Towards a Hardware Trojan Detection Methodology

Paris Kitsos
Informatics Engineering Department
Technological Educational Institute of Western Greece
and Industrial Systems Institute/RC "Athena"
pkitsos@ieee.org

Artemios G. Voyiatzis
Industrial Systems Institute/RC "Athena"
PSP building, Stadiou Str., Platani
GR-26504, Patras, Greece
bogart@isi.gr

***Abstract*— Malicious hardware is a realistic threat. It can be possible to insert the malicious functionality on a device as deep as in the hardware design flow, long before manufacturing the silicon product. Towards developing a hardware Trojan horse detection methodology, we analyze capabilities and limitations of existing techniques, framing a testing strategy for uncovering efficiently hardware Trojan horses in mass-produced integrated circuits.**

Keywords—security, hardware Trojans horses, integrated circuits, detection techniques, trusted hardware

I. INTRODUCTION

Computing devices are nowadays omnipresent, collecting viable information from the physical environment, performing precise computations at unparalleled speeds, and handling our digital environment while interacting with the physical one. The mass production, market penetration, and personalization capabilities of consumer-facing devices render them an attractive target for installing hidden functionality for the benefit of malicious third parties. A similar threat exists even for special-purpose and custom-built devices that are produced in low volumes but may be used in industrial settings, controlling critical infrastructures or key resources of a corporation or a nation.

It may be possible to insert the malicious functionality on a device not only as software but as deep as in the hardware design flow, long before manufacturing the actual hardware. A *hardware Trojan horse* is a modification of the original integrated circuit (IC) design by an intruder aiming to exploit hardware characteristics or hardware mechanisms in order to access and manipulate information stored or processed on the chip.

Hardware Trojan horses are acknowledged as a realistic threat and appropriate countermeasures must be designed for protecting consumer privacy, critical infrastructures, and key resources. Still, a methodological approach on testing an IC for existence of hardware Trojan horses is missing. Towards this direction, we survey proposed detection techniques for developing an optimized testing strategy.

The paper is organized as follows. In Section II, we describe the security issues in the IC design process. In Section III, we introduce the problem of efficient detection of hardware Trojan horses, while in Section IV we analyze the proposed detection methods. Section V discusses current and future research directions and concludes our paper.

This work was financially supported by GSRT Action "KRIPIS" with national and EU funds in the context of the research project "ISRTDI".

II. IC DESIGN PROCESS SECURITY

The IC design process comprises many steps. The first step of the process is the translation of the specification text into architecture and then the description in a hardware design language. Then, the synthesis step produces a gate-level netlist. The placing-and-routing step gives the netlist a physically-realizable form. The produced digital files are then handed to fabrication. Once the foundry produces the actual circuits, the testing step ensures their correct operation, and after the assembly-and-packaging step, the circuits are headed for deployment.

The design of more complex devices requires integration of software and hardware components sourced by different suppliers and vendors. This poses significant stress on the available time and resources devoted for quality assurance processes, including security testing, and increases the security concerns as designs and devices pass through deeper and deeper supply chains.

The choice of an appropriate attack surface heavily depends on available resources and determination of the attacker. In the early and more abstract steps of the design process, it may be easier to modify the (digital) designs. However, it is also more efficient to use automated tools for detection of modifications and more cost-efficient to correct a detected problem.

The fabrication step is considered as the critical point for introducing a hardware Trojan horse [1]. Once the fabrication of circuits starts, the removal of malicious functionality is not possible and the economic burden is enormous.

III. HARDWARE TROJAN HORSES

Most Trojan horses (hardware or software) are composed of two parts: the *trigger* and the *payload*. A Trojan horse has no spreading part, like a computer virus has. The trigger activates the malicious payload when specific conditions are met. The payload performs the malicious action(s) defined by its creator.

