Journal of Cryptographic Engineering (2022) 12:495–522
https://doi.org/10.1007/s13389-022-00295-w

REGULAR PAPER

A comprehensive survey of physical and logic testing techniques for Hardware Trojan detection and prevention

Rijoy Mukherjee[1] · Sree Ranjani Rajendran[2] · Rajat Subhra Chakraborty[1]

Received: 28 June 2021 / Accepted: 19 June 2022 / Published online: 16 July 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
Hardware Trojans have emerged as a great threat to the trustability of modern electronic systems. A deployed electronic system with one or more undetected Hardware Trojan-infected components can cause grave harm, ranging from personal information loss to destruction of national infrastructure. The inherently surreptitious nature and bewildering variety of Hardware Trojans makes their detection an extremely challenging exercise. In this paper, we explore the state of the art of post-silicon testing techniques for Hardware Trojan detection, with our coverage including both physical measurement-based testing, as well as logic testing. We present systematic classification of Hardware Trojans and a taxonomy of detection techniques based on physical and logical testing, and describe these techniques in details, including their stand-out features and strengths and weaknesses. We conclude the paper with an evaluation of the current status of progress, and major directions of future research.

Keywords Hardware Trojans · Logic testing · Malicious design modifications · Side-channel testing

1 Introduction

Electronic hardware constitutes the bedrock of any computing system, on which firmware layer, virtual machine layer (optional), operating system and other system software, and finally application software (which operate on the information being processed) are based. Thus, secure and trusted computing requires the deployment of effective security at different layers, including hardware security, software security and finally data/information security. Although computer system security issues span all three layers, hardware security plays a crucial role, as the robustness of the security of the software layers above the hardware layer implicitly depends on the assumption that the underlying hardware is completely trustable. If that is not the case, then an untrustworthy hardware platform opens new avenues of direct modification of the software/data that depends on it for execution, or by incorrect execution of the software, thereby modifying its functionality. Of even greater concern is hardware that surreptitiously performs operations that are not part of its standard specifications, e.g., secretly leaking sensitive information through a covert channel. These threats are exacerbated by the fact that there is a growing tendency of connecting electronic systems ("smart devices") to communication networks, and the proliferation of the "Internet of Things".

The impact of deployed untrustworthy hardware can be devastating, from a personal scale (e.g., through the leakage of account passwords) to national scale (e.g., a large-scale power outage caused by the national power grid malfunctioning), potentially resulting in human fatalities and financial disaster worth billions. Even when the hardware is not malicious or buggy by design, their complex operation can always be exploited by a resourceful adversary, the impact of hardware threats creates many software and network security issues, e.g., the infamous `Spectre` and `Meltdown` bugs [26] in certain families of modern microprocessors. Hence, secure and trustworthy hardware is indispensable in ensuring secure computing, and hardware can be thought to constitute

✉ Rijoy Mukherjee
rijoy.mukherjee@iitkgp.ac.in

Sree Ranjani Rajendran
rajendrans@ufl.edu

Rajat Subhra Chakraborty
rschakraborty@cse.iitkgp.ac.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India

[2] Department of Electrical and Computer Engineering, FICS Lab, University of Florida, Gainesville, USA

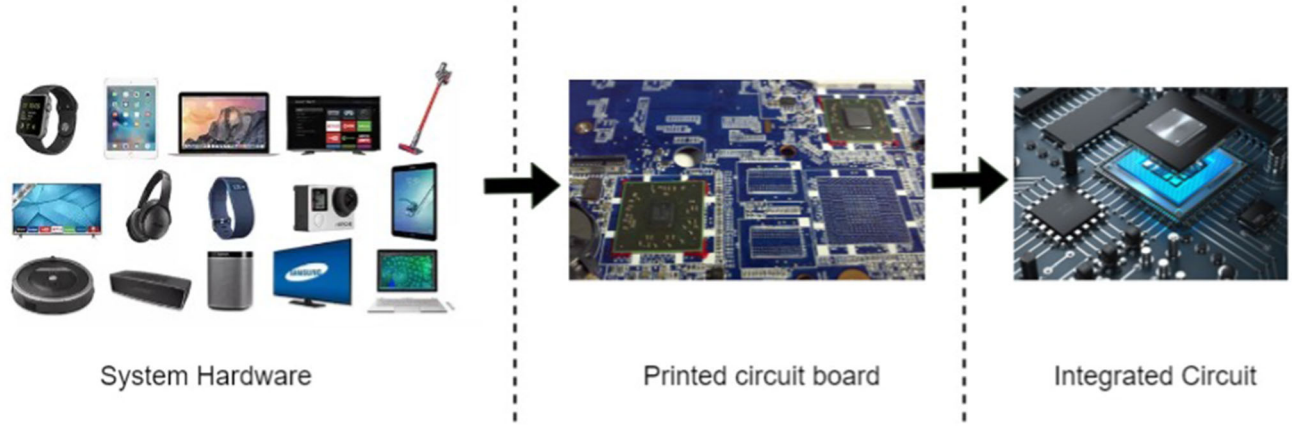

Fig. 1 A hardware component-based hierarchy of electronic systems, exemplified for common consumer electronic gadgets

an (ideally) inviolable "root-of-trust" of any modern electronic system.

Figure 1 shows a typical hardware component-based hierarchy of the modern electronic systems, exemplified through common consumer electronic gadgets. In this, *system hardware* (end device) is at the top level, which is composed of *physical components* like printed circuit board (PCB), peripheral devices and enclosures. The physical components, in turn, are composed of ICs, often totalling in the hundreds in a single electronic system. Hence, it is easy to understand the complexity of the task of verifying the functionality of a system consisting of so many inter-operating complex ICs, given that the task of completely verifying the functionality of even a single standalone IC is often beyond the current state of the art. For the sake of argument, even if we assume that the question of whether an IC exactly implements its stated functional specification is answered satisfactorily, it is well-nigh impossible to answer the question of whether the IC has no *additional surreptitious functionality*.

The growing concern with the trustworthiness of electronic systems is also because of pervasive trends in design and manufacturing of electronic devices and components, particularly semiconductor integrated circuits (ICs). Most major companies that manufacture ICs commercially have design centers distributed all over the world, for the associated cost and talent acquisition advantages. To handle time-to-market pressure and to reduce costs, these companies routinely deploy Electronic Design Automation (EDA) software, standard cell libraries, and third-party hardware intellectual property (3-PIP) modules procured from third-party vendors, Additionally, to avoid the large capital expenditure in setting up, maintaining and upgrading cutting-edge IC fabrication facilities ("fabs"), most IC design companies have transitioned to a *fabless* business model [50,54], whereby they outsource the IC manufacturing, packaging, and testing, to (often) offshore companies specializing in operating fabs. Complex involvement of so many external parties distributed all over the world result in the loss of control of the IC design houses, and also to an extent the companies selling their hardware IP products, over the design and manufacturing process of their own products. As a result, a plethora of vulnerabilities plague ICs and related products, the most well-known among which are: IP piracy, IC piracy (cloning and overproduction) and counterfeiting, reverse-engineering, etc.

However, in this paper, we focus on detection and mitigation techniques for arguably the most challenging malady among these threats—*Hardware Trojans* (HTs) [22,47,69,94,107,121]. Hardware Trojans, as the name suggests, are malicious, hard-to-detect circuit modifications that have the potential of causing ICs to malfunction. These circuit modifications can be performed at an unreliable fab by reverse-engineering the design database for an IC, or can be part of the original design itself (either introduced in-house or through third-party IPs). HTs are surreptitious in nature, and a well-designed HT has the capability of evading traditional pre-silicon design verification (if inserted before the IC is taped out), and post-silicon IC testing. If seemingly benign but actually HT-infected ICs are deployed in-field, they are capable of various malicious operations, viz. incorrect output, performance degradation, denial-of-service, leakage of secret information (e.g., cryptographic keys). In addition, often HTs are intentionally designed to activate at seemingly random points of time, making it even more difficult to pinpoint the exact component and the exact input stimulus which caused the system malfunction.

Since HTs are capable of evading traditional post-manufacturing IC testing schemes, special testing techniques focusing on HTs must be developed, evaluated and adopted for their detection. Over the last decade-and-half, many such HT-centric detection and verification techniques have been proposed, for both ICs and constituent circuit modules. This

effort has been summarized by several published works [23,69,107]. However, with the fast-evolving HT detection research domain due to increasing adoption of advanced machine learning techniques, it becomes necessary to provide a detailed study of current state-of-art techniques.

In this paper, we provide a comprehensive review of the state of the art of HT detection techniques based on logic testing, as well as those based on measurement and analysis of physical parameters (circuit delay, power dissipation, electromagnetic radiation signature, optical signature, image signature etc.) We describe the relative strength of these techniques, as well as point to their important shortcomings, necessitating further research progress. In our coverage, we also include approaches that combine these two main approaches (logic testing and physical characterization), to increase the sensitivity of HT detection. By limiting the scope of our coverage to logic testing and physical parameter-based detection mechanisms, we provide a unique and comprehensive study of this sub-domain.

The rest of the paper is organized as follows. In Sect. 2, we present the necessary background on HTs and their detection techniques, including a taxonomy of HTs. In Sect. 3, we describe techniques for HT detection based on measurement of physical parameters, including relatively recent techniques that avoid the need of a "golden reference." In Sect. 4, we concentrate on logic testing techniques for HT detection. In Sect. 5, we discuss design techniques which enhance circuit testability for HT detection. Section 7 discusses in depth the problem of requirement of golden IC for testing. Section 8 discusses role of machine learning in the domain of physical and logical testing of HTs and points to future directions of research on this topic. Section 9 discusses expected future research directions. Section 10 concludes the paper. In addition, in Appendix A, we have presented a table which summarizes all the works discussed in this paper. In the table, the works have been chronologically arranged, allowing the reader to easily develop an idea of how research interest has evolved on this topic over the years, and to understand the direction in which current research on this topic is aligned to.

2 Preliminaries: Hardware Trojans and their detection

We first present a description of the general structure and some examples of HTs, a taxonomy of HTs, followed by a general overview of HT detection strategy.

