Electromagnetic Circuit Fingerprints for Hardware Trojan Detection

Josep Balasch, Benedikt Gierlichs, and Ingrid Verbauwhede
KU Leuven Dept. Electrical Engineering-ESAT/COSIC and iMinds
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
Email: firstname.lastname@esat.kuleuven.be

Abstract—Integrated circuit counterfeits, relabeled parts and maliciously modified integrated circuits (so-called Hardware Trojan horses) are a recognized emerging threat for embedded systems in safety or security critical applications. We propose a Hardware Trojan detection technique based on fingerprinting the electromagnetic emanations of integrated circuits. In contrast to most previous work, we do not evaluate our proposal using simulations but we rather conduct experiments with an FPGA. We investigate the effectiveness of our technique in detecting extremely small Hardware Trojans located at different positions within the FPGA. In addition, we also study its robustness to the often neglected issue of variations in the test environment. The results show that our method is able to detect most of our test Hardware Trojans but also highlight the difficulty of measuring emanations of unrealistically tiny Hardware Trojans. The results also confirm that our method is sensitive to changes in the test environment.

I. INTRODUCTION

Ensuring the authenticity and integrity of hardware components is becoming one of the major challenges faced by the integrated circuit (IC) industry. Counterfeiting of hardware components has experienced a dramatic increase in the last years, up to a point in which product counterfeit is estimated somewhere between 5% and 20% [1]. At the same time, the globalization of the IC market and the ever growing need for cost reduction compromise the integrity of the IC manufacturing and supply chain. Semiconductor companies routinely outsource chip fabrication to offshore foundries and integrate closed 3rd-party IP cores into larger System-on-Chip (SoC) designs, leading to a loss of control not only during the design process but in particular during the manufacturing process.

In this context counterfeits, relabeled parts, and maliciously modified parts are emerging as one of the major threats for embedded device security [2], [3], [4], [5]. The term Hardware Trojan (HT) describes an intentional (and malicious) modification of an IC with the aim of altering its intended functioning. Scenarios that are most sensitive to HTs are those that employ ICs in security or safety related applications, e.g. military, health care, aviation, communications, power management, and in general critical infrastructures. The purpose of an HT can be manifold: altering the functionality of a device, reducing its reliability and/or availability within a system, providing a backdoor for attackers, or enabling the leakage of sensitive information. The literature provides several taxonomies, see for instance [6], [7], [8]. HTs can be categorized according to multiple parameters, including insertion phase, effect, location, size, etc., leading to thousands of possible variants. Independent of the category, a common characteristic and requirement for all HTs is stealthiness, i.e. the inserted or modified circuitry should be difficult to detect.

HTs are typically composed of two different parts: a trigger and a payload. The HT trigger is responsible for activating the malicious functionality at a particular moment. In some sense, it acts like an always-on sensing circuitry that waits for the occurrence of a particular event (internal or external). The HT payload contains the actual circuitry to execute the malicious functionality and, as opposed to the HT trigger, it remains passive for most of the time.

One typically wants to detect an HT before its payload gets activated, and hence aims to detect the HT trigger. There exist two main categories for HT detection methods. The first one comprises destructive techniques and involves invasive hardware “reverse-engineering” [9]. This approach is not only extremely complex and expensive, but also time-consuming and error-prone. It involves for instance chip de-layering, optical and/or scanning electron microscope (SEM) imaging, and netlist reconstruction (a low level description of the circuit). HTs can then be detected by comparing the reverse-engineered netlist with the Golden one. An example of reverse-engineering an IC in the context of HT detection is given in [10], where the authors additionally propose a method to perform the annotation and schematic creation steps by means of machine learning techniques. Methods in this category have a good detection rate, but they can be applied only to samples rather than entire batches (since the techniques are first of all destructive).

The second category comprises non-destructive techniques. It can be further divided depending on whether the testing is logical or physical. Logic testing applies test patterns and aims to hit the trigger mechanism to reveal the presence of an HT by sensing the now activated payload, see for example [11], [12]. The reliability of this approach is clearly limited, not only due to the stealthiness of HTs but also due to the possibly extremely large search space of activation events. In addition, it requires not only to hit the trigger mechanism but also to notice the effect of the activated payload.

