Introduction to Hardware Trojan Detection Methods

Julien Francq
Airbus Defence & Space – CyberSecurity,
1 Bd Jean Moulin,
CS 40001,
MetaPole,
78996 Elancourt Cedex,
France,
Julien.Francq(at)cassidian.com

Florian Frick
University of Stuttgart,
Institute for Control Engineering of
Machine Tools and Manufacturing Units (ISW),
Seidenstrasse 36,
D-70174 Stuttgart,
Germany,
Florian.Frick(at)isw.uni-stuttgart.de

***Abstract*—Hardware Trojans (HTs) are identified as an emerging threat for the integrity of Integrated Circuits (ICs) and their applications. Attackers attempt to maliciously manipulate the functionality of ICs by inserting HTs, potentially causing disastrous effects (Denial of Service, sensitive information leakage, *etc*.). Over the last 10 years, various methods have been proposed in literature to circumvent HTs. This article introduces the general context of HTs and summarizes the recent advances in HT detection from a French funded research project named HOMERE. Some of these results will be detailed in the related special session.**

I. Introduction

A. General Context

The integrity of Hardware is an absolute necessity for any system. To a great extent, this integrity has never been called into question. However, changing development and especially changing production processes have necessitated the need for a closer examination. Malicious manipulation of Hardware, referred to as Hardware Trojan Horses (HT / Hardware Trojans) is quickly becoming an emerging threat.

One factor for this stems from the production process. While in the past, hardware was largely produced in-house or in trusted fabs, nowadays, the production is increasingly globalized and outsourced. Therefore, the risk of adversaries manipulating the design before or during the production process has also increased.

B. Malicious Payloads of HTs

An important aspect to consider is the intended effect of the HT. Potential aims include the reduction of the reliability or even the destruction of a system, the implementation of a backdoor in order to leak secret information or a change of the functionality. Compared to traditional software based attacks, HT attacks are usually much more complex, which consequently necessitates a higher degree of knowledge and manpower. Due to the advanced nature of HTs, attacks are not suspected to be undertaken by the classic Hacker or Script Kiddie, but rather by larger, organized attackers. Companies, governmental institutions, organized criminals or terrorist organizations are much more likely to have the necessary resources to perform such an attack. As the methods will vary, so too will the motivations behind an attack. Perhaps it is in the intent of the attacker to sabotage a competitor by reducing the lifetime and quality of their product.

Most alarmingly are those potential attacks concerning security. If the backdoors and stolen data belong to critical systems, civil, security-related or even military, needless to say, the potential security risk is dire and danger to health and life, potentially fatal. Within this context is the idea that ”Kill Switches” might be used ([1]). The basic premise being that the designer, the production company or even a third-party could implement a HT that would give them the power to deactivate the system. The concept of Kill Switches is particularly relevant within military equipment, since one must always consider the dual nature of selling such equipment. It always poses the risk that the technology could be used against you. Therefore, a way to deactivate it when necessary would be a valuable feature for the producer.

C. HTs: A Real Threat for ICs?

HTs were considered publicly as a serious risk in 2005 by a report written by US Department of Defense. This report made the Defense Advanced Research Projects Agency (DARPA), the R&D wing of the Pentagon, launch the “Trust in ICs program” in 2007, followed by the “Integrity and Reliability in Integrated Circuits” (IRIS) project (2011–2014). These projects had the goal of finding innovative HT detection methods. Most of the results remain confidential today. Military applications are the most exposed due to the fact that they need a high safety level. It is not only a theoretical threat since some HTs in ICs used by military equipment have been reported [1] [2]. Besides military domain, universities have also recently increased their interest in HT domain and some scientific conferences now regularly schedule sessions on this topic (DATE, CHES, HOST, *etc*.). There is also a 3-year (2012–2015) French-funded research project named HOMERE that studies HT detection methods.

In summary, we can consider that HTs are a real and emerging threat.