2.1 Hardware Trojans: general structure and taxonomy

A Hardware Trojan is a stealthy circuit which usually remain inactive, allowing normal operation of the IC in which it

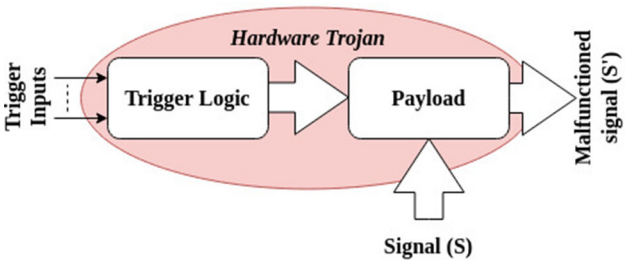

Fig. 2 General structure of a Hardware Trojan [22]

is embedded, until triggered by an internal "rare" logic condition, or on the application of one or multiple fixed input vectors (possibly in a fixed sequence) to the circuit [22,28,60,94,107]. "Stealthiness" can be described a property which can make undesirable malicious changes to a system inconspicuous—that is, conceal any changes made by any harmful actor to the infected system. Once activated or "triggered," a HT may or may not be able to cause a malfunction of the IC, and in case it does cause a malfunction, the malfunction might not be immediate, but only occur after a random time interval of further operation. This uncertainty of observable malfunction actually occurring even when an HT is activated is one of the features of HTs that help to increase their stealthiness. In some cases, the HT might even be *free-running or "always on"*, i.e., not dependent on any external applied stimulus, but autonomously operates and initiates a malfunction only after a fixed time interval of operation, e.g., when a counter circuit (which is part of the HT) reaches its terminal count. In the last case, to evade post-manufacturing testing, the HT designer ensures that the HT triggers after a long enough interval, so that it remains undetected during ordinary post-manufacturing testing.

One of the earliest proposed and subsequently widely studied structures of a generic HT is shown in Fig. 2. A HT consists of two main parts: *trigger logic* and a *payload* [118]. The trigger logic is responsible for continuously monitoring various signals or a series of events at the primary inputs or internal nodes of a circuit, and then generates one or more activation signals. These activation signals are utilized by the payload logic to cause malfunction inside the circuit, usually by altering internal signal values of original circuit (malicious behavior) once the trigger is enabled. The impact of the payload can be more sophisticated, including leakage of a secret information through an information backdoor.

Figure 4 shows two relatively simple examples of HTs, a *combinationally triggered* HT (Fig. 4a) and a *sequentially triggered* HT (Fig. 4b). The combinationally triggered HT consists of only combinational logic gates in its trigger mechanism. The adversary, who is assumed to have access to the original netlist or reverse-engineered netlist of the circuit in which the HT is to be embedded, has determined through analysis of the signal probabilities of the internal nodes, that

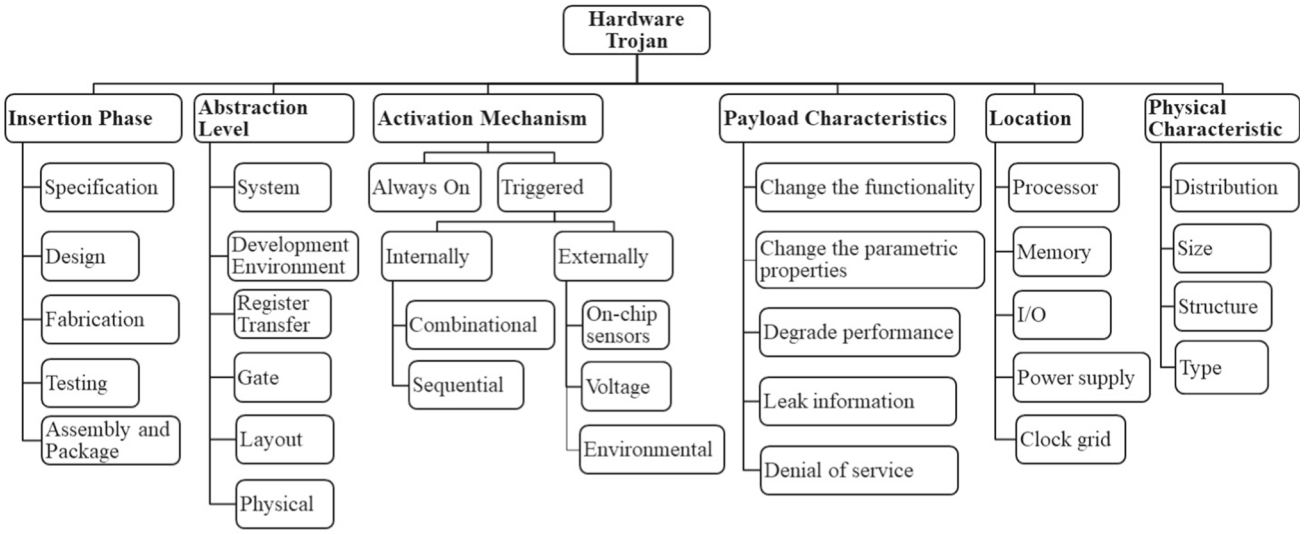

Fig. 3 Hardware Trojan taxonomy (adapted from [108])

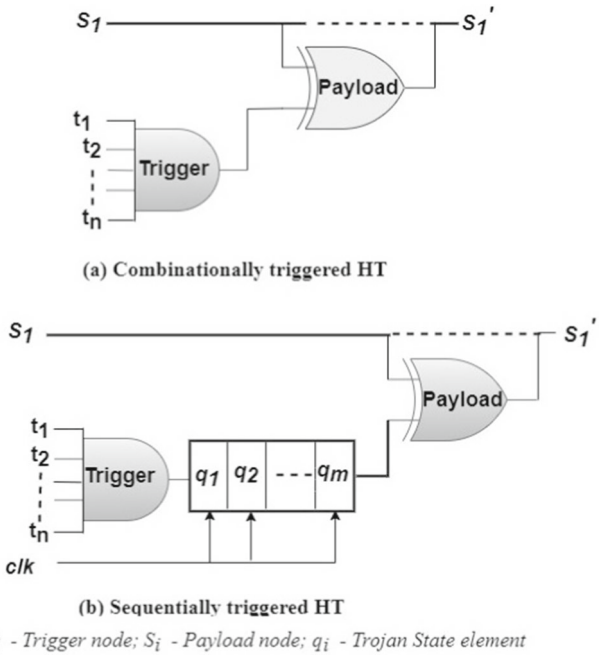

Fig. 4 Examples of combinationally and sequentially triggered Hardware Trojans [30]

t_1 through t_n are *rare nodes* with extremely low probability of them going to logic-1 simultaneously during circuit operations. However, on rare occasions that these nodes do achieve logic-1 simultaneously, they alter (flip) the logic value at internal node S_1. This complemented value may lead to an incorrect output for the infected circuit, if its effect propagates to the primary output. On the other hand, HTs can include state elements (flip-flops) as part of a Finite State Machine (FSM) in its trigger mechanism, as shown in Fig. 4b. Again, on the availability of a set of rare logic conditions at the nodes t_1 through t_n, the FSM performs state transitions on the clock edges, and when the FSM reaches a terminal state, again a logic malfunction is initiated.

Besides the two simple ways of categorizing HTs as exemplified above, over the years, a bewildering variety of HTs have been proposed, and HT design is itself an extremely active area of research. There is no universal consensus about classifying HTs; however, Fig. 3 shows a taxonomy of HTs that covers the most common methods of classifying HTs, based on diverse sets of characteristics, such as: the control exercised by the adversary on HT implementation; the activation mechanism of the inserted HT; the effect on the HT payload; the position of the inserted HT, as well as the physical characteristics such as size [100,102].

The HT taxonomy described in Fig. 3 also classifies HTs according to the different *phases of Trojan insertion*, ranging from HT insertion at the specification phase to the fabrication assembly and packaging phase. If HTs are classified according to their *abstraction level*, the control of the adversary on Trojan implementation can vary from system-level specifications to the actual physical implementation stage in the fab. HTs can be further classified based on the *activation mechanism*, ranging from always on (an activated HT as soon as the IC containing it is powered-on), to activate only when a specific trigger condition (either internal or external). For external triggering, environmental conditions like temperature, external applied stimulus like electromagnetic signals, etc., can be used to trigger HTs. In [49], the authors introduced a classification of *deterministic HT* (H_D) by discovering a crucial set of properties of trigger-activated Trojans. These properties, determining the stealthiness of Trojans, lead to a much more detailed classification of such Trojans and hence assign well-defined boundaries to the

scope of the existing and new countermeasures on the huge landscape of HTs. With the discovered properties, the adversary can design a tremendous number of HT and easy to provide a benchmark of with HT to the community.

Based on their *payload characteristics*, HTs can be categorized to be those changing functionality, or modifying parametric properties, degrade performance, leak secret information, or denial-of-service causing HTs. From another viewpoint, HTs are classified based on their *location*—HTs can be inserted in processor, memory, I/O ports, power supply or at clock grids. Finally, based on the *physical characteristics* like distribution, size, type and structure, HTs can also be classified. Note that classification of HTs into combinationally triggered and sequentially triggered types is also part of the taxonomy tree, under:

Activation Mechanism → Trigerred → Internally

We would not further elaborate on describing structures and operating modes of HT variants, because it is not the main focus of this paper—the interested reader is recommended to refer to [108] for more information on this topic.

2.2 Hardware Trojan detection techniques

Research on detection of HTs has progressed in lockstep with research on their design and implementation. As in any other topic of security-related research, HT designers and those working on techniques to detect them are involved in a cat-and-mouse game, with no clear winner determinable to-date. Since debugging an already deployed electronic system is extremely challenging, it is desirable that HTs are detected as early as possible in the product life cycle, preferably before a constituent IC has been taped out, or at least before the manufactured IC has become part of an electronic system. However, given the wide variations in HT type, structure and mechanism of action, it is unlikely that a "silver bullet" detection technique capable of detecting *every* type of HT would be ever developed [107]. On the other hand, detection techniques may vary based on the resources required and the deployment phase. Some techniques may require the netlist of the design or the layout description, while others may require at least one HT-free instance of the design (a "golden chip"). However, obtaining a trustworthy golden chip (or its accurate simulation model) is not always feasible in practice.

HT detection techniques can be broadly categorized based on the IC life cycle: *pre-silicon* and *post-silicon*. Pre-silicon detection techniques are carried out before IC fabrication and includes code coverage analysis, logic testing, formal verification, structural analysis and functional analysis. Based on the type of intervention to the circuit-under-test (CUT), *post-silicon* HT detection techniques can be classified as shown in the taxonomy of Fig. 5.