A different approach is followed by physical analysis techniques. The overall ideas are highlighted in [13] (Chapter “Hardware Trojan Detection”), where the authors argue that any malicious insertion or modification of a hardware design (such as an HT) should be reflected in changes of physical properties (so called side channels) such as leakage current,
quiescent supply current, dynamic power trace, data path delay, or electromagnetic field characteristics. The main advantage of side channel based approaches is that the presence of an HT can be tested without the need to hit its trigger. A common requirement is, however, the availability of a so-called Golden circuit or Golden model, i.e. an IC or a simulation model known to be HT free that serves as reference.

In this work we propose a new HT detection methodology belonging to the second category, based on physical analysis. In particular, we aim to detect HTs in a circuit by fingerprinting its electromagnetic (EM) emanations, statistical analysis, and comparison to a Golden fingerprint.

A. Related Work

A practical HT detection method using EM radiation is given in [14]. The author successfully identifies HTs using a standard laboratory EM setup (consisting of near-field magnetic probes ETS Lindgren 7405 of 1 cm diameter and a preamplifier from 100 kHz to 3 GHz). However, we note that the conditions are very favorable for successful HT detection because the HT design details as well as the exact time when the HT is activated are known in advance. This may not be the case in a typical HT detection scenario.

Several approaches to increase the sensitivity of the HT detection methodologies using side channels have been investigated. In [6] the authors propose a general method of modeling the noise for increasing the HT detection sensitivity. In [15] the HT detection problem is formulated as a signature outlier identification problem, and the HT detection problem is solved by comparing each signature with an estimated value of other signatures. The method is agnostic with respect to the specific side-channel used (instantaneous power, electromagnetic radiation, etc.). It has the advantage of being somewhat resistant to process variations, it is scalable, and it requires only a Golden netlist for reference, not a Golden IC. The experiments are conducted with HSPICE simulations of post-layout designs and process variations are simulated using Monte-Carlo techniques.

In other cases, spatial partition-based approaches are proposed to allow more local analysis either by using “add-on circuitry” like power ports [16], separate voltage rails [17], or by using chosen input vectors like in [12], [18]. Alternatively, the partitioning can be done on the temporal level as proposed in [19] in order to tackle the problem of process variations. This method combines the current signature of a chip at two different time windows to “completely eliminate the effect of process noise”. The authors argue that their technique provides high detection sensitivity for HTs of various sizes, and stress that it does not require golden ICs as reference. The reported results are based on HSPICE simulations and experiments with an FPGA.

Most of the drawbacks of the proposed methods for detecting HTs using side channels have been summarized in [20]. The main points of this paper are that a) “methods from previous works are not robust with respect to process and test environment variations and therefore cannot reliably detect very small HTs”, and b) almost all previous works lack a thorough experimental section and hence their usefulness is not very clear. The authors analyze the experiments carried out in the literature, and identify two main shortcomings: a) unrealistic, abnormally large HTs are used and b) HTs are inserted at gate level or even at register-transfer level (a high-level representation of a circuit). Such experiments may not reflect a realistic situation, where a small HT would be inserted at low level by an untrusted foundry after place and route in order to hide it. This insightful observation shows that the reported detection capabilities of previous works should be taken with due care.

One of the rare cases where a “realistic approach” based on a “realistic scenario” is described is in [21]. In this paper, the authors consider a cryptographic ASIC designed using a 180nm UMC process and implementing the Advanced Encryption Standard (AES) [22]. The HT occupies a relatively small part of the IC: 0.5% in chip area (190 GE). The first part of the analysis consists in building power templates by calculating the mean of all power traces from each ASIC. After that, the authors calculate the difference of means (DoM) as a first approximation towards distinguishing “trojanized” ASICs from Trojan-free ASICs. The second step consists in applying principal component analysis (PCA) to the mean traces of each device, to provide less redundant information. The output of this step is fed to a Support Vector Machine (SVM) classifier. This experiment assumes that for the training of the SVM classifier, chips without and with an HT are available. Hence this method is limited to detecting known HTs.