978-3-9815370-4-8/DATE15/©2015 EDAA

D. Counteracting HTs

The key challenge is how to verify that the delivered product has not been tampered. Since ICs are highly integrated and very complex, there are no straightforward or economical ways to check a delivered IC. Direct checking by analyzing the IC can only be performed in a destructive way [3] and is only valid for the tested chips and cannot guarantee the integrity of all.

Therefore there is a need for non-destructive test methods which can be applied to every single chip. These are either based on logic testing or Side-Channel Analysis (SCA).

Logic testing [4] [5] tries to find an output which does not fit the original design. While this can potentially prove the presence of a HT, the chances to detect a HT highly depends of the complexity of its trigger.

Side-channels (*e.g.*, power consumption, electromagnetic radiations of an IC) [6] [7] [8] [9] [10] [11] [12] [13], so far mainly known in the context of Side-Channel Attacks [14], reveal information about the interiority of a circuit and have therefore the potential to be used for HT detection. SC-based approaches are very promising but also faces some intrinsic difficulties.

Obfuscation can be seen as one kind of preventive method to prevent stealthy HTs insertion. An attacker needs to have a good knowledge of the IC he targets in order to induce an efficient HT, in particular knowledge of low controllability and observability nodes. If the defender obfuscates its IC (e.g., with the help of functions which uses a key, unknown to the attacker), an attacker will have high chance to induce benign HTs or HTs that will be easily detected by traditional logic testing since they will have higher chance to trigger in test-time.

E. Outline of the Paper

This paper introduces the general context of HTs and gives a summary of the recent advances in the Hardware Trojans detection methods discovered by the French funded research project HOMERE.

In the next section, an overview of the possible infection scenarios and the taxonomy of the HTs will be depicted.

In the third section, known HT detection methods will be shown.

This paper will end with a summary of the main results of HOMERE project until now.

II. HT Insertion Scenarios and HT Taxonomy

Hardware Trojan Horses (HT) are "intentional, malicious manipulations of digital circuits". Beside this common property, HTs could be induced at different steps of the (complex) IC design flow. We will also see in this section that the HT taxonomy is very rich.

A. HT Insertion Scenarios

As mentioned in the introduction, a very critical point in the development process is the chip production in untrustworthy fabs. But there are many other possibilities where a HT could infect a system during the hardware development process. Based on the process introduced earlier, the possible risks are analyzed in this section. Figure 1 gives an overview.