A hardware Trojan horse can be as simple as a single multiplexer with some buffers in the clock distribution network. During the nominal operation, the clock signal is derived through the network. When the hardware Trojan horse is activated, the clock signal passes through the buffer. The result is a performance downgrade or even the destruction of the IC's operation.

The detection of circuit modifications, as in the case of a hardware Trojan horse, is considered extremely difficult [2]. Nanometer-scale fabrication requires very precise and detailed physical inspection for detecting alterations. Traditional

techniques of testing for conformance with specifications may not be sufficient, as additional (malicious) functionality beyond specifications must be detected.

A methodological approach to address the question: "*Is a given Integrated circuit under Test (IUT) free of hardware Trojan horses?*" is still missing. Towards this direction, taxonomies have been developed that classify hardware Trojan horses based on their physical, activation, and functional characteristics [3-4].

There are two main parameters that define the testing strategy for an IUT. The first relates to the availability of a *golden model* i.e., an integrated circuit that is produced in a trusted fabrication plant. This circuit can be used as a reference model for detecting behavior and performance deviations of IUTs. However, the existence of a golden model is a topic of debate [2].

The second relates to the *reuse* of the IUT. If it is not necessary to reuse the IUT, then *destructive techniques* can be used. In this case, a demetallization process of the IUT extracts its layers, followed by image reconstruction and analysis aiming to detect modified transistors, gates, or routing elements. This is an extremely expensive and time-consuming approach. Furthermore, it cannot guarantee that the next chip in a batch is also free of a Trojan horse, unless that chip destructed too. However, it can be used for forensics as to provide the hard evidence of misconduct, especially in cases where the IUT was used in industrial or critical infrastructure environments.

In the next, we focus on *non-destructive techniques for mass-produced integrated circuits*. Ideally, a method should detect both the trigger and the payload part of a hardware Trojan horse. In the case of an *always-on* hardware Trojan horse, a trigger does not exist, as the hardware Trojan horse is always active and may, for example, secretly leak information to a third party [32].

A *condition-based* hardware Trojan horse includes a trigger that is used to activate the payload once some defined conditions are met. The conditions can be *external* to the circuit, such as commands sent through a hidden on-chip antenna or carefully crafted communication patterns. Examples of *internal* conditions include operational parameters, like temperature and power supply, and logic-based activation using as stimuli specific internal logic state, interrupts, instruction, or counter values.

Either always-on or condition-based, an integrated circuit infected with a hardware Trojan horse will exhibit at some time moments operational behavior variations, such as in specific area activity, in power consumption, in thermal emissions, and in output time delay. A detection technique should be able to identify and report these variations, ideally without false positives, in a time- and cost-efficient manner.

IV. Detection Methods

The detection of hardware Trojan horses is a topic receiving significant attention in the research and industry community and various techniques have been proposed in the literature. The non-destructive techniques can be classified into Run-time and Test-time Monitoring approaches.

The *run-time techniques* are typically invasive in nature, where some special circuits are involved. The run-time monitoring circuits can utilize pre-existing redundancy in the circuit as to detour its infected parts. A golden model is not required and all ICs can integrate the monitoring circuits. However, significant performance and power consumption overheads are incurred.

The *test-time techniques* can also be used by special circuits, similar to Design-for-Testability circuits [5], like scan-chains. These circuits can enhance the detection sensitivity or coverage substantially. Test-time techniques can be classified into approaches based on *circuit logic* and on *side-channel analysis*. The circuit logic approaches apply carefully-crafted test vectors for activating an infected circuit and observing the effects of its malicious payload at the primary outputs [2].

The presence of any malicious insertion in the IC is reflected into one or more side-channel parameters, such as quiescent supply current; leakage current; dynamic power trace; electromagnetic radiation (EM) due to switching activity; and path-delay characteristic [6,7,13,33]. Specialized and expensive testing equipment is necessary as to detect the weak side-channel signals produced by hardware Trojan horses on a case-by-case basis. Side-channel analysis does not need to activate the malicious payload in order to detect it.