In contrast to that, most academic works on side channel based HT detection use FPGAs as test vehicles instead of ASICs for obvious practical reasons.

B. Contributions

We propose a new HT detection technique that is based on fingerprinting an ICs EM emanations. As all physical HT detection techniques it requires the availability of a Golden reference, in our case a Golden circuit, but it does not require to fingerprint HTs before being able to detect them. In addition and perhaps more importantly, we evaluate our technique in practical experiments rather than simulations and study its robustness, e.g. to test environment variations. Further, we intentionally study extremely small and supposedly hard to detect HTs to explore our technique’s limits. Our experiments are work in progress and we present results that are sometimes not yet fully conclusive.

II. Testing Approach

In this section we describe our proposed HT detection methodology based on measuring and analyzing EM waves emitted by the so-called Device under Test (DuT) during its normal operation.

Our methodology involves two different phases:

- A learning phase, in which we collect a sufficient amount of EM measurement data from a Golden device in order to characterize some particular attributes of its distribution (to generate a Golden fingerprint).
- A matching phase, in which we collect EM measurements from the DuT and apply a statistical test to determine whether the collected data comes from the
same distribution as the Golden fingerprint (the DuT is trojan-free) or not (the DuT is infected).

The effectiveness of the methodology significantly relies on two parameters. First, the amount of side channel leakage that can be captured from both Golden device and DuT as well as the quality of the measurements. And second, the selection of a suitable statistical test that is both robust and reliable, and allows for fast evaluation. Additionally, it is desirable that the test is able to quantify the confidence we can have in the result.

Our method uses a balanced approach in which a similar number of measurements of the Golden circuit and the DuT must be obtained. In addition, it is important that the measurement conditions are as similar as possible. One is left with two options: either measure the Golden device again and again every time a DuT needs to be tested, or obtain measurements of the Golden device once and for all and try to meet the measurement conditions whenever measuring a DuT. The first approach should intuitively provide better results, but the second approach appears more practical. The challenge in the second approach is to meet the measurement conditions. Hence an important contribution of our work is to challenge the robustness of our method.

After collecting side channel measurements from the DuT, our methodology requires to apply a statistical test to compare the measurement data with the Golden reference data. The main idea is that the EM emissions of a DuT without HT should have similar characteristics as the Golden reference whereas emissions of a DuT with HT should be less similar.

**Statistical test.** We use Welch’s two-tailed T-test to test the hypothesis that the two sets of measurements (from the Golden device and from the DuT) have equal means. The test computes a t-score for each time sample in the measurements as follows:

\[
t = \frac{\mu_0 - \mu_1}{\sqrt{\frac{\sigma^2}{N_0} + \frac{\sigma^2}{N_1}}},
\]

where \(\mu_0\) is the sample mean of the Golden measurement set, \(\mu_1\) is the sample mean of the measurement set from the DuT, \(\sigma^2\) is the sample variance of a set, and \(N\) is a set’s cardinality.

The degrees of freedom \(\nu\) can be approximated with the Welch-Satterthwaite equation. We can use \(t\) and \(\nu\) together with the t-distribution to test the hypothesis and compute a p-value which may or may not give evidence to reject the null hypothesis (that the means are equal). However, we will not compute p-values but work with the t-scores directly. First experiments showed that this is sufficient since we need to carefully analyze the results and tune the thresholds anyway.

### III. Test Environment

**Experimental Setup.** Our experimental setup is depicted in Figure 1. Our platform is a Sasebo-G board [23] specifically designed and developed to enable research on hardware security aspects. The board features two Xilinx FPGAs: a Virtex-II Pro XC2VP7 (referred to as target FPGA) and a Virtex-II Pro XC2VP30 (referred to as control FPGA). Both FPGAs are internally connected via 36 pins. The target FPGA is the circuit to be measured and acts as both the Golden device and the DuT depending on its configuration, while the control FPGA acts as interface between the target FPGA and a control PC.

