Is Side-Channel Analysis really reliable for detecting Hardware Trojans?

Giorgio Di Natale, Sophie Dupuis, Bruno Rouzeyre

Is Side-Channel Analysis really reliable for detecting Hardware Trojans?

Giorgio Di Natale, Sophie Dupuis, Bruno Rouzeyre
LIRMM - UM2/CNRS
161 rue Ada, 34095, Montpellier cedex 5, France
Email: {firstname.lastname}@blind.review

***Abstract*—Hardware Trojans are malicious alterations to a circuit, inserted either during the design phase or during fabrication process. Due to the diversity of Trojans, detecting and/or locating them is a challenging task. Numerous approaches have been proposed to address this problem, whether logic testing based or side-channel analysis based techniques. In this paper, we focus on side-channel analysis, and try to underline the fact that no published technique until now has proven its efficiency on reliable experiments.**

I. Introduction

An IC fabrication process contains three major steps: (1) design, (2) fabrication and (3) manufacturing test. With ever-shrinking transistor technologies, the cost of new fabrication facilities is becoming prohibitive. Outsourcing the fabrication process to low-cost locations has become a major trend in IC industry. This raises the concern about untrusted foundries in which an attacker can manipulate the manufacturing masks in order to tamper the design with the insertion of malicious circuitry that triggers a malfunction, referred to as Hardware Trojans [19]. Very recent issues arose from the possibility of getting Trojans from untrusted IP vendors [12]. However, this topic is not covered in this paper. Since the fabrication process becomes untrusted, testers / IC vendors have to be able to verify the trustworthiness of the manufactured circuit.

Numerous Trojans detection methods have been proposed over the past years. We focus mainly in this paper on side-channel analysis methods, which consist in searching for degradation in performance or power characteristics modification to detect the presence of Trojan. Whether such kind of method is reliable or not is still an open question. These methods have indeed a major weakness: they are not robust with respect to process and test environment variations and therefore cannot reliably detect very small Trojans. This paper aims at examining all the previously proposed approaches in terms of experimentations, in order to show that almost all methods lack an efficient experimentation to prove their usefulness. Moreover, we will show that a smart introduction of the Trojan would make impossible its detection by using any technique based on delay analysis.

This paper is organized as follows. In Section II, we briefly recall the different sorts of Trojans and the proposed detection methods. In Section III, we focus on prior work concerning side-channel analysis methods. These approaches have various advantages and disadvantages; this section highlights a number of these. In Section IV, we detail experiments authors have made to prove the usefulness of their approach and underline by simple examples the weakness of each experiment.

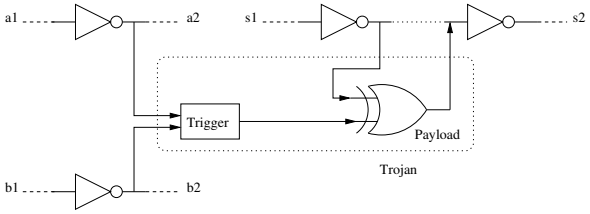


Fig. 1. Trojan circuit model

II. Trojan

A. Trojan taxonomy

Trojans can take many forms. Tehranipoor et al. were the first to propose a Trojan taxonomy [19], [20]. The taxonomy decomposes Trojans according to their physical, activation and action characteristics.

The physical characteristics include the size of the Trojan, its distribution (Trojan dispersed or not across the layout of a circuit) and its type that can be functional or parametric. Functional Trojans modify the functionality of the original circuit (by adding spare logic gates) for rare input combinations, while parametric Trojans target the modification of existing logic (e.g. weakening a transistor or thinning a wire to reduce the lifetime of the circuit) [14].

The activation characteristics refer to the criteria that causes the Trojan to become active. It can be always active (often in the case of parametric Trojans), or triggered by a condition.

The action characteristics refer to the type of behavior induced by the Trojan. It can be either the disruption of the circuit, or a malfunction, or a denial of service.

A more concise and general Trojan architecture is presented by Wolff et al. [21] (see Figure 1). This scheme allows representing functional Trojans, activated by a triggering condition. For most of the input values the circuit will react the correct way, i.e. the trigger is not enabled and the payload has no effects. On the contrary, when the triggering condition is verified (either intentionally by an attacker who knows the triggering condition or accidentally because the circuit elaborated those particular values) the payload will inject an error.