A. Logic Analysis

"*Structural checking*" is a first attempt for an automated method to detect and prevent the integration of malicious logic into an IC at the design phase, before its fabrication [8]. A bi-directional linked-list structure of the entire ASIC is constructed. The list is searched forward from the inputs to detect internal malicious logic and backwards from the outputs to detect external data tampering or information leakage attacks using a trace-back approach. The tool implementing the method can generate a report of the location(s) and type(s) of suspicious logic and then prompt the designer for further investigation. If confirmed to be malicious, the tool can automatically remove the respective logic from the design.

A second method is the "*On-demand transparency*" [9]. A special operation mode (transparency) of the system is defined. In this mode, a signature is generated at the output based on a user-defined key at the system input. The insertion of a Trojan horse causes a different signature to be computed and thus, its presence is detected.

An extension of the previous method, "*key-based obfuscation*" is proposed in [10]. Obfuscation is an efficient technique that transforms a design into a functionally-equivalent but significantly harder to reverse-engineer one. The circuit can operate in two distinct modes (obfuscated and normal) with identical behavior. The transformation obfuscates the rareness of the internal circuit nodes thus, making difficult for an intruder to insert a hard-to-detect Trojan horse. Additionally, it may neutralize some inserted Trojan horses, allowing activation only in obfuscated mode and easy detection. The method introduces modest area and power overheads, while the required modification can be easily automated and integrated into the conventional design flow.

A method for detecting malicious circuits in FPGAs using Error Correcting Codes (ECCs) is presented in [11]. According to this method, the ECC-based Configuration Logic Block (CLB) calculates "parity groups" (PGs) and embeds the check CLBs for each parity group into the FPGA. During a trust-checking phase, a Test Pattern Generator (TPG) and an Output Response Analyzer (ORA) configured in the FPGA are used to check that each parity group of CLB outputs produces the expected parities. The vector produced by the ORA is then checked to determine if it is the expected parity vector for this PG. Experimental results on two medium-sized and a moderately-large circuit showed that it can detect tampers (i) in circuit CLBs-LUT functions, storage elements, and internal interconnections; (ii) in the checker circuit comprising the TPG and ORA; and (iii) insertions of extraneous logic. The method incurs a small hardware overhead and can achieve up to 100% tamper detection with as low as 0% false alarm probability.

B. Path Delay Analysis

A fingerprint-generating method using *path delay information* is proposed in [13]. The path delays of Trojan-free circuits (golden model) are collected in order to construct a series of fingerprints, each one representing one aspect of the total characteristics of a genuine design. Circuits are validated by comparing their delay parameters to the fingerprints. This method is very useful for detecting small-sized Trojan horses since it utilizes the delay paths in a circuit instead of an overall power fingerprint.

A similar in design philosophy method based on the *delay characterization* is presented in [14]. This method can be applied on a large number of internal combinatorial paths to get accurate and precise data about register-to-register path delays. Further, it can be applied on the functional paths of the core circuit without affecting timing and functionality. Actually, the delay measurements can be used to generate long and unique signatures for authentication purposes based on manufacturing variations. The measurements can be performed on functional parts of the circuit. In addition, these delay measurements can be used to detect design alterations and modifications at both test-time and run-time.

Finally, the *clock-sweeping technique* for measuring path delays without additional resources is presented in [15]. Transition Delay Fault and Path Delay Fault patterns are used to obtain high coverage on the nodes of critical and non-critical paths. The main idea is the production of delay signatures for all paths. During clock sweeping, a pattern at different clock frequencies is applied starting from lower speeds. Some paths, being sensitive to the pattern that is longer than the current clock period start to fail when the clock speed increases. The obtained start-to-fail clock frequency can indicate the delays of the paths. In order to detect a Trojan horse, the multidimensional scaling statistical method was used.