We place the Sasebo-G board on a X-Y-Z positioning system and attach it with a custom jig to prevent any movement. We position a near-field probe above the target FPGA to measure its EM emissions. In order to obtain a good signal strength it is important that the probe is as close as possible to the DuT. In our experiments we did not open the DuT’s package but the probe tip touches the package. Getting even closer to the IC would provide an even better signal but it would also require to open the package which would render the method much less practical. The signal measured by the probe is amplified, captured by a digital sampling oscilloscope, and transferred to the control PC for later analysis.

**Golden Circuit.** Our Golden circuit is an implementation of the 128-bit version of the AES block cipher. Cryptographic algorithms are found at the core of most security-related applications, and as such they are attractive targets for attackers. However, and as will become clear when we describe the HTs that we implemented, our method is not limited to detecting HTs in cryptographic circuits.

Our implementation is written in Verilog. The input operands, namely 128-bit plaintext and 128-bit secret key, are provided by the control FPGA to the target FPGA via 8 dedicated interconnection pins, i.e. in 8-bit words and one word per clock cycle. The Golden circuit performs the AES encryption after all input data has been received. After that, it sends the output ciphertext back to the control FPGA using 8 different dedicated interconnection pins.

We use Xilinx ISE Design Suite 10.1 to synthesize and place and route our designs. The Golden circuit utilizes 537 flip-flops and 3 402 look-up tables (LUTs), which altogether correspond to 2 071 slices, i.e. a resource occupation of 42%.

**HT Infected Circuits.** We have implemented several circuits infected with HTs in order to carry out our experiments. All infected designs are modifications of the same Golden reference, i.e. the HT circuitry is directly inserted in the AES-128 implementation. We take as a starting point the Native Circuit Description (NCD) resulting from the place...
and route process and we perform low-level changes using the Xilinx FPGA Editor software. This tool enables manual modification of the FPGA elements, e.g. components, nets, or LUT equations, to name a few. Inserting the HTs at this low level is a laborious task but it results in minimal changes to the circuits netlist and resembles what a clever adversary might do in a chip foundry. In contrast, inserting the HT in the high level Verilog description results in a completely different NCD file due to synthesis, mapping, place, and route.

All HT circuits implemented in this work are externally triggered, i.e. the HT trigger circuitry is directly connected to the 8 input lines from the IOBs (Input/Output Blocks) in order to detect a predetermined activation sequence. The HT activation sequence can be expressed as a Boolean equation and implemented by means of LUTs. Note that it is possible to implement an 8-bit comparator occupying only 3 LUTs (or 1.5 slices) in the target FPGA.

In a very conservative setting, this approach would already allow us to insert a very simple externally triggered HT circuitry occupying only 2 slices (out of 2071 + 2). In practice, however, we expect any HT trigger to be larger. Selecting an 8-bit activation sequence defeats the purpose of the HT being stealthy, because such a short sequence is trivial to detect by logic testing. Therefore, one would expect a realistic activation sequence to be considerably longer, e.g. around 80 bits, with a negligible probability of being hit by chance. We have opted to implement such an 80-bit comparator in our infected circuits. As the data is sent byte-wise from the control FPGA to the target FPGA, the comparator needs a small state machine to keep track of the comparison. We obtain a total size for the HT circuitry of 27 slices (15 for the comparators and 12 for additional logic for the finite state machine). The size of the HT trigger accounts for 1.3% of the whole circuit size.

In order to keep the HTs very small, we have opted to not implement any HT payload. In other words, our infected circuits do not react to the activation sequence. The reason is again rather intuitive: any HT payload circuitry will only result in a larger HT size and, consequently, will likely increase the chances of the HT being detected. By implementing a small HT trigger with minimal changes to the netlist and by omitting the HT payload, we are making our HTs very hard to detect and therefore challenge the detection capabilities of our method.