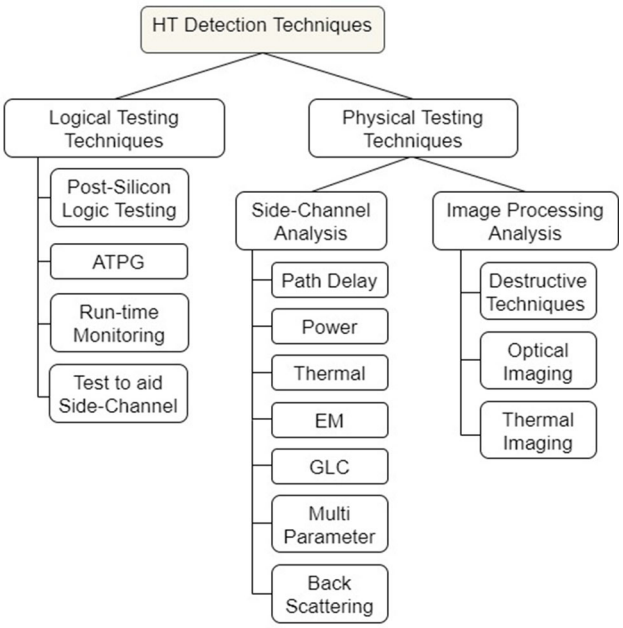

Fig. 5 Taxonomy of *post-silicon* HT detection techniques

Physical testing techniques consists of both destructive and non-destructive techniques. For example, optical inspection of IC, a destructive method of HT detection, requires de-packaging followed by active removal of successive layers of the chip, and optical analysis of the microphotographs of the metal layers. A large number of test-time techniques can be classified to follow the general theme of *Side-Channel Analysis* (SCA), which compares physical characteristics like dynamic power, static power, temperature, EM radiation, path delay of the IC under test, etc., against a reference circuit or its accurate simulation model. Also, some HT detection techniques have been developed to detect HTs at run-time, where a HT is detected in the operation phase by measurement of physical characteristics like SCA. In logical testing, specific test patterns are applied to an IC to detect anomalous activity triggered by the inserted HT either during test-time or run-time. As we will find in the next two sections, HT detection techniques are also equally diverse, incorporating a wide variety of ideas and strategies, where different techniques are applied for different sub-classes of HTs. Sometimes, testing strategies are combined to come with new schemes with improved detection coverage for a wider class of HTs.

3 Physical parameter measurement-based Hardware Trojan detection

Based on the taxonomy presented in Sect. 2.2, we now provide a detailed overview on the recent physical parameter-

based HT detection techniques. Broadly, we discuss two main classes of techniques: side-channel analysis-based Trojan detection and image processing-based Trojan detection.

3.1 Side-channel analysis-based Trojan detection

Side-channel Analysis (SCA), which depends on the collection and characterization of measurable physical parameters during circuit operation, has been an extremely successful technique to attack cryptographic implementations. However, similar techniques have also been widely adapted for post-silicon detection of HTs and are considered a powerful tool for the purpose. The main insight is that any malicious addition or alteration of circuit elements during IC design or fabrication is expected to have certain impacts on power consumption, delay of the circuit paths, the electromagnetic (EM) dissipation, or thermal characteristics of the interconnects and gates in the infected circuit. The presence of a HT in an IC can be suspected by accurately recording these parameters from an operational IC, and then comparing them with their expected value in a Trojan-free IC. Figure 6 shows the general flow involved in SCA-based HT detection schemes.

The main attractiveness of such techniques for HT detection is that they are non-disruptive, and do not affect the functioning of the IC. However, they also face major challenges. Accuracy of SCA-based detection methods suffer due to presence of noise resulting from process, measurement and environmental variations during the chip fabrication. Faithful measurement of physical parameters becomes exceedingly difficult and expensive for modern multi-Gigahertz ICs, while process variation induced noise exacerbates owing to the ever-shrinking size of transistors in the ICs. Due to process variations, side-channel signatures measured for two ICs of similar make and model, even when applied the same input pattern, might be different. Such factors can easily mask the small variations induced by an inserted small HT which is rarely triggered. Conversely, if any variation is observed, it becomes difficult to conclusively assign the observed variation to the presence of HT in the IC. Hence, the variability in silicon fabrication process as well as side-channel parameter measurement needs to be isolated for accurate detection of Trojans.

Another major shortcoming of physical parameter characterization-based HT detection techniques is that they rely on the existence of an accurate "golden model." This golden model is either obtained directly from a known HT-free instance of the IC ("golden IC"), or an extremely accurate simulation model of a golden IC. As mentioned previously, obtaining either of them in practice is often difficult. One possibility of obtaining a golden IC model is characterizing an instance of the IC for side-channel parameters, and then completely reverse-engineering it to a netlist-level description. If the reverse-engineered netlist matches a golden netlist available with the IC design house, then the IC instance can be concluded to be a golden IC [21]. However, the process of IC reverse-engineering is an expensive and time-consuming procedure at the current state of the art. Besides, absence of HT in one instance of an IC does not guarantee that all instances of the IC are HT-free. Another approach of obtaining a golden IC is to fabricate a small number of the ICs in a "trusted" foundry [21], and these chips can be considered as golden ICs. But then again, the existence of a foundry that can be certified to be "trusted" is a difficult proposition. Similarly, constructing an accurate device-level circuit simulation model for an entire IC (including all packaging and pin-related parasitics) is also quite challenging. But due to process variations, even after the availability of a golden IC or its model, side-channel analysis-based HT detection remains challenging.

In the rest of this section, we discuss various side-channel analysis-based HT detection techniques, for diverse physical parameters. We also present techniques proposed over the

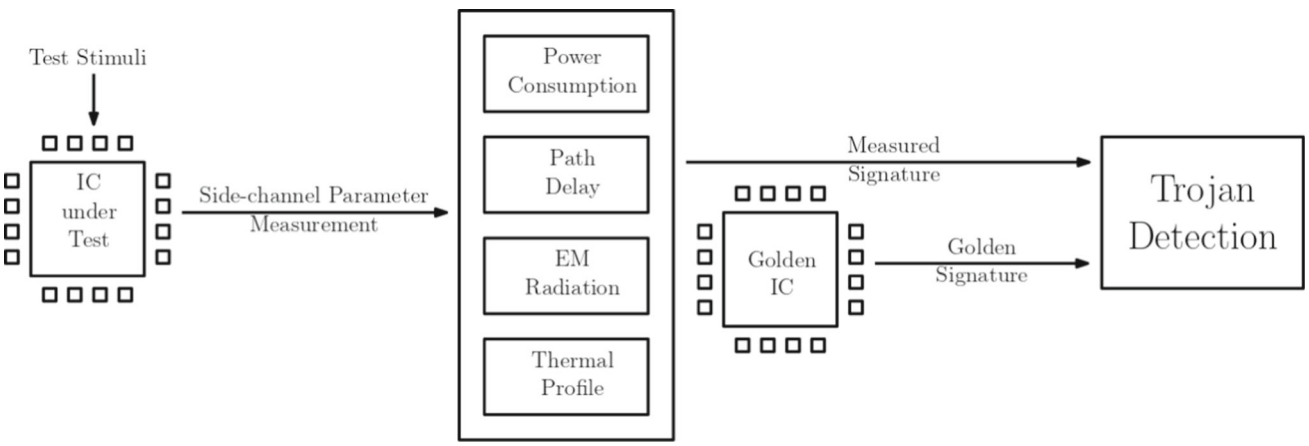

Fig. 6 Side-channel analysis-based Trojan detection workflow

years to overcome or bypass the challenges of SCA-based HT detection.

3.1.1 Power consumption characterization-based side-channel analysis

Agrawal et al. were the first to propose side-channel information for Trojan detection, where they used IC power consumption [4]. The golden IC's power signature was obtained by applying random test patterns, and power measurement. After obtaining the golden signature, the same test patterns are applied to the IC under test. *Principle Component Analysis* (PCA) was used to compare side-channel power fingerprint of the IC under test with golden model for de-noising. A shortcoming of this work was that it was primarily based on circuit simulation models, and was not very effective in detecting small HTs when process variation effects were taken into account. Wang et al. in [112] stated that HT detection sensitivity in the presence of unavoidable process variation can be significantly enhanced by measuring currents locally and from multiple spatially distributed power ports or pads. They developed a "multi-supply transient-current integration" methodology to detect a HT. The localized-current is measured from various power ports or controlled collapse chip connections on the die. Comparing the results obtained for golden chips against the IC under test, the presence of HT can be inferred with relatively high accuracy if the current integration results are very different from the golden IC results.

Aarestad et al. proposed a static current measurement-based side-channel approach in [1]. In general, presence of circuit switching activity inside a HT-infected IC on application of test patterns is necessary for Trojan detection through transient current analysis; however, this is difficult to achieve in practice, in the absence of knowledge of the HT's structure and mode of operation. The proposed method isolates the effect of the contribution of HT in static current consumption and thus removes the requirement of switching activity for HT activation. Another observation was that the static current consumption behavior measured through each of the supply ports is unique, which is influenced by transistors that are in the vicinity of the supply port. Multiple supply port technique combined with power signal calibration technique was shown to increase detection sensitivity drastically.

Banga et al. developed a novel two-stage test generation technique that aims at magnifying the difference between side-channel signatures of infected and HT-free ICs [13–15]. Firstly, in the circuit partitioning stage, the IC under test is partitioned into regions based on structural connectivity. In the activity magnification stage, new test patterns concentrating on the identified regions are applied to magnify the difference in power profiles between the golden and Trojan-infected IC. Thus the region-aware pattern generation approach helps in identifying the potential HT insertion regions, by which activity within a portion of the circuit is increased, while that in the rest is simultaneously minimized. Rad et al. proposed a region-based transient power signal analysis for HT detection, to overcome the small ratio of HT-current to the circuit background current [89,90]. Again, supply current is measured from multiple ports individually by applying test patterns, and a statistical analysis of the transient current waveform generated is performed. Different process models for golden and HT-infected designs is produced, which is then calibrated for process variation.

Hou et al. [55] proposed a HT detection method that focuses on the intrinsic relationship between transient current and static current signature to eliminate the effects of process variation, enabling detection of malicious hardware modifications. By application of test vectors on IC under test, they acquired the transient current and static current signature. Then they compared the curve plot between transient current and static current signature with that of golden IC signature curve to determine the presence of HTs. Wilcox et al. [117] proposed a static leakage current characterization-based side-channel method to detect HT in ICs. Note that all modern ICs manufactured using nanometer-scale Complementary Metal-Oxide-Semiconductor (CMOS) device technology suffer from nonzero gate terminal current and OFF-state current, termed as "leakage current." Static leakage current is measured from multiple power ports on the chip. The authors proposed a novel chip-averaging method targeted for removing intra-die variations, thereby improving the Trojan's detection sensitivity. Scatter plots of currents along with PCA-based ellipse analysis was used to differentiate between random noise and anomalies due to HT. Scatterplots of currents measured from pairs of adjacent power ports are created for application of the 2-D ellipse statistical method. Ellipse statistical limits are derived from 30 ICs, and 12 other ICs are used for evaluation as control samples. Data points that fall outside the ellipse bounds are considered as true positive detections.

Lecomte et al. [67] described an on-chip monitoring methodology using embedded sensor network. It aims at checking the integrity of a whole production lot instead of checking for infection of a particular IC. Due to the presence of HT, power distribution of IC is modified resulting in drop of the static voltage in the glue logic and it is reflected in the sensor network. The principle behind this methodology is to detect, due to an embedded sensor network, an eventual alteration of the inner structure (the presence of an HT), a modification of its floorplan (rough counterfeit), or a degradation induced by the aging effect (reused IC). These alterations modify the IC power distribution, and in particular the static voltage drops in the glue logic and hence that in the sensor array.