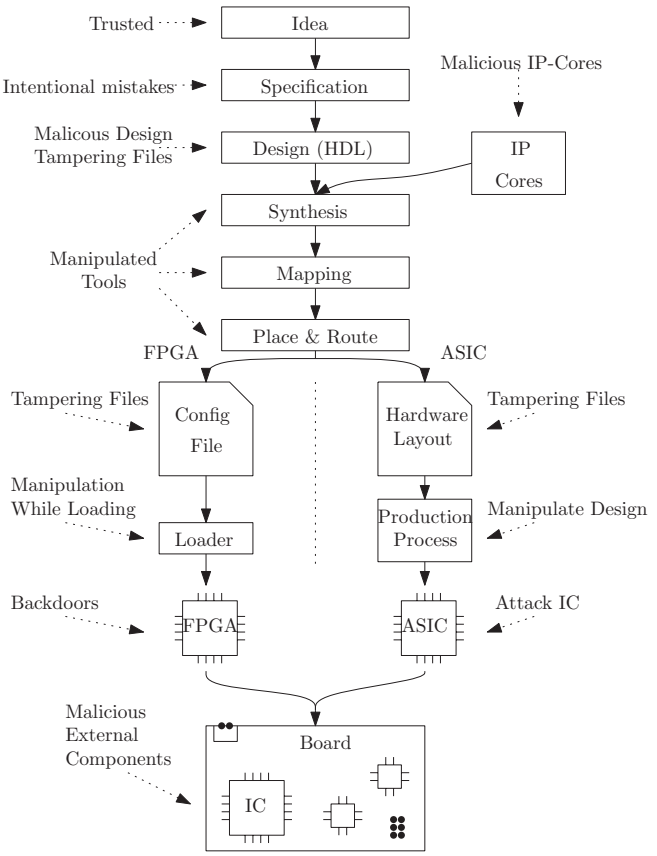

Fig. 1. Hardware Development Process with Attack Scenarios

- **Idea:** The idea is the trusted starting point.
- **Specification:** The specification is not necessarily done by the trusted party itself. By intentionally specifying certain aspects wrong, a HT could be inserted. For example, if the specification is written by a malicious third party who defines a key size that is too weak.
- **Design:** The design phase offers many opportunities to tamper the design. The implementers themselves are an obvious risk. But even when they are perfectly trusted, the source files are still a potential target. Someone who is able to manipulate them could insert virtually any change desired. Since it is unlikely that design files are checked repeatedly once verified, the chances of detection are quite low.
- **IP-Cores:** IP-Cores which are obtained from third parties could be malicious as well. Heavily depending on the purpose of the design, the HT might be purely internal, for example, a backdoor to access the component or a time bomb. External effects are conceivable as well, for example a component connected to a bus could access and manipulate other components.

- **Tools (Synthesis, Map, Place and Route):** The tool chain also offers various opportunities to attack the design flow. A simple example is that a kind of filter is inserted in the chain, which replaces a genuine component by a malicious version. Another possible way is by the reconfiguration of parameters, ranging from timing conditions to driver strength configurations.
- **HW-Layout (ASIC):** The HW-Layout can be compromised as well. An attack on the way from the designer to the producer is a possibility, as well as an attack directly performed by the producer itself.
- **Production (ASIC):** During the production process, for example, the masks could be tampered.
- **ASIC:** Even the produced ASIC itself is not safe. One very simple but potentially efficient attack is the degradation of quality and lifetime, which could, for example, be achieved by treatment with heat. Much more sophisticated, very complex and expensive, yet effective is a post-production manipulation by the use of Focused Ion Beam (FIB). This technology is usually used to manipulate ASIC prototypes and is perfectly capable of changing logic. One frightening aspect of this attack is that even verified hardware could be manipulated later.
- **Configuration File (FPGA):** The configuration file contains all the information necessary to configure the FPGA. Therefore, manipulating it offers all possibilities. Depending from where the file is stored, many different scenarios are plausible.
- **Programmer (FPGA):** The Programmer of an FPGA should program the FPGA with the configuration file inputted. One scenario is that the programmer tampers with the design. Especially critical is the fact that there is often no way to verify the design on the FPGA after configuration.
- **FPGA:** The FPGA itself could be malicious (The most known HT case so far). Backdoors are especially a high potential risk.
- **Target System:** The other components connected to the IC could also introduce a risk. For example, components connected to an internal bus could be abused. Also, components that appear to be harmless and non-critical can be used for malicious purposes. For example, a power source which introduces glitches sporadically could be used to cause errors and therefore enable a fault attack.

As a result of all of the potential situations described above, it is exceedingly difficult to pinpoint one clear, precise definition for Hardware Trojans. The transition to simple errors as well as to other kind of attacks is not clear.

B. HT Taxonomy

1) Logic vs. Non-Logic Modifications: Many of the HTs mentioned so far are based on the manipulation of the logic function of the system, which is usually done by either adding, removing or manipulating gates.

The logic does not necessarily need to be changed in order to implement a HT since it could also be done by parametric manipulation or non-logic design changes. Parametric changes could affect, for example, the driver strength or the wire size.

Non-logic design changes could be the integration of circuitry which does not affect the logic. Also the existing logic can be redesigned in a way that the logic function is not affected.