We have implemented six different, yet very similar variants of HT infected circuits. The FPGA floorplan of each variant is depicted in Figure 2. Grey dots represent unused slices, blue dots represent slices occupied by the Golden circuit, and the additional HT circuitry (27 slices plus some routing) is represented in red. In the first three circuits we place the HT circuitry in empty spots along the upper side of the FPGA (HT a: left, HT b: center, HT c: right), while in the last three circuits we place it symmetrically along the lower side (HT d: left, HT e: center, HT f: right). In all cases we let the FPGA editor tool route the input signals from the IOBs to the LUTs.

Measurement Collection. Following the methodology introduced in Sect. II, we have collected multiple sets of EM measurements for all different FPGA designs, i.e. Golden circuit (during learning phase) and infected circuits (during matching phase). We took these measurements at different times of the day (and deliberately did not use the temperature control in our lab) and with more or less time in between them. All measurements are obtained by using a single setup as depicted in Figure 1.

Each EM measurement corresponds to one execution of the AES cipher. It covers a time window ranging from input data reception to output ciphertext transmission. In other words, it captures the EM emanations during both I/O communication and encryption using AES. The control PC is configured to provide fixed inputs to the target FPGA, i.e. for each measurement input data and encryption key are the same. This choice reduces variations of the EM emissions due to varying data. The oscilloscope’s sampling frequency is set to 250 MS/s and the FPGA designs run at 3 MHz. With these parameters, one measurement is 5 000 samples long.

We used a magnetic near-field microprobe (Langer ICR HH 500-6) capable of detecting magnetic fields emerging vertically from the DuT’s surface. The measuring frequency range of the microprobe goes from 500 kHz to 6 GHz. The probe tip diameter is 500 µm. For each design, we began by positioning the EM probe close to the top left corner of the FPGA surface and gradually moved it on the X and Y axes in steps of 500 µm. The total scanned area is represented as a matrix of 17 rows and 21 columns. At each position we took a set of 4 000 measurements. Overall, for every circuit we need to collect and analyze 17 × 21 × 4 000 measurements of 5 000 samples each. This amounts to roughly 5 GBytes of data and a collecting time of about 1.5 hours per design. The computation of the results is comparably cheap and can be done in a matter of minutes.
IV. RESULTS

A. Golden vs Golden

For the first experiment we collect three independent sets of measurements of the Golden circuit. Figure 3 shows the results obtained when comparing the first and second sets (left) and the first and third sets (right), respectively. We present the results as a $17 \times 21$ bitmap where each point corresponds to an EM probe location above the FPGA. For each location we obtain a (time domain) vector of t-test scores. We reduce this vector to its highest absolute value and assign that value to the corresponding point in the bitmap. Warm colors indicate statistically more significant differences and cold colors indicate less significant differences (dark blue indicates no significant difference). Note that all plots except those in Figure 3 use the same scale for colors and can hence be easily compared.

![Fig. 3: Results for two variants of Golden vs Golden.](image)

Intuitively, one expects the plots to show no or only little significant differences. As the measurements in each set originate from the same underlying circuitry, their distributions should be rather similar and, consequently, the t-test scores should be quite low. The two sets of measurements that we used for the plot on the left hand side were taken immediately one after the other. We think this is the reason why the plot shows no significant warm or hot zones, but mostly cold noise. In contrast, the plot on the right hand side shows warm noise (green, yellow, orange) and two hot zones (red). The two sets of measurements that we used for this plot were taken on two different days and at different times of the day. We think that differences in room temperature cause the high t-test scores (which forced us to use a different scaling for these plots).

B. Golden vs HTs

For the second round of experiments we collect a set of measurements for each of the six different HT infected circuits described in Section III. Figure 4 shows the resulting bitmaps when comparing them against the Golden circuit measurements (obtained at a similar room temperature) using our proposed method. A quick look at the plots already reveals significant differences with respect to the results in Figure 3 (recall that the color scaling differs). The noise level is generally low and a few hot zones can be appreciated at different locations within the plots, indicating that the underlying circuitry in the designs differs at those particular locations.