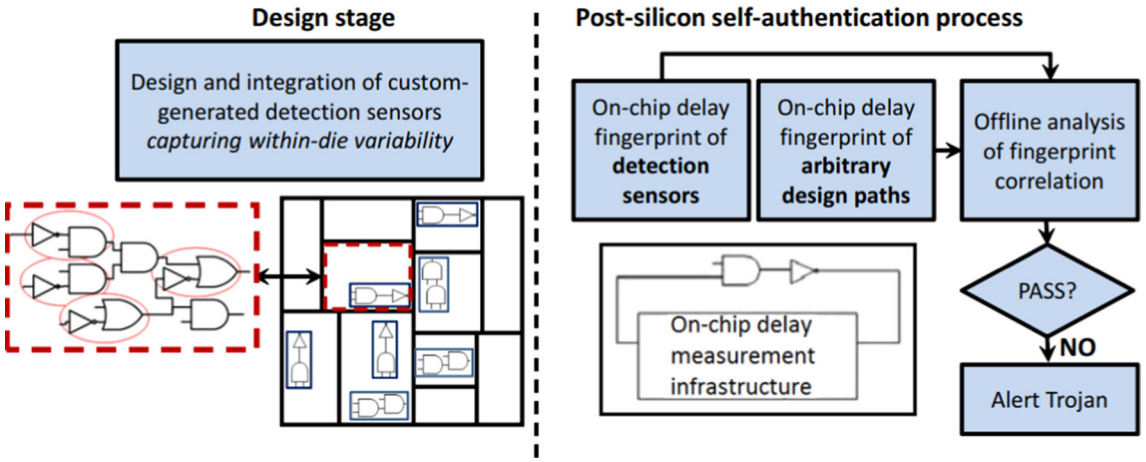

Fig. 7 Example of self-authentication framework based on [70]

3.1.2 Path delay characterization-based Hardware Trojan detection

Insertion of additional HT circuitry can modify the internal path delays of ICs, primarily because of the additional capacitive load of the inserted HT logic gates. Delay characterization-based side-channel analysis for HT detection was first introduced in [63]. A clock-delay measurement technique was used to measure selected register-to-register (shadow registers) path delays. HTs can be detected if one or a group of path delays exceeds the threshold which is determined based on process variations. Using the same approach, it has been shown in [92] that delay-based techniques combined with statistical analysis can enhance HT detection significantly even in the presence of high level of process variations. Jin et al. [127] proposed a fingerprinting method using path delay information of the entire chip. An IC has many delay paths having different characteristics, which can generate a series of path delay fingerprints. First, several instances of a particular IC are selected. Path delay information is collected by using high-coverage test patterns. These chips are then checked through reverse-engineering to obtain a collection of golden ICs. The delay characteristics are compared with golden IC delay fingerprints, to infer the presence or absence of HTs in them.

In most of the delay-based methods, some extra circuit is added to detect HTs, including *Ring Oscillators* [44,97,130]. But the main demerit of these methods is they suffer from relatively large area overhead. HT detection using delay characterization also suffers from the bottleneck of process variations and requirement of the existence of golden ICs. Also, it is extremely hard to detect a HT inserted on short delay paths in the circuit, as fast activation stimulus are required to test these paths, resulting in detection inaccuracy. Li et al. [70] described a self-authentication framework that uses on-chip detection sensors for self-authentication of each IC (Fig. 7). Using the information from the detection sensors, a prediction is made about the delay of each considered path, which is then compared against the on-chip delay fingerprint made directly on the path. The two fingerprints are then analyzed to detect a HT.

Lamech and Plusquellic proposed a embedded test structure termed *REBEL* (from "REgional dELay Behavior") for detecting HTs in [66], with about 10% design overhead. REBEL uses delay chain in a segment of a scan chain. It validates delay measurement easily and also speed up the process. Esirci et al. in [41] devised a HT detection mechanism where two highly correlated paths are selected, one of which is the shortest path that passes through an interconnect suspected to have an HT. Using an efficient algorithm, a suspected HT-infected path is extracted, followed by extraction of multiple correlated path using delay parameter in the circuit. After extracting the correlated path candidates, the correlation coefficient of each candidate with respect to suspected path is computed and the one having the highest correlation is marked as the correlated path. The resultant path delay ratio(ratio of suspected path to correlated path) values of the HT-infected IC must be a significantly larger from the ratio values of the HT-free IC. A delay-based detection method has been proposed in [59] which uses a high-resolution on-chip embedded test structure called a time-to-digital converter (TDC) that provides timing resolution of approximately 25 ps. TDC is used to obtain high-resolution measurements of path delay. A novel chip-averaging technique is also devised which reduce the adverse effects of intra-die process variations on HT detection.

At the other end of the spectrum, Cha and Gupta in [27] proposed a path selection scheme that intentionally targets paths having the smallest delay values, in order to maximize the impact of a HT on each path's delay. They argue that as the ratio of delay of path to standard deviation of process variations increases, smaller the number of IC instances that are needed to be tested. They expressed the problem of finding a minimal set of paths as an integer linear programming prob-

lem. Amelian et al. developed an algorithm in [8] that takes a circuit netlist as input and returns k of the shortest paths. For detecting HTs, if delay of each of the k paths is different from the corresponding path in golden circuit, the IC is marked as HT-infected. In [128], the authors have leveraged symmetries in different transistor-level paths having same topology to detect HT. Equivalent delays are possible due to the fact that inter-die process variations (process variations affecting portions of different chips) affect them identically, and intra-die variations (process variations affecting portions of the same chip) will be limited if the paths are in close proximity. This method has limited usage for cases when symmetries do not exist everywhere in an IC. Therefore, it should be combined with other HT detection techniques.

Clock glitches have been proposed in [79] to measure path delays for authenticating the FPGA IP block and detecting HT anomalies. Similarly, clock glitching method has been used in [42] to measure path delays, and various statistical techniques have been used to reduce the adverse effects of inter-die and intra-die process variations. In [38], Cui et al. proposed a two-phase technique using the order of path delay in path pairs for HT detection. During design phase, a full-cover path set is generated that covers all nets of the design. The order of paths in all pairs serves as the fingerprint of the IC design. During test phase, the actual delay of the paths in the full-cover path set is measured in the IC under test, and the order of paths in these pairs is compared to the fingerprint generated in the design phase. The paths connecting to a HT will add extra delay, thus changing the relative order of some path pairs. By comparing the orders of the path pairs, the presence or absence of an inserted HT can be inferred.

Sabri et al. proposed an integrated methodology consisting of SAT-based test pattern generation and delay-based Hardware Trojan detection called *DELPA* [98]. In pre-silicon stage, the test pattern pairs are generated in such a way that all the circuit paths are activated to expose any malicious timing variations. The SAT-based test scheme uses a SAT solver to generate test pattern so as to perform path-delay analysis alongside the clock-sweeping technique for measuring the genuine path-delay. By applying the generated test pattern pairs to the golden IC in post-silicon stage, the genuine functionality and timing specifications are obtained in the presence of process variation. Then in detection phase, the suspected ICs' fingerprints are compared with the golden one to distinguish Trojan-infected ICs. Further, MUX-based debugging technique is used to localize the trace of inserted Trojans.

3.1.3 Electromagnetic radiation analysis-based Hardware Trojan detection

Electromagnetic (EM) radiation arises due to flow of current inside a chip. So, HTs will indirectly influence the EM radiation of the IC. While requiring somewhat specialized equipment and expertise, EM-measurement-based SCA has many advantages over other side channels. In EM measurement, a set of magnetic near-field probes is used to acquire the radiation pattern. Non-contact detection of HT is possible by placing the probe right above the chip in its close vicinity. The probe can further be attached to a stepper mechanism which can be controlled manually to step over the chip to gather a detailed radiation map. HT detection based on EM radiation was first proposed in [106]. A sequential Denial-of-Service (DoS) HT was placed next to the I/Os of the FPGA and their effect on the EM emissions was measured. By comparing the EM emission of HT-infected FPGAs with the golden one, it was found that HTs placed at the corner of the FPGA are easier to detect than at the center due to their proximity to power line. A similar approach was presented in [79,81] where presence of HT was detected by comparing directly a golden *Advanced Encryption Standard* (AES) encryption hardware execution EM traces, and HT-infected AES hardware execution traces with the same plaintext. Jap et al. proposed a novel HT detection method in [61], based on an one-class *Support Vector Machine* (SVM) machine learning algorithm, using EM-based side-channel profiling.

Balasch et al. proposed a two-phase technique in [12]: a *learning phase* to generate a golden fingerprint by collecting sufficient amount of EM measurement data from a known golden IC, followed by a *matching phase* to collect EM measurements from the ICs, and apply *Welch's t test* to determine whether the readings comes from the same distribution as the golden fingerprint. The analysis also suggested that just like previous, specific signal routes present in some of the HTs are easier to detect than others. A major shortcoming of this method is the requirement of a golden chip. In [52], He et al. proposed a HT detection method which uses golden chip-free EM spectrum modeling and side-channel statistical analyzing. For the modeling process, simulation data from RTL design are used to generate the EM spectra and the magnitude of each frequency spot. EM spectra are calculated by summing up the transitions of all registers/LUTs in the circuit. Also, target FPGA implementation is taken into consideration. To distinguish the simulated EM spectrum from the extracted EM spectra in actual IC, *Chirp Z-transform* (CZT) and Euclidean distance algorithm are used.

Chen et al. [31] described a HT detection method by analyzing the EM radiation of clock trees in an IC, specifically, a FPGA. First, EM radiation emitted by the FPGA clock tree is collected by scanning the surface of the FPGA, and obtain various EM profiles. Then, 2-D PCA projection is applied on the EM profiles and two-norm of transformed matrix is calculated. Lastly, a backpropagation neural network is trained to identify FPGAs with/without Trojans.

3.1.4 Thermal profile characterization-based Hardware Trojan detection

When a HT gets activated at run-time, the power consumption of the IC containing is expected to change, and the same is reflected in the IC's thermal profile. Most modern electronic systems are already equipped with thermal sensors, which can then be utilized for temperature profiling based HT detection. This is the main insight behind thermal profiling-based HT detection, and like all other SCA techniques, they also suffer from issues of process variation and possible requirement of a golden model. Temperature characterization for HT detection have been explored in [18,45]. It generally consists of three phases: design, test, and run-time phases. In the design phase, the HT-free design's thermal model is modelled using prototype ICs. During test phase, HT-infected ICs are detected based on other test-time detection schemes. Also we calibrate each IC due to fabrication variation. The run-time phase integrates the information from the previous phases with thermal sensor measurements to detect inserted HTs that are activated at run-time. Another mechanism was proposed that utilizes the correlation between local sensors and keeps a track of the IC's thermal profile using a *Kalman filter* (KF) which explicitly accounts for noise measurement.