Path delay analysis may be a promising approach but cannot be applied to large or complex circuits for now. Covering all the paths of an IC and collecting accurate measurements of delay is very time-consuming and difficult with available technologies. Furthermore, simulation results indicate that attackers may easily reduce the Trojan horse's impact on delay near to zero [16].

C. Current Analysis

A statistical screening of test vectors that reduces false positives and false negatives in a hardware Trojan horse detection scheme based on *leakage current* side-channel analysis is described in [12]. A number of test chips embodying the same IUT are given. The procedure consists of three tasks. At first, conduct a single-chip test for all test chips and declare findings for each chip. So, for each test chip, many test vectors to the IUT on the chip as input vectors of the circuit are applied. For each test vector, the empirical power consumption and its ideal power consumption at their respective states driven by the test vector are measured. Then, two declarations (Positive and Negative) about the Trojan horse presence on the chip are defined and the probabilities of false positive and false negative declarations in the preceding task are analyzed. Finally, a statistical conclusion suggesting whether the IUT is Trojan-free is drawn. After completing all single-chip tests on many test chips, the likelihood of Trojan horse presence based on the outcome is determined.

The work in [17] detects hardware Trojan horses based on *current analysis*. The local transient current, I_{DDT}, on multiple power supply ports is measured in respect to a calibration technique in order to transform and normalize the current values. Ten different layouts of the standard IC were constructed. The first layout was the Trojan-free. The remaining nine layouts integrated different models of Trojan horses. After a statistical analysis, a scatterplot is created that is used to decide whether an IC is affected or not. The main phases of this method are two. During the first phase, the elliptical bound of the scatterplot is defined, while during the second phase the tests of the ICs are analyzed to detect Trojan horses. Simulation experiments were held to measure detection results. The method reliably detects Trojan horses that consist of four or more standard cell gates that can be distributed all over the IC. If the Trojan horse is smaller, the proposed technique is inefficient.

A detection method based on *non-intrusive external IC quiescent current measurements* (leakage current, I_{DDQ}) is proposed in [18]. The authors define a new metric called *consistency* and based on this metric and the properties of the objective function, they present a robust estimation method that estimates the gate properties while simultaneously detecting the Trojan horses. During the first step of the method, a generation of the input vectors to enable leakage current measurements is executed. Then, the measurement vectors are applied and map the measured values to gate leakages. Finally, the anomalies are detected by comparing with the Trojan-free nominal simulation values while considering noise/process variation sensitivity.

A similar method that is based on the analysis of a circuit's *steady-state currents* (I_{DDQ}s), which are measured simultaneously from multiple places across the 2-D surface of the IUT, is presented in [19]. The method also incorporates a technique for virtually eliminating process and test

environment variations effects, which act to reduce detection sensitivity of traditional testing approaches. The chips incorporate an array of cells that allow a Trojan horse to be emulated in many distinct locations on the IC. The design permits control over the position and the magnitude of the Trojan horse current as well as the magnitude and distribution characteristics of the overall leakage current. The proposed analysis demonstrates that the detection sensitivity is strongly correlated with (i) the magnitude of the Trojan horse current, (ii) the position on the power grid, from which the Trojan horse sinks current, and (iii) the pattern and variation in the chip-wide leakage current.

A design approach based on *current sensors* in an IC for verifying the security against the hardware Trojan horse attacks is proposed in [20]. This approach takes advantage of the dynamic supply current analysis for identifying malicious hardware in general. In the case of a realistic power-grid model, a good structure of integrating sensors can improve the Trojan horse detection sensitivity compared to external current measurements based on side-channel analysis. The sensor design takes advantage of the existing power-gating transistors in a design that are used as to achieve low-power operation. Supply-gating transistors are used to turn off parts of the circuit during idle states as to reduce their leakage current. A current monitor circuit utilizes these transistors as to measure supply current. Finally, a control circuit collects the data from multiple current sensors and sends them using the output pins of the IC.