2) Trigger Condition: The payload of HTs could be continually active or activated by a certain event, *i.e.* a trigger condition. Triggers are usually based on rare conditions in order to avoid an accidental activation. A rare condition is a signal or a signal combination which occurs quite seldom.

Combinatorial Trigger: A combinatorial Trigger is based on one or more, usually rare, input signals. If those signals fulfill a certain condition, the trigger is active. If the input signals change, the trigger is deactivated again.

Sequential Trigger: A sequential trigger has states and therefore requires memory. Typically, state changes are triggered by a rare event. By increasing the number of stages, an arbitrary complex trigger can be constructed.

A trigger needs not necessarily to be a logical function, but could also be based on a physical property like temperature, voltage or clock frequency.

Which kind of trigger is selected heavily depends on the purpose of the HT. The following list gives some examples based on the intent.

Activate Backdoor: Rare condition which is unlikely to appear in normal use, but can be easily forced on purpose.

Random Failure: Trigger based on random but rare events.

Systematic Failure: Trigger based on a precise condition.

3) Effect: An important aspect is the intended effects of the HT. Possible aims include reduction in system reliability, the implementation of a backdoor in order to leak some secret information, a change of the functionality, or even complete destruction of the system. A HT could also be designed in a "Stand Alone" manner, meaning the malicious aim is fully implemented in it. On the other hand, a HT could also serve as a support for another attack, e.g. a software-based one.

III. HT Detection Methods

Since the emergence of HT threat, many ideas have been proposed in the attempt to solve the issue. Varying widely, the approaches differ in basic approach, targeted type of HT, cost and reliability. This section gives an overview of all the existing approaches.

A. Overview

There are different kinds of detection mechanisms that vary under certain criteria. Some require an adaption in the hardware design process or additional circuitry, while others do not induce any change. There are approaches which continuously monitor the system during runtime; others are just applied once after the production of the chip to verify its integrity. Certainly, all of these approaches have different advantages and disadvantages and are designed for different kinds of HTs. Figure 2 gives an overview of the available detection methods.

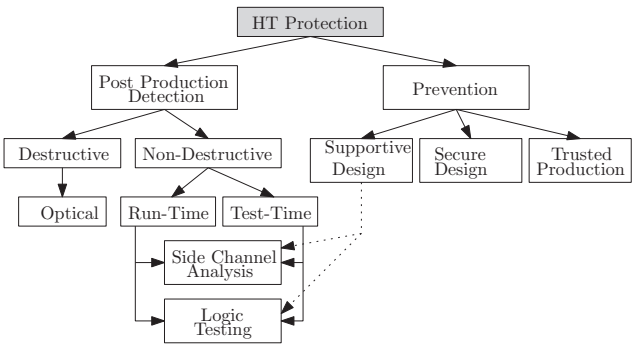

Fig. 2. Overview HT Detection Methods

Looking at the approaches to verify chip integrity after the production process, two main techniques are available: Logic testing and Side Channel Analysis (SCA).

Logic testing is based on comparing the actual outputs of the chip to the expected outputs, derived from a model. This technique is intrinsically difficult to apply due to the overwhelming number of possible inputs and internal states. The fact that HTs can have a sequential structure makes this approach even more complicated.

B. Destructive vs. Non-Destructive

It is possible to verify a chip design by optical methods. In order to perform such an analysis, the chip must be opened and analyzed layer by layer. To reach the lower layers, the higher layer needs to be removed. After which, the chip is destroyed.

At the end of the analysis, the question of if the chip was infected can be definitively answered, but the result is only valid for the tested chip. It does not allow for any conclusions on the untested chips. Using statistical approaches may help, but since a malicious attacker is assumed, statistical models help only to a limited extent. Depending on the attack scenario, a single chip within thousands can be enough to reach the attack's objective.

Besides this theoretical problem, a practical limitation is the cost of such an analysis. Since modern technologies are so small, highly specialized equipment is needed.