3.1.5 Gate-level characterization

A post-silicon technique called "Gate-level Characterization" (GLC) has been explored in the literature [7,87,113–115] for gate-level timing and power estimation, to aid HT detection. Side-channel characteristics such as delay and power consumption are modeled using a linear system of equations of gate characteristics, and deviation from the expected value is modeled statistically. HTs are detected based on the deviations from golden signatures. But the linear system of equations is expected to be greater than that of number of gates which results in an increase in measurement cost as the circuit size grows. Although elegant in theory, since the number of equations increases rapidly as the size of the design increases, the difficulty of scalability of this approach based to industrial-scale designs is a severe practical limitation.

3.1.6 Multi-parameter analysis for Hardware Trojan detection

Multi-parameter analysis refers to the usage of more than one side-channel characteristics for detection of HT. In [77], the authors used the correlation between maximum operating frequency (F_{max}) and transient (dynamic) current (I_{DDT}) of an IC to eliminate the impact of process variation of the IC. The authors demonstrated through experimental results that by combining the approach with logic testing, test time can be decreased and HT detection coverage can be increased. Multimodal Trojan detection presented in [64] similarly shows that combining several side-channel parameters results in increase of the HT detection sensitivity. In [46], a HT detection methodology based on combination of different methods, including both logic testing and side-channel analysis method. It consists of a three-stage methodology that is deployed at the design time of an FPGA IP core and is extended during its operation.

3.1.7 Backscattering side-channel analysis for Hardware Trojan detection

In [82], the authors introduced backscattering SCA for HT detection. The technique is powerful, as it is also able to detect the existence of different types of dormant HTs and HTs which have very less activity after being triggered, while being tolerant to process variations. A sinusoidal EM signal is transmitted at a certain frequency toward a FPGA chip, and the backscattered signal is received and recorded. The backscattered signal, if modulated by on-chip switching activity, should contain not only a component at the sent frequency, but also side-band components at different frequencies. Figure 8 shows the basic principality of backscattering. Advantages of using backscattering for HT detection are that the backscattered signal carries information about the current state of on-chip circuits and their impedance values, unlike other SCA techniques that revolve around information on small changes in current consumption. Furthermore, strength of the backscattered signal can be modulated; its frequency can be shifted to avoid noise, interference, and poor signal propagation, and it can be more accurately focused on a specific part of the chip. Like other HT detection techniques based on side-channel analysis, first a circuit that is known to be HT-free is characterized as a golden reference, and then an unknown circuit is classified into HT-free or HT-infected based on a statistical model. Experimental results showed that the backscattering-based HT detection, after training with an HT-free design on one DE0-CV board, accurately detected dormant HTs for three

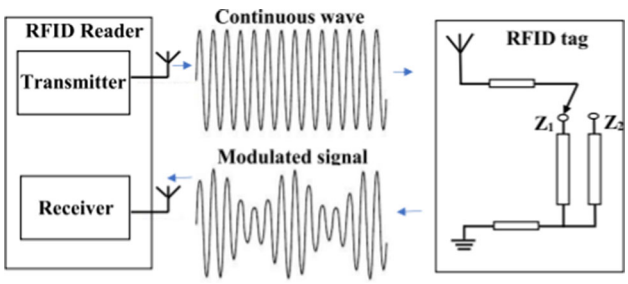

Fig. 8 Principality of backscattering side-channel analysis for Hardware Trojan detection

different HT designs, on nine other DE0-CV boards with no false positives. In a recent work [3], the authors described a "near-field" backscattering measurement setup for detection of Trojans.

3.2 Image processing-based Trojan detection

SCA techniques of HT detection methods discussed in previous sections have been demonstrated to be efficient in detecting Hardware Trojans. However, two limitations can impact the reliability and efficiency of these methods. Firstly, the amplitude of change in signal values in the presence of HTs can be small enough to stay within possible process variation tolerances. The second challenge, an even greater one, is the requirement of a golden IC sample to validate the side-channel characteristics of IC under test.

Image processing-based HT detection methods offer an attractive alternative. Some of them use destructive reverse-engineering techniques to de-package an IC and obtain images of each layer, in order to reconstruct the design-for-trust authentication of the IC under test [88]. Destructive reverse engineering has the potential of achieving high accuracy rate of HT detection in IC, but it incurs high cost, since the IC under test becomes unusable after the test. Since HTs consists of altered cells or circuit connections, a common practise of inspection is to inspect only the active layer or metal layer IC [88]. This method reduces delayering cost attached with full reverse-engineering and has been found to be quite accurate and robust to detect various HTs.

Another recent innovation in image-based HT detection involves optical and thermal imaging of the IC chips. These methods are non-destructive in nature and thus less costly than the above method. But the main disadvantage lies in the fact that they also suffer from the variations in the manufacturing process. Also, the time required to image the chips and the resolution of backside imaging is challenging.

In summary, HT detection systems using these two types of methods follow the following general image processing pipeline, as below:

1. Image Acquisition: Generally Scanning Electron Microscope (SEM) and Near-Infrared (NIR) images are collected from real ICs to detect HTs.
2. Image Preprocessing: Common preprocessing techniques include the convolution of SEM images with 2-D Fourier, Gaussian, median filters, image rotation correction and histogram equalization.
3. Feature Extraction: Shape features and thermal map properties are mostly used for detecting HTs.
4. Classification: ML-based techniques such as k-nearest neighbor, support vector machine classifiers, neural networks and image matching are mostly used for the detection of HTs.

Figure 9 shows the general flow involved in Image Processing-based HT detection schemes.

3.2.1 Destructive techniques and layout analysis based

Over the last few years, the advancements in the field of optical or X-ray microscopy, these machines are now relatively easily available, and can be rented or purchased easily. Courbon et al. [32,33] proposed the basic concept of image processing to detect Trojans using SEM images by correlation with a golden circuit or by correlation with GDSII file. They have used front-side SEM imaging and basic image processing functions like histogram equalization and image subtraction to detect HTs inserted in the form of logic gates and transistors. However, the scope of the approach was relatively limited, as it covered only addition of logic gates or transistors as a Trojan insertion approach.

Bao et al. [19,20] proposed a machine learning-based technique to detect Trojans. Their approach detects the changes in the metal layers in the ICs, but does not cover the detection of Trojans implemented by modifying substrate. Images obtained from the imaging step of reverse-engineering are used, and features are extracted from them to characterize an IC's physical layout. The authors developed two classifiers: a SVM classifier and a K-Means clustering approach that

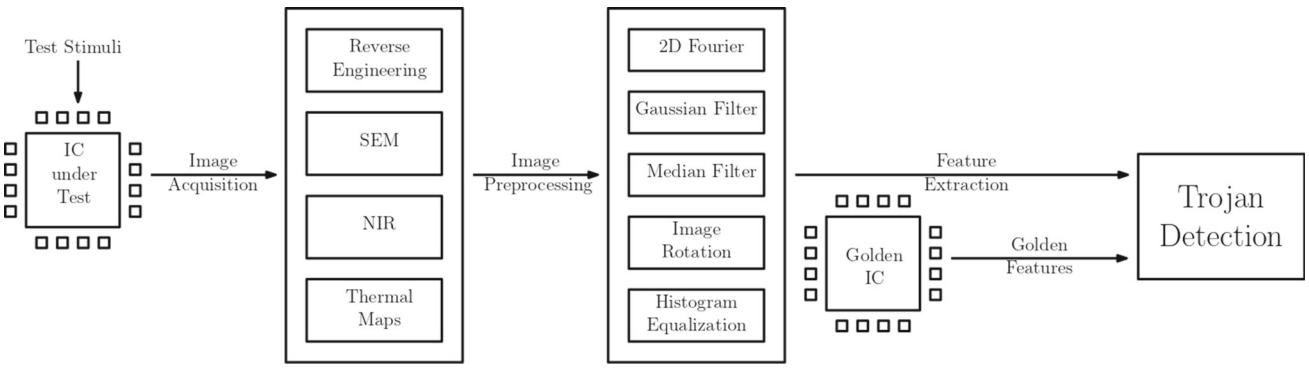

Fig. 9 Image processing-based Trojan detection workflow

distinguishes between expected and suspicious structures in the ICs and ultimately detects HT-infected ICs.

Vashistha et al. presented the *Trojan scanner*, which compares a trusted GDSII layout (golden layout) and scanning electron microscope (SEM) images to identify the malicious modifications made in the IC netlist during the manufacturing process [109]. The process is semi-invasive, where a chip's backside is thinned so as to get a detailed imaging of the active layer. A unique descriptor for each standard digital logic cell is prepared based on different features using computer vision algorithms. Then a machine learning model is trained with golden layout and SEM images of an IC under authentication which can detect any modifications either in the form of additional gates or modified gates. The authors extended the approach in [103] with electrical testing to detect HTs. The process is based on combining backside imaging with logic tests. *Golden Gate Circuits*, a combination of logic gates and test infrastructure, was proposed to be inserted in the unused space of the IC. It is used to enhance the accuracy of the machine learning classifier.

Stern et al. developed *SPARTA*, which is a non-destructive backside laser probing approach for sequential Trojan detection [104,105]. *SPARTA* creates a 2-dimensional frequency map of the backside silicon using electro-optical frequency mapping (EOFM), which exposes the activity of clocked elements in the IC. Comparison of clock activity within a fabricated IC is made with the original clock tree created in the design phase, so the requirement is not of any golden samples, but rather the golden design since the layout sent to the foundry should exactly match the fabricated IC.

3.2.2 Optical imaging-based techniques

Zhou et al. proposed an optical method that can rapidly and accurately detect malicious tampering and HTs inserted during the chip fabrication stage [134,135]. They engineer the filler cells in a standard cell library to be highly reflective at near-IR wavelengths which produces bright spots that can be readily observed in an optical image taken through the backside of the chip. The pattern produced by their locations acts an easily measured watermark of the circuit layout without any measured "golden chip" reference. Any replacement, modification or re-arrangement of these cells to insert a Trojan can therefore be detected through rapid post-fabrication backside imaging. The setup described by the authors was able to detect HTs that have power consumption less than 2% of the total power consumption, with area that is less than 0.1% of the total area and was robust to measurement noise and $\pm 10\%$ process variations.

3.2.3 Thermal imaging-based techniques

The authors in [56,85] proposed a multimodal characterization framework that uses thermal maps and power maps to detect and locate HT in ICs. Infrared imaging was performed to obtain the thermal maps of ICs. Then, random vectors were applied to the ICs, and estimated power trace of each block was obtained using simulation, to create the steady-state thermal maps of the ICs. These golden thermal maps are used as the training set to perform 2-D PCA on the thermal maps under tests for HT detection.