A scalable detection and diagnosis approach that uses *segmentation and Gate Level Characterization* (GLC) is presented in [21]. The overall leakage current for a set of different input vectors is used. The core idea is to divide the large circuit into small sub-circuits by using input vector control, so that the segmented circuits have some desirable properties (e.g., small number of gates) for obtaining accurate Trojan horse detection results. After a segmentation process, test points are inserted to deal with possible large segments. Based on the segmentation approach, it is possible to obtain accurate GLC results and Trojan horse diagnosis on large circuits by tracing gate level leakage power.

In general, current analysis methods can detect Trojan horses occupying a small percentage of the overall circuit. The efficiency of each technique is heavily dependent on the quality of the test pattern generation procedure.

D. Power Analysis

A statistical approach for detecting Trojan horses based on the analysis of *power supply transient signals* is used in [22]. In principle, it is a power supply transient analysis (I_{DDT}) technique that is robust to the adverse effects of process and environment variations. Initially, I_{DDT} measurements from power ports on the IC are collected. For each pairing of power ports, a scatterplot is constructed using the areas produced from simulations of some Trojan-free and some other Trojan-inserted model. The process of calibration on each IC is taken using special calibration circuits connected through a scan chain and inserted directly below each of the power ports. The calibration circuits are designed to generate a simple stimulus to the power grid.

The *transient power analysis* method is also used [23]. The proposed technique increases the probability of generating a transition in infected circuits when operating and an analysis of the transition generation time is performed. The methods using transient power analysis require patterns that increase the malicious circuit activity whereas keep the overall circuit activity low as to magnify the Trojan horse contribution into the circuit power profile.

The transition probability is estimated based on the number of clock cycles needed to generate a transition on a net. Aiming to increase the transition probability of nets whose transition probability is lower than a specific probability threshold, an efficient dummy flip-flop insertion procedure is proposed in a way that does not affect the functionality of the circuit. This approach increases the probability of Trojan horse activation and of observing an erroneous response to the applied vectors.

Regional activation, in which transitions are limited to a target region of circuit while other regions are kept quiet, is an effective way to increase the ratio of Trojan-to-circuit power consumption [25]. Power consumption of a Trojan horse is expected to be negligible compared to the whole IUT, assuming that the Trojan horse is small compared to the overall circuit. Hence, to improve Trojan horse detection, the ratio of Trojan-to-circuit power consumption must be increased. Generating transitions in a target region and keeping other regions idle can increase Trojan-to-circuit power consumption ratio since it reduces the amount of circuit power consumption.

A novel *layout-aware scan-cell reordering* method to localize design switching into a specific region and improve detection is presented in [24]. The proposed method constructs scan chains by considering the physical information of scan cells. Random patterns are used to generate switching in the target regions and correlate with monitored power consumption

A promising run-time and low-overhead technique for hardware Trojan horse detection is *temperature tracking*. Apart from the correlation between power and temperature, a Trojan horse can cause a significant variant in the chip's power consumption. Thermal sensors required for detection are already embedded in many chips. A framework for temperature tracking consisting of design-time, test-time, and run-time phases is proposed in [26].

In the design phase, some statistical characteristics of switching activity, power consumption and thermal dynamics are collected and then the thermal sensors are placed. In the test-time phase, a calibration of chip due to the process variation takes place. Finally, during the run-time, the information from thermal sensors of the previous phases are collected and processed in order to detect Trojan horse activation.

E. Hybrid Analysis

The segmantation and Gate Level Characterization technique presented in [21] is exteneded to include thermal conditioning in [27]. This approach solves the collinear correlation problem and improves the detection efficiency under leakage power variations.