B. Prior work in Trojan detection

Two main categories exist concerning Trojan detection: destructive methods, where the circuit is unpackaged and analyzed by means of "reverse engineering" techniques, or non-destructive methods, which rely on outside measurements without physically tampering the design.

Destructive methods are very expensive and time consuming, and they become even more difficult with shrinking technologies. Although they are more reliable in detecting Trojans, they cannot guarantee the not analyzed ICs to be Trojan free.

Non-destructive methods are categorized as either *logic testing* or *side-channel analysis*. As described afterwards, a combination of both is also an option.

Concerning *logic testing*, the assumption is that an attacker will try to hide the Trojan of IC's functional behavior i.e. a Trojan will be mostly inactive and triggered under very rare conditions. Based on this assumption, the Trojan detection methods aim at optimizing pattern generation techniques to maximize the probability of inserted Trojans getting triggered and therefore detected by logic testing [8], [21].

On the contrary, *side channel analysis* methods focus on observing some physical parameters of the circuit (such as power consumption or timing) while elaborating some input test data. The main trend is to rely on golden ICs (i.e. circuit that have been ensured to be Trojan-free by destructive methods) to make comparison with the circuits under test. The introduction of additional logic gates should become visible because of an increase of the power consumption of the circuit or an increase of the delay in the logic path containing the trigger or the payload. More details will be given in the next section.

An idea introduced in [8] consists in using logic testing techniques in conjunction with side-channel analysis techniques. In fact, all techniques that help triggerring a Trojan help also magnifying the Trojan's impact on power signature.

Both logic testing methods and side-channel analysis methods aim at testing manufactured circuits. Another approach, called *Design for Hardware Trust*, consists in modifying the IC design flow in order to incorporate into the ICs some features that should improve Trojan detectability.

Chakraborty et al. presented two approaches, both consisting in modifying the state transition function of a circuit [6], [7]. The main idea of these modifications is to make the Trojan either more detectable or functionally benign.

Salmani et al. presented another approach in [18] consisting in inserting *dummy flip-flops* i.e. flip-flops that aim at removing rare triggering conditions in circuits.

III. Side channel analysis

A. Dynamic power

Agrawal et al. were the first to address the Trojan issue and proposed a side-channel analysis detection scheme based on transient power analysis [1]. They proposed an approach consisting in generating fingerprints of Trojan-free ICs and using these fingerprints to check whether the profile of an IC under authentication masks the fingerprints or not.

In [16], the authors analyze regional transient power supply signals and propose signal calibration techniques to reduce test environment variations.

Based on the assumption that the power analysis depends strongly on the effectiveness of pattern to magnify Trojan contribution to circuit power consumption, the method presented in [3] consists in applying a vector to a circuit, and keeping intact the inputs for several clock cycles in order to see activities converging to a specific portion of the circuit. Furthermore, a technique is presented which gives, for each gate of a circuit, the probability that it is connected to the Trojan.

A similar approach is presented in [9], in which a region-based vector generation method is propose that aims to induce maximum activity in one region and minimum activity in other regions.

Considering design for hardware trust, Salmani et al. presented in [17] a method consisting in reordering scan cells based on their geometric position. This can significantly restrict switching activity into a specific region and therefore help enhancing side-channel techniques.

B. Static power

In [2], a method is presented that uses static power to perform gate leakage estimation.

C. Delay

Jin and Makris proposed in [11] a new fingerprint generating method using path delay information. The basic idea is to measure the delay of each path of the circuit and to compare it with golden references.

In the field of design for hardware trust, Li et al. propose in [13] a detection framework based on self-authentication of each circuit. The idea is to put additional gates on the circuit to be able to compare on-chip delays, in order not to rely on a golden IC.

D. Power and delay

The only approach to our knowledge exploiting both power and delay is presented in [15]. A gate-level characterization is done and a detection is made of gates which have inconsistent characteristics compared to their original specified characteristics. Both functional and parametric Trojans are discussed.