The plots corresponding to HT a and HT b appear to be quite similar. At first glance one can notice a common, horizontally shaped hot spot in the middle part of the bitmap concentrating the most significant differences. A second roundly shaped warm area can be seen in the upper side of both bitmaps, albeit at different positions. For the case of HT a it appears to be slightly shifted to the left. We attribute this to our method successfully detecting the presence of the HT trigger circuitry. Recall that for HT a the comparators are placed in the upper left corner, while for HT b they are located in the upper center section. This intuition can be confirmed by looking at the remaining plots. The bitmap for HT c exhibits a hot zone in the upper right area, while the location of the warm area in the plots for HT d, HT e, and HT f is clearly in the lower area.

![Fig. 4: Results for Golden vs HT infected circuits.](image)

Based on our analysis we assume that the horizontally shaped hot spot in the middle part of the bitmaps for HT a and HT b is caused by horizontal signal routings. More precisely, observe in Figure 2 that only HT a and HT b have horizontally routed signals in the upper half of the FPGA. We further assume that the roundly shaped warm areas in all six plots are caused by our HT circuitry.

C. Golden vs smaller HTs

To check our assumption and to further challenge our method’s ability to detect extremely small HTs, we repeat the above experiment but this time implement the unrealistically tiny 8-bit comparator as HT trigger and again no HT payload. In other words, we reduce the HT circuitry from 1.3% to 0.1% of the whole circuit. The signal routings from the IOBs to the HT circuitry follow the same paths as shown in Figure 2. For this third round of experiments we again collect a set of measurements for each of the six different HT infected circuits and compare them to measurements of the Golden circuit. Figure 5 shows the resulting bitmaps.

In contrast to the plots in Figure 4 only the plots for HT a and HT b show warm and hot zones that indicate the presence of an HT. The plots for HT c to HT f show only cold noise and hence our method is unable to detect these HTs. The warm and hot spots in the plots for HT a and HT b look very similar to those in Figure 4. Following the previous analysis we assume that, for these extremely small HTs, our method is unable to detect the HT trigger circuitry but detects the horizontal HT signal routings of HT a and HT b.
The Research Foundation - Flanders (FWO).

317930 HINT. Benedikt Gierlichs is a Postdoctoral Fellow of and by the European Commission FP7-ICT-2011-8 grant

method’s robustness to changes in the test environment more investigate this further. As future work, we plan to evaluate our (partially) shielded by the above layers. It is, however, hard to
detect because they are on the top metal layer. Other signals may be routed on lower metal layers whose emanations are
detect because they are on the top metal layer. Other signals may be routed on lower metal layers whose emanations are (partially) shielded by the above layers. It is, however, hard to investigate this further. As future work, we plan to evaluate our method’s robustness to changes in the test environment more comprehensively.

V. CONCLUSION

We proposed a novel HT detection technique based on fingerprinting the EM emanations of integrated circuits. We evaluated our technique in practical experiments with a Golden circuit and HTs that we implement in an FPGA. We studied its robustness to variations in the test environment and showed in particular that changes in the temperature have a negative effect. We further demonstrated that our method is able to detect extremely small HTs located at six different positions within the FPGA. Our analysis also suggests that specific signal routings present in some of the HTs are easier to detect than others. In addition, we experimented with unrealistically tiny versions of the same HTs and found that our technique is able to detect only those with the specific signal routings. We speculate that those signal routings may be easier to detect because they are on the top metal layer. Other signals may be routed on lower metal layers whose emanations are (partially) shielded by the above layers. It is, however, hard to investigate this further. As future work, we plan to evaluate our method’s robustness to changes in the test environment more comprehensively.

ACKNOWLEDGMENT

This work is supported in part by the Flemish Government FWO G.0550.12N, by the Hercules Foundation AKUL/11/19, and by the European Commission FP7-ICT-2011-8 grant 317930 HINT. Benedikt Gierlichs is a Postdoctoral Fellow of the Research Foundation - Flanders (FWO).

REFERENCES


Fig. 5: Results for Golden vs smaller HT infected circuits.