For all of these reasons, this approach does not solve the HT issue. Nevertheless, it has contributed important information to the field of HT detection process: If an IC is suspected to be infected by any other method, the optical method can be used to analyze the HT and generate usable evidence.

C. Logic

Logic testing is usually used for functional testing and verification in the post production process. Thereby, a set of input vectors is inputted to the system and the outputs are observed. Errors are detected by comparing the actual outputs to the expected results.

The same approach can be used to search for HTs. One advantage is that if an unexpected logic behavior is detected, the presence of a HT (or at least a production error) is proven.

Besides this, the logic testing approach has some major disadvantages: First, the HT needs to affect a output, which is not necessarily the case. The key problem is that the HT needs to be activated. This can, as stated earlier, be arbitrarily complex. Modern devices have hundreds of I/Os and therefore testing all possible combinations is a very long process. Assuming that the trigger also depends on the internal state of the device or is sequential, it means that it is practically impossible to activate the HT.

Methods are proposed for how this could be improved. For example, calculations are done for how to maximize the activity of rare signals ([5]). Still, the efficiency is questionable.

Nevertheless, these approaches can be useful to support other approaches.

D. Side-Channels

Another basic approach is to use Side Channels to detect HTs ([6]). The basic assumption is that every component contributes to the SC and therefore a change should affect the SC. A difference between other methods is what is done with the SC measurement.

Many approaches are based on a reference SC measurement from a guaranteed HT-free circuit, a so-called Golden Circuit (GC). This assumption is quite ambitious.

There are many different approaches available for how this comparison could be done and optimized ([7], [15], [12], [10], [16]).

A common problem of all SCA based approaches is the presence of noise and device specific production variance. Solutions to overcome this problem are proposed, for example calibration mechanisms ([17], [18], [19], [20], [10], [16]).

IV. Main Research Results of HOMERE Project

HOMERE project has brought advances in HT detection. We will give in this section an overview of the main research results of HOMERE. Some of them will be described in more details in the related special session.

A. Preliminary Work

Before starting trying to detect HTs, the first step is to infect benchmark circuits with HTs. Designing many infected infected ASICs for the project would have been too costly for the project, so it has been decided to use infected descriptions of ICs implemented in FPGAs.

In many papers, the infection step is often described loosely as "we just added one more gate". Certainly, on the Register-Transfer Level (RTL) abstraction layer, this expression is quite unambiguous, but the effects on the resulting hardware are by far not as straightforward.

Firstly, it would make a dramatic difference depending on if this additional gate is added at a hardware description language level or directly to a netlist. Even if the gate is added on the netlist level, this could lead to very different results. The

main reason for this is that the tools used to create the actual hardware are based on the nets. A small change in the design could lead to a completely different result.

Consequently, this poses the question of if the detected anomalies are actually caused by the single additional gate or are a result of the overall change in the design (different wire length, *etc*.).

To overcome this problem, Marchand *et al*. [21] have used configuration tools (for Xilinx FPGAs, it is FPGA Editor) to infect circuits, which allow to induce minor modifications (at FPGA Slice level) and maintain the initial structure of the IC before infection.

B. Test-Time Methods

1) Optical Methods: Standard destructive reverse-engineering techniques (usage of Chemical Metal Polishing followed by Scanning Electron Microscope image reconstruction and analysis) can be used to detect HTs if we can compare ICs with a "golden" circuit. However, this task is very expensive since it takes a lot of time to realize properly: if the obtained images are blurry for any reason (like bad mechanical or chemical preparation), the reverse-engineering is hard. Moreover, due to Moore's law, ICs will be more and more densely integrated, and so this method will become more and more difficult in the next future. Therefore, there is a need for medium cost HT optical detection methods.