TABLE I
OVERVIEW OF SIDE-CHANNEL ANALYSIS LITERATURE

Researchers	Side-channel	golden IC	Benchmarks	Trojans		Detection
				Models	Insertion	measurements
Agrawal et al. [1]	Dynamic power	Yes	RSA	Counter: 16-bit, Comparators: 3 & 8-bit	RTL level	Gate level
Jin et Makris [11]	Delay	Yes	DES	4-bit counter, 2-bit comparator	Gate level	Gate level
Rad et al. [16]	Dynamic power	Yes	ISCAS85	Comparator	Gate & Layout levels	Layout level
Potkonjak et al. [15]	Static power & Delay	Yes	ISCAS85	1 inverter	Gate level	Gate level
Banga et Hsiao [3]	Transient power	Yes	ISCAS	About 20 gates	Gate level	Gate level
Alkabani et al. [2]	Static power	Yes	MCNC91	1 & 3 two-inputs gates	Gate level	Gate level
Du et al. [9]	Power	Yes	32-bits ALU, FIR filter	*small*	Gate level	Gate level
Salmani et al. [17]	Power	Yes	ISCAS89	Comparators: 4 to 18 inputs	Layout level	Transistor level
Li et al. [13]	Delay	No	ISCAS89	Chain of inverters	Layout level	Layout level

IV. ARE THESE METHODS RELIABLE?

A. Experiments in literature

In this section, focus is made on the experiments done in literature to prove the uselessness of each method. Table I summarizes each above-mentioned side-analysis method (first column) with respect to several criteria. The second column shows on which side-channel the analysis is based. The third column tells whether a golden IC is needed or not. The fourth column specifies the used benchmarks. The fifth and sixth columns detail the Trojan model and the level of insertion. The last column tells at what level measurements were made.

In our opinion, two fundamental shortcomings are common to nearly all of these experiments:

- Non realistic Trojans were used. Among others, chains of inverters presented in [13] seem not convincing enough to demonstrate the usefulness of the method, especially as the delay of the described Trojan varies from 3% to 10% of the circuit delay. Such kind of structures are chosen to be adapted to the method and do not resemble at all what practical Trojans would look like (Interested readers can refer to [4], [5], [10] for realistic Trojans).
- In numerous approaches, Trojan were inserted at gate level, or even RTL level. Furthermore, timing and/or power analysis were made at gate level. To our knowledge, the only reference to post layout inserted Trojan and measurements is in [16]. Tools such as Synopsys PrimeTime and PrimePower give accurate calculations and are very useful given design closure in today's IC market. However, simple experiments can show that they are not accurate enough when dealing with the Trojan issue, especially Trojan put in untrusted foundry. To be as specific as possible, Trojan should be inserted at the layout level and experimentations should also be made a the layout level.

Much effort has been made to take into account environment variations and therefore Trojans getting smaller, which is the main weakness of these methods. However, not only the experiments made do not reflect what a Trojan may look like, but also, they have not been conducted fairly accurately.

B. Delay analysis

One limitation of delay analysis based techniques (often not mentioned in the published papers) is related to the fact that measuring such paths is very hard, especially for short paths. For instance, if the trigger is inserted between two flip flops, its delay would be of few ps. This delay is not measurable by using modern tools. Moreover, as we show afterwards, a smart introduction of the Trojans would make ineffective any detection method based on the delay measurement.

Referring back to the mechanism of trigger and payload presented in Figure 1, one can conclude that a Trojan will have very little impact from a delay point of view. Indeed, regardless of the size and delay of the Trojan, the original paths ($a1- > a2$, $b1- > b2$ and $s1- > s2$ in the figure) are almost unchanged:

- Trigger: the only difference in trigger paths relies in the connection of the Trojan logic. This has a delay impact since this connection adds a capacitance to the total capacitance of the path the Trojan is taping the signal from. However, besides adding the logic gates belonging

