs. More precisely, the proposed technique allows deciding if a suspected IC is identical to the reference ICs, but does not allow deciding if an IC is a counterfeit or not. Additional techniques should be employed to take this decision. This limitation is essentially due to the aging process of ICs that can induce changes in IC fingerprints. However, we believe there is room in developing experimental techniques or techniques based on aging models to derive the fingerprints of aged ICs to overcome this limitation.

VI. CONCLUSION

This paper first describes experiments conducted to quantify the impact of an HT insertion on an on-chip sensor network

and on the behavior of a test chip. During these experiments, it has been observed that the static impact of an HT insertion itself is significantly greater than the dynamic impact associated with its switching activity.

Based on this observation, an efficient approach for detecting HT and rough counterfeits has been introduced. It is based on the monitoring of the static distribution of the supply voltage over an IC's surface, but also on a new variation model of the performance of CMOS logic.

This model allows extracting IC signatures and more precisely design signatures by getting rid of process variation issues. The model and all related methods introduced in this paper have been successfully validated and characterized on a set of 24 FPGA boards.

In addition to the proposed detection methodology, an adaptive distinguisher has been introduced. It aims at improving the decision making thanks to an adaptive statistical threshold. The principle of the proposed distinguisher is to set the threshold by considering that HTs are compact in space and/or have a limited influence in time.

REFERENCES

[1] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *IEEE Des. Test Comput.*, vol. 27, no. 1, pp. 10–25, Jan. 2010.

[2] K. Xiao and M. Tehranipoor, "Bisa: Built-in self-authentication for preventing hardware Trojan insertion," in *Proc. HOST*, Jun. 2013, pp. 45–50.

[3] S. M. H. Shekarian and M. S. Zamani, "A trust-driven placement approach: A new perspective on design for hardware trust," *J. Circuits, Syst. Comput.*, vol. 24, no. 8, p. 1550115, 2015.

[4] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proc. IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug. 2014.

[5] F. Courbon, P. Loubet-Moundi, J. J. A. Fournier, and A. Tria, "A high efficiency hardware Trojan detection technique based on fast SEM imaging," in *Proc. DATE*, Mar. 2015, pp. 788–793.

[6] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *Proc. IEEE SSP*, May 2007, pp. 296–310.

[7] J. Li and J. Lach, "At-speed delay characterization for ic authentication and Trojan horse detection," in *Proc. HOST*, Jun. 2008, pp. 8–14.

[8] M. Abramovici and P. Bradley, "Integrated circuit security: New threats and solutions," in *Proc. CSIIRW*, 2009, Art. no. 55.

[9] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "MERO: A statistical approach for hardware Trojan detection," in *Cryptographic Hardware and Embedded Systems—CHES* (Lecture Notes in Computer Science), vol. 5747, C. Clavier and K. Gaj, Eds. Berlin Germany: Springer, 2009, pp. 396–410.

[10] S. Narasimhan *et al.*, "Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust*, Jun. 2010, pp. 13–18.

[11] K. Bowman *et al.*, "Dynamic variation monitor for measuring the impact of voltage droops on microprocessor clock frequency," in *Proc. IEEE CICC*, Sep. 2010, pp. 1–4.

[12] C. Lamech, R. M. Rad, M. Tehranipoor, and J. Plusquellic, "An experimental analysis of power and delay signal-to-noise requirements for detecting Trojans and methods for achieving the required detection sensitivities," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 1170–1179, Sep. 2011.

[13] X. Zhang and M. Tehranipoor, "Ron: An on-chip ring oscillator network for hardware Trojan detection," in *Proc. DATE*, Mar. 2011, pp. 1–6.

[14] M. H. DeGroot and M. J. Schervish, *Probability and Statistics*, 4th ed. London, U.K.: Pearson, 2011.

[15] A. Ferraiuolo, X. Zhang, and M. Tehranipoor, "Experimental analysis of a ring oscillator network for hardware Trojan detection in a 90nm ASIC," in *Proc. ICCAD*, 2012, pp. 37–42.

[16] Y. Cao, C.-H. Chang, and S. Chen, "Cluster-based distributed active current timer for hardware Trojan detection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 1010–1013.

[17] O. Soll, T. Korak, M. Muehlberghuber, and M. Hutter, "Em-based detection of hardware Trojans on FPGAs," in *Proc. HOST*, 2014, pp. 84–87.

[18] J. Balasch, B. Gierlichs, and I. Verbauwhede, "Electromagnetic circuit fingerprints for hardware Trojan detection," in *Proc. EMC*, Aug. 2015, pp. 246–251.

[19] N. Xuan Thuy, N. Zakaria, S. Bhasin, G. Sylvain, and D. Jean-Luc, "Method taking into account process dispersion to detect hardware Trojan Horse by side-channel analysis," *J. Cryptogr. Eng.*, vol. 6, pp. 239–247, Sep. 2016.

[20] C.-W. Liu and Y.-W. Chang, "Floorplan and power/ground network co-synthesis for fast design convergence," in *Proc. ISPD*, New York, NY, USA, 2006, pp. 86–93.

[21] Federal Information Processing Standards Publication 197. (2007). *Microblaze Development Kit Spartan-3E 1600E Edition User Guide*. [Online]. Available: http://www.digilentinc.com/Data/Products/S3E1600/ug257.pdf

[22] Federal Information Processing Standards Publication 197. (2001). *Specification for the Advanced Encryption Standard (AES)*. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[23] A. Keshavarzi *et al.*, "Measurements and modeling of intrinsic fluctuations in mosfet threshold voltage," in *Proc. ISLPED*, New York, NY, USA, 2005, pp. 26–29.

[24] A. Bhavnagarwala *et al.*, "Fluctuation limits & scaling opportunities for CMOS SRAM cells," in *IEDM Tech. Dig.*, Dec. 2005, pp. 659–662.

[25] F. Stellari, P. Song, and H. Ainspan, "Functional block extraction for hardware security detection using time-integrated and time-resolved emission measurements," in *Proc. IEEE 32nd VTS*, Apr. 2014, pp. 1–6.

Maxime Lecomte received the M.S. degree from the Ecoledes Mines de Saint-Etienne, Saint-Étienne, France, in 2013. He is currently pursuing the Ph.D. degree with CEA-TECH, Gardanne, France, where he is involved in the development of embedded techniques to detect hardware Trojans and participates to security characterization activities of the laboratory.

His current research interests include secure IC design, side-channel analysis and fault injection techniques, and IC integrity and authenticity.

Jacques Fournier received the M.S.E.C.E. degree from the Georgia Institute of Technology, Atlanta, GA, USA, the Ph.D. degree in computer science from the University of Cambridge, Cambridge, U.K., and the Habilitation degree from the University of Limoges, Limoges, France.

He was with the Security Laboratory of Gemalto for over eight years before joining CEA Tech, Gardanne, France, in 2009. He joined CEA Leti, Grenoble, France, to manage cybersecurity research programs in 2016. His current research interests include trusted hardware, cryptographic accelerators, secure embedded software, and secure architectures for embedded systems.

Philippe Maurine received the M.S. and Ph.D. degrees in electronics from the University of Montpellier, Montpellier, France, in 1998 and 2001, respectively.

Since 2003, he has been an Assistant Professor with the Laboratory of Informatics, Robotics, and Microelectronics, University of Montpellier, developing microelectronics in the engineering program of the University. His current research interests include adaptive system-on-chip design, secure IC design, secure embedded software, side-channel analysis, and fault injection techniques.