A novel scalable side-channel approach, named *self-referencing*, along with associated vector generation algorithm

to improve the Trojan horse detection sensitivity under *large process variations* in presented in [28]. It compares transient current signature of one region of an IC with that of another, thereby nullifying the effect of process noise by exploiting spatial correlation across regions in terms of process variations. To amplify the Trojan horse effect on supply current, a region-based vector generation approach is used, which divides the IUT into several regions. For each region, it finds the test vectors that induce maximum activity in that region, while minimizing the activity in other regions. These vectors should be able to trigger most of the feasible Trojan horses in a given region. Self-referencing between the measured supply current is performed when these test vectors are applied. The collected values are then compared with those of golden models at different process corners, as to detect the existence of a hardware Trojan horse in the tested regions.

F. Other Techniques

A novel randomized approach to *probabilistically compare the functionality* of the implemented circuit with the design of the circuit is presented in [29]. The proposed technique constructs a unique probabilistic signature of a circuit and a probability distribution on the inputs such that the probability distribution of the output is unique for every functionally-distinct circuit. The technique infers the presence of a Trojan horse in an IUT. The output of the technique is either an input pattern that distinguishes the functionality of the IUT from its design or a quantitative confidence level that the IUT is Trojan-free. The confidence level can be improved by running the analysis technique for a longer time.

The detection of Trojan horses in *deployed devices* is examined in [30]. It is a last layer of defense, in the case a hardware Trojan horse escapes detection and complements existing run-time techniques. There, the Secure Heartbeat and Dual-Encryption (SHADE) is proposed. The principal components of SHADE are two: its architecture and a specially instrumented compiler.

The SHADE architecture consists of two processing elements (guards) that adversarially check each other for correctness and responsiveness, essentially providing a secure execution environment using untrusted hardware. The trusted compiler is responsible for preparing the applications to run under the SHADE architecture. The compilation process is not executed by the system that is instrumented with SHADE. Instead, it is part of the board assembling.

In order for SHADE to be effective, a non-collusion assumption is made that two different foundries are used for the two guards so that information leakage occurs only if both collude. SHADE also assumes that the engineers and processes directly employed by the chip designer and the board developer are trusted entities, so that the board is assembled at a trusted location under the control of either the designer or the developer.

A medium-cost detection technique based on *image processing* is presented in [31]. It compares optical microscopic pictures of the silicon product (high-resolution images of the layers) with the golden models that are the original (pre-fabrication) images of the circuit layers represented in the Cadence Design Systems software. A cross-corellation of the images is shown to detect an implanted hardware Trojan horse in an AES-128 block cipher implementation. We note that depending on the process used, it may be necessary to remove some layers, effectively destroying the IUT.

V. Discussion and Conclusions

The analysis of hardware Trojan horse detection methods and techniques highlights a wealth of detection techniques focusing on different aspects of the operation of an integrated circuit, hardening the work of attackers on implanting a Trojan horse during fabrication and going unnoticed.

At the same time, a Trojan-free IC is even harder to verify, as the complexity of integrated circuits steadily increases. Each technique can only conclude that the given parameter of a circuit's operation (e.g., power consumption, current, or thermal emission) does not indicate the presence of a hardware Trojan horse with a high confidence level. Yet, it does not prove or ensure that the IUT is Trojan-free, even more that the whole batch or production are also Trojan-free.

The required infrastructure for testing and the increasing sophistication of the techniques indicate that it may not be possible to collect all resources under the same roof. Rather, a collaborative approach, such as the one pursued in the context of the TRUDEVICE network (http://www.trudevice.com/) may be preferable, where research teams and institutions with different skills and equipment combine forces as to achieve economies of scale and repetitiveness of the experiments and detection techniques using different equipment.

The time and space complexity of the available techniques remains prohibitively high in many cases for deployed circuits; advances towards reducing this complexity e.g., by smart and adaptable search, are more than welcomed.

Security testing after fabrication may indicate that the production can be trusted with high level of confidence. In some cases, this may not be enough for the customers, as they need to ensure that the delivered and deployed circuits are legitimate and originate from the specific fabrication process (e.g., an attacker may have mixed some legitimate and fake ICs after the production). For this, time- and cost-efficient techniques at customer premises or trusted labs are necessary.