Cozzi et al. proposed a low-cost compact measurement setup, based on a mono pixel IR sensor which provided a large acquisition bandwidth and a higher detectivity at equivalent temperatures [35]. They used the lock-in thermography-based correlation technique to compute amplitude and phase values at every position on the die. The process included comparing golden thermal maps generated from measurements done on a golden IC with the corresponding thermal maps obtained from the IC under test. HTs were detected by simple difference of means between corresponding positions of the maps or using statistical tests such as the Welch's t-test. In [34], the authors improved the efficiency of the procedure by applying Welch's t-test to phase values. Later in [36], the authors proposed to improve the clarity of thermal images by applying Kolmogorov–Smirnov test.

In [116], the authors described a new HT detection methodology that is based on thermal maps and Inception neural networks (INNs). 50,000 thermal maps related with the Trojan-free chip and 100,000 thermal maps (400 emulated Trojan-infected chips are used, and each emulated chip generates 250 thermal maps) pertaining to the emulated Trojan-infected chips are used for training the CNNs and INNs. By utilizing the customized filters within the INNs to analyze the thermal maps of ICs, the authors were able to achieve a classification accuracy of the embedded Trojans of over 98.2%.

Yang et al. proposed a chip-level HT detection technique by exploring the relationship between time and temperature changes [126]. The method tracked the temperature rise process of the chip by infrared camera and extracted the time feature as the detection basis. The functioning Hardware Trojan ICs are assumed to spend a shorter time to reach a temperature threshold; the time versus temperature changes was analyzed. This method is low cost and easy to implement; however, its accuracy is affected by environmental changes.

4 Logic testing-based Hardware Trojan detection

We now shift our attention to logic testing-based HT detection.

4.1 Post-silicon logic testing

In post-silicon logic testing techniques, external test stimuli are applied to the ICs, and its response is compared with

the expected outputs. Hardware Trojans are detected if the response obtained for one or more test vectors differ from the expected ones. HTs which cause functionality change in IC are most likely to be detected by logic testing methods. However, an intelligent adversary would naturally design stealthy HTs which remain dormant most of the time and evade the relatively small number of test vectors applied during post-manufacturing testing, until triggered once deployed [11,30,39,99,110,118]. So, the ultimate aim of logic testing schemes is to activate dormant HTs with the minimum number of test patterns. The test patterns are chosen such that they will trigger HTs to cause malfunction. In the context of Fig. 2, an adversary would choose the signals with low controllability values for logic-0 or logic-1 as trigger condition [11,30,39,99,118], as it is unlikely to set all the trigger activation signals to their respective low controllability values from primary inputs. An additional challenge is to propagate the logic malfunction caused a triggered HT to the primary output, only then the HT is detected.

Although some of the earliest propose approaches of HT detection employed logic testing [30,118], relatively few works have been published on this topic compared to SCA-based approaches for HT detection. The major challenge is the prohibitively large size of the HT design space that the test generation algorithm has to consider, in the absence of any *a priori* knowledge about them. However, in [48], pre-silicon-based logic testing tool has been proposed with a powerful HT detection algorithm called *Hardware Trojan Catcher (HaTCh)*. The Trojans considered in this work are deterministic HTs proposed in [49] and proved that HaTCh offers negligible false negative rate and controllable false positive rate.

Hence, conventional Automatic Test Pattern Generation (ATPG) algorithms to generate directed test patterns are often ineffective for HT detection. Random test vector-based HT detection is also ineffective and typically achieves extremely poor detection coverage [28]. In the rest of this section, we discuss logic testing-based techniques to detect and also facilitate other HT detection approaches such as SCA. We cover different a wide range of approaches, including: modified ATPG, N-detect ATPG, redundant circuit detection, code coverage analysis, rare node activation, etc.

4.2 Automatic test pattern generation (ATPG)

In [132], a case study is performed to identify HT-infected circuit by using code coverage analysis and ATPG. In this proposed scheme, in the first step through formal verification and code coverage analysis, redundant and unused parts of the design are identified, followed by the ATPG tool in the second step to activate dormant HTs with some particular patterns. For RTL descriptions of the constituent circuits, code coverage analysis was carried out in the design to verify that there are no rarely triggering events or hidden scenarios to leak secret information and serve as a backdoor [17,132]. Even with the 100% code coverage, HTs may exist in the design.

Both Zhang et al. and Wolff et al. have proposed Trojan detection schemes that directly uses ATPG tools or algorithms to generate test patterns [118,132], but only with moderate success, limited to small circuits. The known complexity of full-sequential ATPG for non-scan sequential designs reduces the effectiveness of full-sequential ATPG-based test pattern generation for HT detection. Banga et al. in [17] proposed a HT detection scheme enhanced by SAT solver, utilizing N-detect full-scan ATPG for test generation [10]. However, for those designs with non-scan regions, this method of test generation fails.

To overcome the difficulty of applying ATPG tools and algorithms to generate effective sets of test vectors of reasonable size, while having high HT detection coverage, a statistical technique termed "Multiple Excitation of Rare Occurrence" (MERO) was proposed by Chakraborty et al. [30], in which the probability of HT activation is increased by exciting several rare nodes multiple times to their rare values. MERO is motivated by the N-detect test methodology [10] previously proposed for microprocessor testing and applied to gate-level design descriptions. In effect, a subset of the most likely HT instances are selected. It also includes heuristics to improve the HT detection coverage, but like any other statistical ATPG technique, cannot guarantee HT detection. MERO was later extended and improved in [99], to generate more compact set of test patterns using, genetic algorithms, Boolean satisfiability (SAT) solvers, and improved heuristics, while achieving higher HT detection coverage. In [62], Jha et al. have proposed a probabilistic approach of HT detection. For a specific set of input patterns (directed test generation), a unique probabilistic signature of the circuit is constructed and compared with that for a known HT-free circuit. The difference in these signatures, if any, points to the presence of HT. The difficulty of probabilistic approaches for HT detection lies in their relatively large test generation time, as well as their inability to generate effective test vectors to detect inserted HTs with complicated trigger logic conditions.

In recent years, formal methods and ATPG schemes have been combined in several works HT detection, and also for verifying security properties [76]. In [37], Cruz et al. have combined model checking technique with ATPG to generate a test set. In this approach, the entire design is partitioned based on an inserted scan chain, and for non-scan circuit elements, constraints are engendered by using model checking. ATPG will generate testset based on the constraints and the scan-chain elements. Design inserted with partial scan chain can use this approach for test generation. Still, none of the

existing ATPG-based HT detection scheme can be claimed to be scalable to detect stealthy HTs in industry-scale designs.

In [86], the authors proposed an efficient logic testing approach for HT detection that utilizes a stochastic reinforcement learning framework to enable fast and automated generation of effective tests. For a given circuit, the approach considers both rareness and the testability of signals using a combination of *Sandia Controllability/Observability Analysis Program* (SCOAP) measurement and dynamic simulation. Next, the intermediate results from analysis are fed into the reinforcement learning model as primary inputs which are trained with a stochastic learning scheme to generate test vectors. Experimental results demonstrated that the approach can drastically reduce the test generation time while it is able to detect a vast majority of the Trojans in all benchmarks compared to state-of-the-art methods.

4.3 Run-time monitoring of Hardware Trojan activity

Although the Trojan activation schemes are useful to trigger Trojan action, they can detect Trojan only in the test mode. Hence to detect Trojan during the normal operating mode, *Run-time Monitoring*. Most of the Hardware Trojans are stealthy in behavior, and it is highly desirable to detect any inserted Trojan before deployment. However, since the state-of-the-art Hardware Trojan detection and prevention schemes cannot guarantee detection coverage of overall classes of Trojan, run-time monitoring approaches emerge as a defense mechanism. Security monitors have been proposed in [2,21] as a real-time functionality monitoring unit of conventional ASICs by adding reconfigurable logic to the design. Finite state machines (FSM) are included in the original design to monitor the signal behavior of the design. Whenever there are unwanted events in the interested signals, security monitors will give alerts or alarms to initiate countermeasures. These security monitors can be customized to identify malfunction created by Hardware Trojan, such as access to some protected memory space or entering the test mode during regular operation. However, it has been assumed that the attacker cannot use the security monitors circuitry. In [24], Bloom et al. had proposed a module that acts as a verifiable *hardware guard*, which has been applied to identify Trojan during run-time execution of the processor. The operating system (OS) checks the functionality periodically toward the DoS and privilege escalation attacks. A *memory guard* which checks the memory concurrently was proposed in [25]. Ngo et al. in [80] had proposed a circuit encoding technique, called *linear complementary pair* (LCP) code to detect and prevent Hardware Trojan. Invalid codewords are produced by LCP whenever an embedded Trojan activates and causes a malfunction. These invalid codes are helpful to detect Trojan, and this method is resistant to side-channel and fault-injection attacks. However, although run-time monitoring schemes can potentially reach close to 100% Hardware Trojan detection success rate, the hardware overhead is quite high. But even with the inserted Trojan detected, it is not always easy or even feasible to replace the infected IC from the system. This motivates Trojan prevention techniques, as described next.

4.4 Test pattern generation to aid side-channel analysis

Besides logic testing, test patterns are also generated to stimulate ICs in ways that facilitate other HT detection approaches, with the aim of improving HT detection coverage. For example, ATPG to enhance the sensitivity of side-channel analysis for HT detection has been widely explored [16,62,101,118]. However, again the need for a golden reference and process variation effects create major challenges in the efficacy of these techniques. Hence, in [93,95], Sree et al. proposed a divide-and-conquer technique to detect HTs, without referring to any golden chip, and validated the proposed technique using a metric termed the *power metric*. In this technique, the entire circuit design is divided into segments or regions, and the set of input patterns are applied to extract the power signature of those regions. The power signature of one region is compared with that of structurally similar to other regions, to cancel process variation effects. In addition, test vectors are generated and applied to toggle each node of the segment under test, which aids HT detection.

Another such algorithm for test pattern generation is described in [57,58] called "Multiple Excitation of Rare Switching" (MERS), which is an improvement of the MERO algorithm. The order of test vectors also matters in MERS, as the amount of switching in a circuit depends on the order in which pairs of vectors are applied. To further improve sensitivity to SCA, Hamming-distance-based reordering and simulation-based reordering was adopted. However, it was found that the test generation time using MERS grows exponentially with circuit complexity. Also, the increase in side-channel sensitivity was found to be marginal in the face of process variations. To overcome these limitations, the authors of [74] developed a *Genetic Algorithm* (GA)-based test generation algorithm that can lead to drastic increase in sensitivity, while significantly reducing the test generation time. In a recent work [75], the authors proposed a SAT-based ATPG technique to aid HT detection by delay-based SCA. It maximizes observable path delays by changing critical paths to activate trigger conditions. A Hamming-distance-based reordering is done on the generated tests to increase the probability of constructing a critical path from the trigger to the payload and maximize the deviation in delay between a golden IC and HT-infected IC.