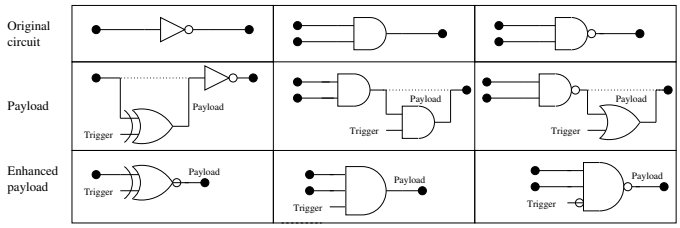
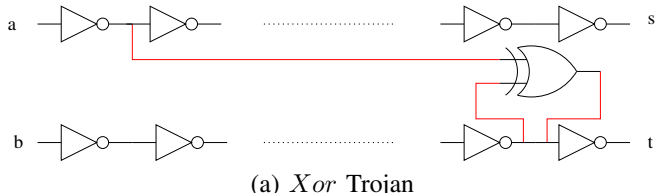
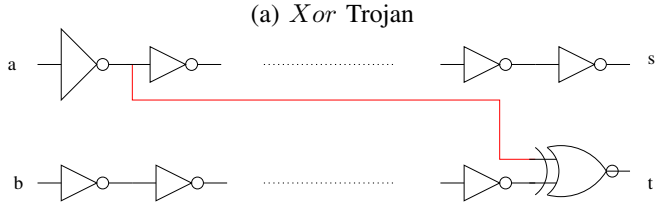


Fig. 2. Different kinds of payloads



(a) Xor Trojan



(b) Corresponding enhanced $Xnor$ Trojan

Fig. 3. Example: Xor and corresponding $Xnor$ Trojan

to the Trojans, it is reasonable to think that an attacker is also able to increase the drive strength of the door prior to the connection of the Trojan. This can balance the slight increase in delay.

- Payload: the only difference in the payload path typically relies in the adjunction of one gate (e.g. a XOR gate such as seen in literature) to deliver the payload under certain conditions. The path will therefore be affected by a minor delay (e.g. in a 65nm process, the intrinsic delay of a XOR gate is about a hundred ps). Furthermore, this payload mechanism is simplistic, and an attacker will be able to conceive more elaborate mechanisms to activate the payload, such as depicted in Figure 2. In this figure, for each of the 3 examples, we have shown a corresponding enhanced version. Such kinds of mechanisms could generate a delay up to ten times smaller than the adjunction of a XOR gate (e.g. transforming a NAND2 gate into a NAND3 gate leads to a increase of 5ps in the 65nm process).

A simple example consisting of two chains of inverters and an XOR gate as Trojan is presented in Figure 3.a.

Table II depicts the delay overheads for both paths according to several variants of the Trojan ($a- > s$ is the trigger path and $b- > t$ is the payload path). Xor refers to the Trojan of Figure 3.a, And and Or to the same type of Trojan, but with an And and an Or gate. $Xnor$, $Nand$ and Nor correspond to the enhanced versions with (1) an inverter with a bigger drive strength to limit the impact of the trigger, (2) a modification of the inverter at payload into a $Xnor$, a $Nand$ or a Nor gate to limit the impact of the payload (cf. $Xnor$ version in Figure 3.b).

TABLE II
SIMPLE TRIGGERS & PAYLOADS EFFECTS ON DELAY

Delay	Xor	Xnor	And	Nand	Or	Nor
Trigger	+5%	0%	+5%	0%	+5%	0%
Payload	+35%	+20%	+20%	0%	+25%	+5%

As expected, changing the power drive strength of the inverter totally masks the impact of the Trojan on the path to which the trigger is connected. Concerning the payload, one can easily point out the variety of results, with an example (the $Nand$ gate) for which the payload has been successfully hidden.

V. CONCLUSION

While side-channel analysis has been reported as an effective approach to detect Trojans, it seems that most approaches in literature lack at presenting satisfactory experimental results to prove the usefulness of the detection method.

Side-channel analyses are sensitive to environment variations, especially with increasing process variations in nanoscale technologies. Therefore, much effort has been made to address problems related to variability. However, too little effort has been made to assess the veracity of the simulation imprecisions.

Besides, delay based techniques do not seem realistic. Not only measuring numerous paths on a chip is very hard, especially for short paths, but also, simulation results show that an attacker can easily reduce the Trojan impact on delay to almost zero. Simple examples proposed in this paper have highlighted this issue.

In future work, larger examples will be studied. We plan to propose a realistic Trojan on a cryptographic circuit. To further investigate the simulation imprecisions issues, we will insert the Trojan at RTL level, gate level and layout level and analyse side-channels at gate-level and layout-level. We will also try to *hide* the Trojan from a delay point of view by applying simple mechanisms presented in this paper.

As regards design for hardware trust, it seems like a promising approach that needs to be developed. It indeed allows to create *testable* circuits i.e. circuits that contain ways to detect Trojan without the use of Goden ICs, such as presented in [13]. It allows also to create circuits that harden the insertion of a Trojan.