In [22], Bhasin *et al*. showed the efficiency of a medium cost HT detection method if the placement or the routing have been redone by the foundry. It consists in the comparison between optical microscopic pictures of the silicon product and the original view from a GDSII layout database reader. Mesoscopic imaging (obtained with a $\times 150$ optical microscope) is enough for this comparison. Visual exact shape recognition can be used, but an automated method, based on a similarity tool (cross-correlation) is also shown. It appears that any single modification of the sole top-level metal layer can be easily detected visually. It allows to successfully detect a HT which injects a fault in an AES-128 that is exploitable in a specific Differential Fault Attack.

2) Delay Analysis: A HT injected in an IC has high chance to modify internal IC delays. Finding an efficient way to measure them can allow high percentage HT detection. In [23], Exurville *et al.* investigated an internal IC delays measurement using a clock glitch injection tool. The influence of synthesis options and inter die variations on the measurements has been also studied.

3) Side-Channels: Due to process variations and measurement noise, it appeared difficult at first sight to have a high HT detection probability [24]. In [25], Ngo *et al.* assessed the exact efficiency of HT detection methods based on ElectroMagnetic (EM) radiations of the tested IC. This side-channel has been preferred to power analysis for its better spatial and temporal resolution. The authors proposed a metric to measure the impact of the process variations, the exact size of the HT and its location on HT detection. The authors also give the detection probability of a HT as a function of its activity, even if un-triggered. The method has been tested on an AES-128 cryptographic core running on a set of Virtex-5 FPGAs. Three different HT sizes have been implemented for experiments purposes: 0.5%, 1% and 1.7% of the original IC. 3 different HT placements have been tested: within the boundary of AES crypto-processor, outside the boundary of AES crypto-processor and dispersed over the FPGA. The authors showed that the HT detection probability is superior to 90% with a false negative rate of 5% to detect a HT bigger than 1% of the original circuit. Another result is that the impact of HT placement has very little influence on the detection probability.

4) Logic Testing: In [26], Dupuis *et al.* showed a method based on logic testing allowing activating stealthy HTs and then detect their payload in test-time. A new procedure that allows identifying the ideal locations for an HT insertion is proposed. It is focused on finding:

- signals with low controllability,
- paths that are not critical in terms of delay,
- multiple gates combination that are close one to the other in the circuits layout, and close to available space.

At the end, test vectors are generated to excite these discovered locations.

C. Preventive Methods

1) Circuit Congestion: In [22], Bhasin *et al*. analyze the ability of an attacker to introduce a HT without changing neither the placement nor the routing of the cryptographic IP logic. On the example of an AES engine, the authors show that if the placement density is beyond 80%, the insertion is basically impossible. Therefore, this settles a simple design guidance to avoid HT insertion in cryptographic IP blocks: have the design be compact enough, so that any functionally discreet HT necessarily requires a complete re-place and re-route, which is detected by mere optical imaging (and not complete chip reverse-engineering). It is then highly recommended to avoid dead space in an IC in order to prevent HT insertion.

2) Hardware Logic Encryption Technique: In [27], Dupuis *et al*. propose an encryption technique that also helps thwarting HT insertion. Assuming that an attacker will attach a HT to signals with low controllability in order to make it stealthy, the principle of the encryption is to minimize the number of signals with low controllability. This method is part of so-called "Design for Hardware Trust" concept which consists in incorporating into the ICs some features that should improve the HT detectability. This hardware combinatorial encryption technique adds an external key to the circuit so that the circuit operates correctly only if the correct key value is provided. If this key is unknown for an attacker (or can not be deduced because the key is generated by a Physically Unclonable Function – PUF), the attacker can not guess the real interesting signals to connect a HT. Compared to [28], in which the modification of the probabilities of the signals is effective in scan mode, this encryption mode modifies the probabilities also in functional mode.

Three steps are required: finding signals with a low controllability to find the future encryption sites, computing a timing analysis of the IC to find the paths with positive slack time (which allows the countermeasure to have a low impact on the IC operating frequency), and AND/OR gates insertion algorithm which minimizes the number of signals having unbalanced probabilities below a chosen threshold.