Malicious modification of integrated circuits is an acknowledged security threat. The global and deep supply chains, the possibly untrusted fabrication facilities, and the increased sophistication of integrated circuits intensify the need for efficient and trusted hardware Trojan horse detection techniques. Given the complexity and the broad spectrum of possible attacks, we identified the lack of a capable detection methodology, explored the advantages of and areas of improvement for available detection techniques options, and framed an approach for defining a testing strategy aiming to cover as many cases as possible operational characteristics of circuits towards increasing the trust that are Trojan-free.

References

1. R. S. Chakraborty, S. Narasimhan and S. Bhunia, "Hardware Trojan: Threats and Emerging Solutions", Proceedings of the 2009 IEEE

International workshop on High Level Design Validation and Test (HLDVT 2009), San Francisco, CA, pp. 166-171, 2009.
2. M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection", IEEE Design and Test of Computers, January/February 2010, vol. 27 no. 1.
3. R. Karri, J. Rajendran, K. Rosenfeld and M. Tehranipoor, "Trustworthy Hardware: Identifying and Classifying Hardware Trojans", IEEE Computer, October 2010, vol. 43, no. 10.
4. X. Wang, M. Tehranipoor and J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions", Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, Washington, DC, USA, pp. 15-19, 2008
5. L. – T. Wang, C. –W. Wu, X. Wen, X. Wen, "VLSI Test Principles And Architectures: Design for Testability", Academic Press, 2006, ISBN 0123705975.
6. A. Das, C. E. Veni Madhavan, "Public-key Cryptography: Theory and Practice", Pearson Education India, 2009, ISBN 81-3170-832-2.
7. S. Mangard, E. Oswald, T. Popp, "Power Analysis Attacks: Revealing the secrets of smartcards", Springer, 2007, ISBN 978-0-387-30857-9.
8. S.C. Smith, J. Di, "Detecting Malicious Logic Through Structural Checking", 2007 IEEE Region 5 Technical Conference, 20-22 April 2007, Fayetteville, AR, USA, pp. 217-222, 2007
9. R. S. Chakraborty, S. Paul, S. Bhunia, "On-demand transparency for improving hardware Trojan detectability", IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08), pp. 48-50, 2008.
10. R. S. Chakraborty, S. Bhunia, "Security against Hardware Trojan through a Novel Application of Design Obfuscation", in IEEE International Conference on Computer Aided Design 2009 (ICCAD 2009), pp. 113-116, November 2-5, 2009, San Jose, California, USA, 2009.
11. S. Dutt, L. Li, "Trust-Based Design and Check of FPGA Circuits Using Two-Level Randomized ECC Structures", ACM Transactions on Reconfigurable Technology and Systems (TRETS), Volume 2 Issue 1, March 2009.
12. Y. Gwon, H. T. Kung, D. Vlah, K.-Y. Huang, Y.-M. Tsai, "Statistical Screening for IC Trojan Detection", In 2012 IEEE International Symposium on Circuits and Systems (ISCAS 2012), Seoul, Korea, pp. 85-88, 20-23 May 2012.
13. 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.
14. J. Li, J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection", in IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08), pages 8-14, 2008.
15. K. Xiao, X. Zhang, and M. Tehranipoor, "A Clock Sweeping Technique for Detecting Hardware Trojans Impacting Circuits Delay", IEEE Design & Test, Vol. 30, Issue. 2, pp: 26-34, 2013.
16. S. Dupuis, G. Di Natale, B. Rouzeyre, "Is Side-Channel Analysis really reliable for detecting Hardware Trojans?",XVII Conference on Design of Circuits and Integrated Systems (DCIS'2012), Avignon, France, 2012.