In another recent work [83], a self-referencing superposition on circuit activity has been proposed to enhance SCA

techniques. This is achieved by magnifying the Trojan circuit effect and canceling the non-Trojan noises. To determine the magnitude of intra-die process variation, the authors used evaluation criteria called *Super Relative Power Difference(S-RPD)*. S-RPD computes the magnitude of the difference between expected nominal power and the observed power on application of two different test patterns. The effectiveness of Trojan circuit detection is determined from the maximum S-RPD intra-die process variation magnitude. Later, the same authors proposed a three-phase method HT detection technique in [84]. In Phase 1, suspicious signals are excited from the Trojan circuitry using high-coverage test patterns. In Phase 2, that suspicious signal is magnified and any other signals that are caused simply by process variation are weeded away. In Phase 3, the Trojan circuit is isolated through a test pattern superposition method. The principle of superposition dictates that the net composition of responses for multiple independent test pattern is equivalent to the response of those test patterns applied concurrently. The application of superposition identifies a difference between our first and second test patterns which isolates and exposes the Trojan signal to its full magnitude resulting in Trojan detection.

5 Design techniques to enhance circuit testability for Hardware Trojan detection

We now discuss the design techniques that enhance the circuit testability for hardware Trojan detection. These techniques either enhance Trojan detection or improve detection accuracy at the design level or enhancing Trojan resolution [91] or to prevent Trojan insertion at design level [29]. In [91], authors proposed a statistical approach of Trojan detection based on the analysis of regional transient power supply signals. This work analyzes the relative effectiveness of four different signal calibration techniques to reduce the effect of process and test environment variations. The Trojan resolution of the proposed transient signals (AC)-based method has been enhanced by the signal calibration component, namely AC sampling. The impedance variations in the chip and test environments are captured while calibrating AC sampling, under the condition that the sample is collected close in time to the delivery of the calibration status. Some of the techniques that facilitate Trojan detection have been discussed next.

5.1 ATPG methods for Hardware Trojan activation

Trojan activation [131] is a technique used to build trust in the design by accelerating the Trojan detection process. The malicious design embedded in the original design will help activate a stealthy Trojan so that the Trojan detection schemes will easily predict the Trojan's presence. Power analysis-based Trojan detection schemes use these Trojan activation methods. The main idea is that whenever an embedded Trojan is activated, it consumes more power, and it further helps to distinguish the power traces of Trojan-infected circuits from the golden circuit. methods. The main idea is that whenever an embed is categorized as *region-free Trojan activation* and *region-aware Trojan activation*. The main idea is to organize a group of related gates in a complex design to form a *region* [16]. They have described as follows:

5.1.1 Region-free Trojan activation

These methods will depend on accidental or systematic Trojan activation, and they are independent of the region. A proper example is a randomization-based probabilistic approach of Trojan detection proposed by Jha et al. in [62]. In this scheme, a unique probabilistic signature of the circuit is constructed for the specific input patterns applied and compared with the original circuit. The presence of Trojan is confirmed if there exists a difference in the outputs. In the case of manufactured ICs, input patterns are applied based on probability to obtain a confidence level regarding whether the original design and the fabricated chip are similar to each other. In [119], rarely activated nets have been used as Trojan triggers, and low observability nets have been chosen as a payload. The set of vectors thus generated to enable rarely triggering nets are then combined with traditional ATPG methods to activate a Trojan. Simultaneously, the limitations of region-free schemes are relatively low detection complexity, and high computational complexity as the entire design has enabled Trojan action. This motivates the region-aware Trojan activation approaches, described next.

5.1.2 Region-aware Trojan activation

These techniques rely on a divide-and-conquer paradigm to partition a given design into smaller *regions* and then focus on activating the Trojans in each of these regions individually. Banga et al. [16] had developed a test generation-based two-stage technique to magnify the power signatures extracted from Trojan-infected and Trojan-free ICs. In this technique, activity of each region is magnified by applying well-designed input patterns, and the corresponding power signature is measured. Then, the difference in the power signatures between the Trojan-infected and Trojan-free design will detect the Trojan. Saran et al. in [101] proposed a segmentation-based Trojan detection by extracting the fingerprint of a specific region, and comparing it with the corresponding fingerprint of the golden design. The hardware Trojan's existence is ensured if the fingerprints differ from each other. Ranjani et al. [95] had demonstrated a "golden chip free" Trojan detection technique by dividing the entire design into regions and comparing the power signature of

one region with other similar regions. The idea is for the same set of input patterns, the power signatures of the similar regions should be the same if the power signature differs due to Trojan's presence. This method is further enhanced in [93] by choosing a power metric as an evaluation factor. For a Trojan-free design, the power metric will be "1" and in case of Trojan-infected design, the power metric will not be "1". The reason is that the extra power consumption of the Trojan module will modify the parameter value. Thus, the Trojan activation schemes support the other Trojan detection schemes to provide absolute results even for more complex circuits.

5.2 Hardware Trojan prevention techniques

Other than hardware Trojan detection schemes, Trojan prevention schemes have been proposed for over a decade. One such Trojan prevention technique is an obfuscation technique proposed in [29]. In this key-based obfuscation technique, the state transition function of the design is modified by expanding its reachable state space and enabling the circuit to operate in two distinct modes, namely *the normal mode* and *the obfuscation mode*. The rareness of the internal circuit nodes is modified by obfuscation; as a result, the adversary finds it difficult to insert hardware Trojan, thus providing security at modest design overhead. The *Scalable Attack-Resistant Obfuscation* (SARO) strong obfuscation technique has been proposed in [6], which not only locks the design but also hides any structural signatures that might help attackers gain information about the system. Additionally, the proposed approach applies a design modification process that is randomized in different design aspects, which significantly reduces the accuracy of structural analysis attacks that are based on machine learning and pattern recognition. The authors in [9], had developed a database of hardware obfuscation open source benchmarks. These benchmarks include some circuits which are obfuscated by some common circuit obfuscation methods, created to facilitate the researchers in this domain. These benchmarks have been made publicly available on the Trust-Hub web portal [108]. The authors also evaluate the relative effectiveness of several candidate obfuscation approaches toward HT detection/prevention.

6 Hardware Trojan detection effectiveness estimation

HT detection approaches can be validated by a metric similar to *fault coverage* in traditional circuit testing as a measure of confidence, to quantify its effectiveness. Measuring HT detection coverage exactly is infeasible, since it is impossible to enumerate all HT instances. Hence, it is necessary to have a statistical measure to build an assurance of HT existence. In [132], Zhang et al. have used *coverage metrics* includes *code coverage and functional coverage* to estimate the assertions of all functions in the specification as properties. The code coverage metrics are used to identify the suspicious parts of the circuits. However, extra functionality in hardware design is identified by the verification approach of system specification and implementation [65]. In [111], authors proposed a criterion known as *control value* to identify a suspicious input, in a technique termed *Functional Analysis for Nearly Unused Circuit Identification* (FANCI). Here, the control value of an input w_1 on an output w_2 quantifies what fraction of the rows in the truth table for w_2 are directly influenced by w_1.

$$\text{Control Value}_(w_1, w_2) = \frac{\text{counter}(w_1)}{\text{size}(w_2)} \tag{1}$$

where counter(w_1) denotes the total number of rows of w_1 which determines the value of output w_2 in the truth table; size(w_2) denotes the total number of rows in which w_2 has a true value in the truth table. Once the control values of all inputs are calculated, a threshold is derived for the control value and the inputs whose control values are below the threshold are considered as the suspicious inputs. The threshold is calculated using a heuristic which is actually weighted average of the control values. The heuristic weights them by how often they are the only wire influencing the output to determine how much an output is influenced overall by its inputs. The purpose is to learn more about the output wire and less about the individual inputs. Trivially, the control values should be zero or one in absence of HT. The control values is probabilistically likely to vary by only a very small amount from threshold in presence of HT. Hicks et al. in [53] combined static and dynamic approaches of verification techniques and formulated the Trojan detection, by considering the suspicious circuits as the unused circuit identification (UCI). The UCI algorithm identifies the suspicious HT by tracing all the signal pairs with equal or similar properties. During run-time, an exception notification logic is added to the isolated suspicious logic. In [96], authors proposed the *Trojan Assurance Levels* (TAL) metric for the assessment of Trojan presence/absence by locating the insecure area of the chip design. The mathematical expression of TAL is derived by evaluating the circuit functionality, structure and functional interactions at different levels of abstraction. Hardware Trojan detection process is more efficient in the specific regions mentioned by TAL, as these are the potential regions of HT insertion.

In [30], trigger and HT coverage are computed by a random sampling approach. A specific set of HT structures are randomly selected based on the number of trigger nodes. Then, for any input pattern, Trojans with false trigger conditions are eliminated from further scrutiny. The circuit under

test is simulated by the input patterns in given set of vectors, and the triggering condition is checked for each vector. Trojan is detected when the applied vector propagates the malfunction effect to primary outputs. Trigger coverage and Trojan coverage are attained from the percentage of Trojans activated and detected. Effective Trojan measures are obtained by considering an adequate number of Trojans samples from the universal Trojan space. Further, to improve the HT detection efficiency, Li et al. in [68] proposed an acceleration approach based on signal word-level statistical properties with mean (μ), standard deviation (σ) and autocorrelation (ρ). This method increases the probability of HT activation, with less detection time by dramatically enhancing the rare nodes transition activity.

7 Removing the requirement of a "golden reference"

In [120], HT detection without the need for a golden reference was stated as one of the major research work still left unsolved. In this section, we present recent techniques which have been developed in the last few years to address this issue. Two types of golden models are generally used: (i) golden simulation model or (ii) golden IC instance. Narasimhan et al. made the first attempt to address the issue in [78], through a technique termed "Temporal Self-referencing" (TeSR). The procedure compares an IC instance's transient current signature with itself at different time instances. A Trojan-free IC's transient current signature will remain constant over different time instances while undergoing the same set of state transitions. But in a Trojan-infected circuit, the current signature will not be constant due to switching activity in the Trojan circuit. TeSR, however, only applies to sequential HTs, and its effectiveness may be limited by process variation.

In [129], the authors formulated the HT detection problem as an "outlier identification" problem, for a given set of side-channel signatures. The technique termed "HTOutlier" does not require the existence of a trustworthy golden IC for reference. Given a set of signatures generated by different input patterns for an IC wherein some are affected by the HT while others not, the technique detects whether HT exists or not. Outliers are detected by comparing each measured signature with an estimated value derived from other measured signatures. Experimental results showed that the backscattering-based HT detection, after training with an HT-free design on one DE0-CV board, accurately detected dormant HTs for three different HT designs, on nine other DE0-CV boards with no false positives. HTOutlier has the advantages of being somewhat both process variation resistant and scalable.