The method has been validated on an 128-bit AES and it has been shown efficient in terms of probability changes and overhead (adding 1024 encryption gates represents an area overhead of only 7.3%).

3) IC States Encoding: In [29], Ngo *et al.* proposed the concept of "encoded circuit", as a technique to protect HT insertion. Encoded circuit is based on the theory of codes. It encodes the internal state with a chosen code of security parameter d, such that knowledge of less than d bits of the encoded state reveals no information about the actual state. This method impedes the insertion of HT trigger part, and actively detects HT through payload part. This method also forces an attacker to implement a HT with at least d trigger signals, and if d is sufficiently big, the inserted HT will be visible with optical/visual means.

ACKNOWLEDGMENTS

This project has been funded by the French Government (BPI-OSEO), under grant FUI #14 HOMERE (Hardware trOjans : Menaces et robustEsse des ciRcuits intEgres). The first author would like to thank all the HOMERE partners for their outstanding work all along the project.

REFERENCES

[1] S. Adee. The Hunt for the Kill Switch. In *Proc. IEEE Spectrum*, volume 45, pages 34–39, 2008.

[2] S. Skorobogatov and C. Woods. Breakthrough Silicon Scanning Discovers Backdoor in Military Chip. In *Proc. Cryptographic Hardware and Embedded Systems – CHES*, volume 7428, pages 23–40, 2012.

[3] J. Kumagai. Chip Detectives. In *IEEE Spectrum*, volume 37, pages 43–48, 2000.

[4] S. Jha and S. K. Jha. Randomization Based Probabilistic Approach to Detect Trojan Circuits. In *Proc. IEEE High Assurance Systems Engineering Symposium – HASE*, pages 117–124, 2008.

[5] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia. MERO: A Statistical Approach for Hardware Trojan Detection. In *Proc. Cryptographic Hardware and Embedded Systems – CHES*, volume 5747, pages 396–410, 2009.

[6] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan Detection using IC Fingerprinting. In *Proc. IEEE Symposium on Security and Privacy – SP*, pages 296–310, 2007.

[7] M. Banga and M. S. Hsiao. A Region Based Approach for the Identification of Hardware Trojans. In *Proc. IEEE Workshop on Hardware-Oriented Security and Trust – HOST*, pages 40–47, 2008.

[8] Y. Alkabani and F. Koushanfar. Consistency-based Characterization for IC Trojan Detection. In *Proc. IEEE International Conference on Computer-Aided Design – ICCAD*, pages 123–127, 2009.

[9] M. Banga and M. S. Hsiao. A Novel Sustained Vector Technique for the Detection of Hardware Trojans. In *Proc. IEEE International Conference on VLSI Design*, pages 327–332, 2009.

[10] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey. Hardware Trojan Horse Detection Using Gate-Level Characterization. In *Proc. IEEE Design Automation Conference – DAC*, pages 688–693, 2009.

[11] D. Du, S. Narasimhan, R. S. Chakraborty, and S. Bhunia. Self-Referencing: A Scalable Side-Channel Approach for Hardware Trojan Detection. In *Proc. Cryptographic Hardware and Embedded Systems – CHES*, volume 6225, pages 173–187, 2010.

[12] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, and S. Bhunia. Multiple-Parameter Side-Channel Analysis: A Non-Invasive Hardware Trojan Detection Approach. In *Proc. IEEE Workshop on Hardware-Oriented Security and Trust – HOST*, pages 13–18, 2010.

[13] H. Salmani, M. Tehranipoor, and J. Plusquellic. A Layout-Aware Approach for Improving Localized Switching to Detect Hardware Trojans in Integrated Circuits. In *IEEE International Workshop on Information Forensics and Security – WIFS*, pages 1–6, 2010.

[14] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology – CRYPTO, LNCS*, volume 1666, pages 388–397, 1999.