17. R. M. Rad, J. Plusquellic, M. Tehranipoor, "Sensitivity Analysis to Hardware Trojans using Power Supply Transient Signals", in IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'08), pp. 3-7, 2008.
18. Y. Alkabani, F. Koushanfar, "Consistency-based Characterization for IC Trojan detection", Proceedings of the 2009 International Conference on Computer-Aided Design (ICCAD '09), pp. 123-127, 2009.
19. J. Aarestad, D. Acharyya, R. Rad, J. Plusquellic, "Detecting Trojans Though Leakage Current Analysis Using Multiple Supply Pad IDDQS", IEEE Transactions on Information Forensics and Security, Volume 5 Issue 4, pp. 893-904, December 2010.
20. S. Narasimhan, W. Yueh, X. Wang, S. Mukhopadhyay, S. Hunia, "Improving IC Security against Trojan Attacks through Integration of Security Monitors", IEEE Design & Test of Computers, 2012.
21. S. Wei, "Scalable Hardware Trojan Diagnosis", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol: 20 , Issue: 6, pp. 1049-1057, 2012.
22. R. M. Rad, X. Wang, M. Tehranipoor, J. Plusquellic, "Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans", 2008 International Conference on Computer-Aided Design (ICCAD 2008), San Jose, CA, November 10-13, pp. 632-639, 2008.
23. H. Salmani, M. Tehranipoor, J. Plusquellic, "New Design Strategy for Improving Hardware Trojan Detection and Reducing Trojan Activation Time", in IEEE International Workshop on Hardware-Oriented Security and Trust (HOST'09), pages 66-73, 2009.
24. H. Salmani, M. Tehranipoor, J. Plusquellic, "Layout-Aware Scan-Cell Reordering for Improving Localized Switching to Detect Hardware Trojans", IEEE International Workshop on Information Forensics and Security (WIFS 2010), pp: 1-6, 2010.
25. M. Banga and M. Hsiao, "A region based approach for the identification of hardware Trojans", IEEE Intl. Workshop on Hardware-Oriented Security and Trust (HOST'08), pp. 40-47, 2008.
26. D. Forte, C. Bao and A. Srivastava, "Temperature Tracking: An Innovative Run-Time Approach for Hardware Trojan Detection", IEEE/ACM International Conference on Computer-Aided Design (ICCAD), November 2013.
27. S. Wei, M. Potkonjak, "Scalable Segmentation-based Malicious Circuitry Detection and Diagnosis", 2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2010), pp. 483-486, 2010.
28. D. Du, S. Narasimhan, R. S. Chakraborty, S. Bhunia, "Self-Referencing: A Scalable Side-Channel Approach for Hardware Trojan Detection", 12th international conference on Cryptographic hardware and embedded systems (CHES 2010), pp. 173-187, 2010.
29. S. Jha and S. K. Jha, "Randomization Based Probabilistic Approach to Detect Trojan Circuits," IEEE High Assurance Systems Engineering Symposium (HASE08), pp. 117-124, 2008.
30. G. Bloom, B. Narahari, R. Simha, J. Zambreno, "Providing secure execution environments with a last line of defense against Trojan circuit attacks", Computers & Security, Volume 28, Issue 7, October 2009, pp. 660-669.
31. S. Bhasin, Jean-Luc Danger, S. Guilley, X. T. Ngo and Laurent Sauvge, "Hardware Trojan Horses in Cryptographic IP Cores", Tenth Workshop on Fault Diagnosis and Tolerance in Cryptography, August 20, 2013, Santa Barbara, CA, USA.
32. Y. Jin and Y. Makris. Hardware Trojans in wireless cryptographic ICs. IEEE Design Test of Computers, 27(1):26-35, 2010.
33. I. Exurville, J. Fournier, J.-M. Dutertre, B. Robisson, A. Tria, "Practical measurements of data path delays for IP authentication & integrity verification", 8th International Workshop on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC 2013), pp. 1-6, 2013.