Furthermore, the effectiveness of side-channel-based techniques can be improved by adopting such techniques, which can add circuitry to support the measurement and analysis process.

REFERENCES

[1] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan Detection using IC Fingerprinting. In *IEEE Symposium on Security and Privacy (SP'07)*, pages 296–310, 2007.

[2] Y. Alkabani and F. Koushanfar. Consistency-based characterization for IC Trojan detection. In *International Conference on Computer-Aided Design (ICCAD'09)*, pages 123–127, New York, New York, USA, 2009. ACM Press.

[3] M. Banga and M. S. Hsiao. A Novel Sustained Vector Technique for the Detection of Hardware Trojans. In *22nd International Conference on VLSI Design (VLSI'09)*, pages 327–332. Ieee, Jan. 2009.

[4] A. Baumgarten, M. Steffen, M. Clausman, and J. Zambreno. A case study in hardware Trojan design and implementation. *International Journal of Information Security*, 10(1):1–14, Sept. 2011.

[5] G. T. Becker, A. Lakshminarasimhan, L. Lin, S. Srivathsa, V. B. Suresh, and W. Burleson. Implementing hardware Trojans: Experiences from a hardware Trojan challenge. In *IEEE 29th International Conference on Computer Design (ICCD'11)*, pages 301–304. Ieee, Oct. 2011.

[6] R. S. Chakraborty and S. Bhunia. Security against Hardware Trojan through a Novel Application of Design Obfuscation. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD'09)*, pages 113–116, 2009.

[7] R. S. Chakraborty, S. Paul, and Swar. On-Demand Transparency For Improving Hardware Trojan Detectability. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08)*, pages 48–50, 2008.

[8] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia. MERO: A Statistical Approach for Hardware Trojan Detection. In *International Conference on Cryptographic Hardware and Embedded Systems (CHES'09)*, pages 396–410. Springer, 2009.

[9] D. Du, S. Narasimhan, R. S. Chakraborty, and S. Bhunia. Self-referencing : A Scalable Side-Channel. In *International Conference on Cryptographic Hardware and Embedded Systems (CHES'10)*, pages 173–187, 2010.

[10] Y. Jin and N. Kupp. Experiences in hardware Trojan design and implementation. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '09)*, pages 50–57, 2009.

[11] Y. Jin and Y. Makris. Hardware Trojan Detection Using Path Delay Fingerprint. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08)*, pages 51–57, 2008.

[12] Y. Jin and Y. Makris. Proof Carrying-Based Information Flow Tracking for Data Secrecy Protection and Hardware Trust. In *IEEE VLSI Test Symposium (VTS'2012)*, pages 252–257, 2012.

[13] M. Li, A. Davoodi, and M. Tehranipoor. A Sensor-Assisted Self-Authentication Framework for Hardware Trojan Detection. In *Design, Automation and Test in Europe (DATE'12)*, 2012.

[14] L. Lin, M. Kasper, T. Güneysu, C. Paar, and W. Burleson. Trojan Side-Channels : Lightweight Hardware Trojans through Side-Channel Engineering. In *International Conference on Cryptographic Hardware and Embedded Systems (CHES'09)*, pages 382–395, 2009.

[15] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey. Hardware Trojan horse detection using gate-level characterization. In *46th ACM/IEEE Design Automation Conference (DAC'09)*, pages 688–693, New York, New York, USA, 2009. ACM Press.

[16] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic. Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD'08)*, pages 632–639, 2008.

[17] 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'10)*, pages 1–6, 2010.

[18] H. Salmani, M. Tehranipoor, and J. Plusquellic. A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20(1):112–125, 2012.

[19] M. Tehranipoor and F. Koushanfar. A Survey of Hardware Trojan Taxonomy and Detection. *IEEE Design & Test of Computer*, 27:10–25, 2010.

[20] X. Wang, M. Tehranipoor, and J. Plusquellic. Detecting malicious inclusions in secure hardware: Challenges and solutions. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08)*, pages 15–19. Ieee, June 2008.

[21] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty. Towards trojan-free trusted ICs: problem analysis and detection scheme. In *Design, Automation and Test in Europe (DATE'08)*, pages 1362–1365. IEEE, 2008.