[15] M. Banga and M. S. Hsiao. VITAMIN: Voltage Inversion Technique to Ascertain Malicious Insertions in ICs. In *Proc. IEEE Workshop on Hardware-Oriented Security and Trust – HOST*, pages 104–107, 2009.

[16] Y. Jin and Y. Makris. Hardware Trojan Detection using Path Delay Fingerprint. In *Proc. IEEE Workshop on Hardware-Oriented Security and Trust – HOST*, pages 51–57, 2008.

[17] R. Rad, J. Plusquellic, and M. Tehranipoor. Sensitivity Analysis to Hardware Trojans using Power Supply Transient Signals. In *Proc. IEEE Workshop on Hardware-Oriented Security and Trust – HOST*, pages 3–7, 2008.

[18] S. Narasimhan, X. Wang, D. Du, R. S. Chakraborty, and S. Bhunia. TeSR: A Robust Temporal Self-Referencing Approach for Hardware Trojan Detection. In *Proc. IEEE Workshop on Hardware-Oriented Security and Trust – HOST*, pages 71–74, 2011.

[19] R. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic. Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans. In *Proc. IEEE International Conference on Computer-Aided Design – ICCAD*, pages 632–639, 2008.

[20] R. Rad, J. Plusquellic, and M. Tehranipoor. A Sensitivity Analysis of Power Signal Methods for Detecting Hardware Trojans Under Real Process and Environmental Conditions. In *Proc. IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, volume 18, pages 1735–1744, 2010.

[21] C. Marchand and J. Francq. Low-Level Implementation and Side-Channel Detection of Stealthy Hardware Trojans on Field Programmable Gate Arrays. In *IET Computers and Digital Techniques*, volume 8, pages 246–255, 2014.

[22] S. Bhasin, J.-L. Danger, S. Guilley, X. T. Ngo, and L. Sauvage. Hardware Trojan Horses in Cryptographic IP Cores. In *Proc. IEEE Fault Diagnosis and Tolerance in Cryptography – FDTC*, pages 15–29, 2013.

[23] I. Exurville, J. Fournier, J.-M. Dutertre, B. Robisson, and A. Tria. Practical Measurements of Data Path Delays for IP Authentication and Integrity Verification. In *Proc. IEEE International Workshop on Reconfigurable and Communication-Centric Systems-on-Chip – ReCoSoC*, pages 1–6, 2013.

[24] G. Di Natale, S. Dupuis, and B. Rouzeyre. Is Side-Channel Analysis Really Reliable for Detecting Hardware Trojans? In *Proc. IEEE Conference on Design of Circuits and Integrated Systems – DCIS*, pages 238–242, 2012.

[25] X. T. Ngo, Z. Najm, S. Guilley, S. Bhasin, and J.-L. Danger. Method Taking into Account Process Dispersion to Detect Hardware Trojan Horse by Side-Channel. In *Proc. Security Proofs for Embedded Systems – PROOFS*, 2014.

[26] S. Dupuis, G. Di Natale, M.-L. Flottes, and B. Rouzeyre. Identification of Hardware Trojans Triggering Signals. In *Proc. Workshop on Trustworthy Manufacturing and Utilization of Secure Devices – TRUDEVICE*, 2013.

[27] S. Dupuis, P.-S. Ba, G. Di Natale, M.-L. Flottes, and B. Rouzeyre. A Novel Hardware Logic Encryption Technique for thwarting Illegal Overproduction and Hardware Trojans. In *Proc. IEEE International On-Line Testing Symposium – IOLTS*, pages 49–54, 2014.

[28] H. Salmani, M. Tehranipoor, and J. Plusquellic. A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time. In *Proc. IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, volume 20, pages 112–125, 2012.

[29] X. T. Ngo, S. Guilley, S. Bhasin, J.-L. Danger, and Z. Najm. Encoding the State of Integrated Circuits: A Proactive and Reactive Protection against Hardware Trojans Horses. In *Proc. ACM Workshop on Embedded Systems Security – WESS*, 2014.