In [71], Liu et al. utilized on-chip process control monitors to capture process variations for each chip. They used golden parametric signature obtained by combining trusted simulation model, including parameters from the die, and statistically construct a trusted region for side-channel-based detection. The procedure is however difficult to execute due to the requirement of precise model of process variation. Xue et al.in [125] expressed the HT detection problem into a two-class machine learning classification problem. The model was trained using simulated ICs during the IC design phase. However, it does not take into consideration the reduction in performance due to inaccurately simulated IC. So, they proposed a co-training-based detection technique in [123,124]. First, two classification algorithms are trained using simulated IC's current signature during IC design flow. During testing, the two algorithms can identify different patterns in the unlabeled ICs, and thus will be able to label some of these ICs for the further training of the another algorithm.

Zheng et al. devised a technique termed "Self-similarity-based Microchip Integrity Analysis" (SeMIA) [133] for IC integrity validation. By analyzing dynamic current values in self-similar structures, it could identify both recycled chips and HT present without requiring any golden IC signatures. Since only self-similar and adjacent logic blocks are used, it eliminates the effect of inter-die and intra-die process variation. When used in conjunction with other HT detection techniques such as self-referencing techniques described above which are mostly applicable for sequential circuits, SeMIA can provide comprehensive HT detection coverage.

Xue and Ren in [122] devised a self-referencing-based HT detection methodology which doesn't require golden IC sample. Ideal model of static power consumption by clock tree and gates is generated by simulation and is multiplied by a scaling factor for estimating the effect of process variation on the measurement of the same. IC is partitioned into n segments, and it is ensured clock tree is in a separate single segment. Each segment uses separate power rails. For each segment, a set of m equations is formulated with assumption. Each segment consists of k gates with m static states. For a combinational circuit with x primary inputs, there are 2^x possible number of static states. Each static state is a state with specific static power consumption. Scaling factor is estimated by minimizing the sum of the squared difference of the calculated static power consumption and the measured static power consumption, under constraints for the values of the gate scaling factors. The estimated values of the scaling factor of the clock tree in each segment are used to assert presence of HT in a segment of the circuit.

Faezi et al. constructed a hierarchical temporal memory (HTM) model based on the power consumption of the IC, which monitors the side-channel emissions during the testing and at the run-time, interpolate the data and find the anomalies that indicate the existence of a HT in the IC [43]. Process variations have no effect on the proposed detection mechanism since it relies on IC under test for training. Since the

model is solely trained based on IC under test, golden chip is not required. HTM was found to detect 92.20% of triggered HTs while consuming less power compared to state-of-the-art machine learning techniques.

8 Role of machine learning in HT detection

We have discussed about physical parameters-based testing in Sect. 3. One common drawback in SCA-based detection techniques was the requirement of presence of golden IC. We have discussed this problem in details in Sect. 7. Other issue associated with these techniques that was mentioned commonly is the presence of noise and PVs. The accuracy of such SCA-based detection techniques relies heavily on the signal-to-noise ratio. They masks effects of Trojan instances and makes it difficult to separate from the measurable characteristics of an IC in side-channel analysis and image processing techniques [22]. Even though for large Trojans, it is possible to detect them with high detection accuracy, but the techniques performs poorly in the case of small Trojans. This problem persists even for HT detection techniques involving optical and thermal imaging of the IC chips.

Machine Learning algorithms are expected to reduce the impact of PVs and noise and improve the accuracy. Recently research along this line to determine whether the IC under test is infected by a Trojan instance or not has drawn plenty of attraction. Experiment performed in [85] showed that when random PVs increased from 20 to 40%, the HT detection accuracy in AES chips decreased from 89 to 44% and the false positive rate of detection increased from 5 to 8%.

Consider the popular preprocessing algorithms associated with machine learning. They can be very effective in reduce the impact of PVs and noise by extracting relevant features and reducing the data dimensionality. One major drawback that can arise is some HTs, having negligible effects on circuits, may be removed as unrelated or redundant features or noise, thus affecting the detection accuracy. Thus we can claim that the performance of ML-based Trojan detection is highly dependent on the selection of relevant features, learning models and parameters.

In the following section, we will analyze the various phases involved in ML methods to dissect the challenges and present state-of-the-art solutions through the prism of the research works discussed in the paper related to SCA and image processing-based techniques. Figure 10 represents the general steps involved in ML-based HT detection schemes.

8.1 Datasets

Dataset is an essential component of any ML-based algorithm. Acquisition of accurate data for proper detection of HT is required as it can directly affect the prediction results of the learning models. In this paper, we discuss physical detection techniques of hardware Trojans which include measurement of power, time, temperature and scanning of images. Each of these requires high accuracy-based devices to collect proper data which may or may not be cheap always. Also different features of the same ICs or different training models for the same features need different numbers of observations for learning the underlying pattern. Due to variance in post-silicon experiments to extract data, these numbers also depend on the environment in which the experiments

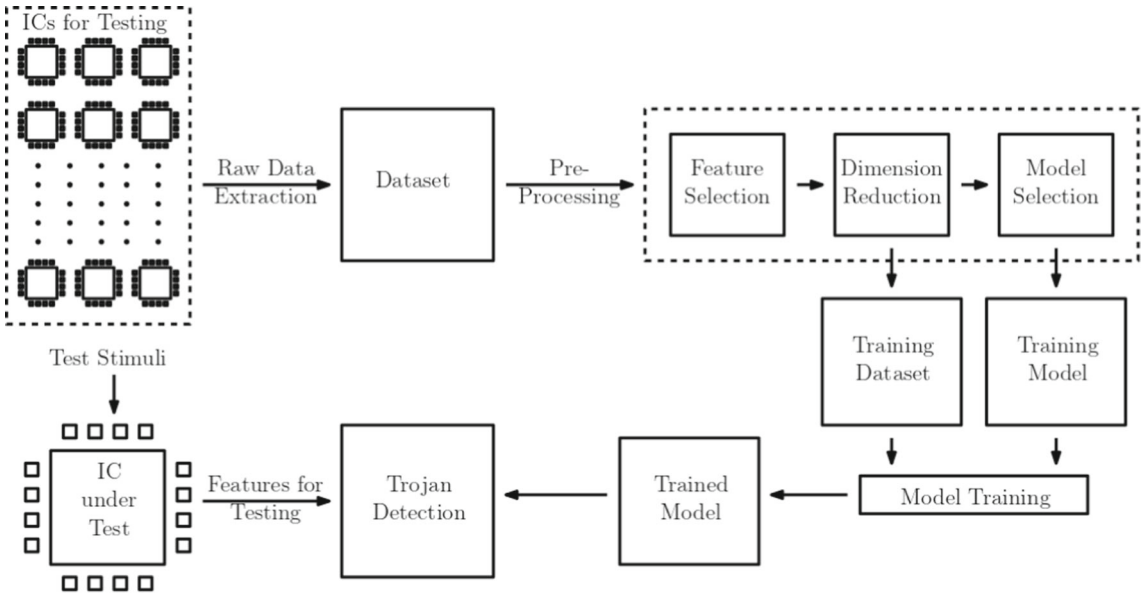

Fig. 10 General pipeline in machine learning-based Trojan detection

are carried out. Also, the risks of over-fitting and decreased generalization remain unsolved.

As discussed earlier, to eliminate the adverse effects of experimental and environmental noise, real data collected from hardware should be adequately de-noised when working with data with high levels of noise. For example, Gaussian filters have been found to be effective for noise elimination in image processing-based HT detection techniques. Even after year of research, removal of PVs and noise still remains an active area of research.

8.2 Features selection

In machine learning, it is always important to select effective relevant features from the raw dataset as it can significantly affect the performance of the learning models. Consider the work by Lodhi et al. in [72,73]. When they trained a K-NN model based on propagation delay signature, the HT detection accuracy was found to 93.12%. But when the same K-NN model was trained based on power consumption signature, the HT detection accuracy increased to 99.02%. This shows the importance of selecting essential features for achieving optimum results.

An important challenge in this aspect remains which of the features to be selected so as to get optimum detection accuracy. Also consider the case of multi-parameter analysis discussed in Sect. 3.1.6. It gives rise to another problem of how many such features to be selected or which combinations of features for achieving better results. No work have been focused on this aspect, and it remains an open research problem even in the domain of machine learning.

8.3 Dimensionality reduction

Continuing from previous section, we now focus on the dimension of the selected features. If feature dimension is too large, model learning time increases, whereas if feature dimension is too small, then over-fitting may occur leading to sub-optimal classification. Thus, both accuracy and computational efficiency of HT detection will depend on this step. In cases where the number of relevant features is very large and mutually correlated, redundant features will affect the trained models. Some features bear little or nor useful information and lead increase of computational complexity leading to over-fitting. Therefore, dimensionality reduction techniques are used to drop some features and improve the performance of the models.

PCA, a very common dimensionality reduction technique, has been shown to successfully address side-channel signature-based features [5]. In addition to PCA, there are several other dimensionality reduction methods that have been applied to side-channel features, e.g., 2DPCA in [85]. In both the cases, these techniques have been helpful to minimize the presence of noise.

Thus, the dimensionality reduction of HT-related features has been found to enhance the performance and reduce the utilization overhead of the learning models. Although primarily they have been found to be effective for side-channel analysis-based methods, their impact in image processing-based techniques is yet to be investigated. Also, no work has been found in the literature on how to determine the optimum number of dimensions for any particular HT detection problem like using elbow method.

8.4 Model selection

Proper selection of machine learning model depends most on the application problem to be solved at hand. Like a classification problem will require a supervised learning-based model, whereas a clustering problem will require an unsupervised learning-based model. For example, the work in [61] showed the application of both assuming that golden designs or ICs are available as training datasets or not. Moreover selection also depends on the feature type of the dataset. Example, side-channel signatures generally require K-NN or SVM type learning models [61], whereas image processing-based techniques also uses neural network-based models [116]. Machine learning techniques like outlier detection have appeared in studies to decrease the impact of random PVs and noise.

Selection of a proper ML model can help to achieve the best prediction outputs for HT detection problems. Each ML-based approach has its own computing requirements which also has to be considered. With the advancement in field of new ML models like reinforcement learning and deep learning, the problem of detection of HTs in IC can gather even more attraction. Lastly, concepts like ensemble learning and boosting have been suggested in the ML literature to improve the performance of ML models but not much of that has been explored in the field of HT detection.

9 Future research directions

Over the years, research in SCA-based HT detection have primarily focused on: (i) improving detection techniques by embedding additional circuitry; (ii) new methods of measuring side-channel signatures; (iii) post measurement statistical analysis of the side-channel measurements, and (iv) utilizing new measurable physical parameters for HT detection. All of these above lines of research have primarily been focused on dealing with challenges of reducing process variation and avoiding the requirement of a golden IC. Though traditionally side-channel analysis for HT detection has been based on power and path delay characterization, new techniques

have been developed in recent years that involve temperature and EM-profile characterization. Research involving power characterization-based SCA in the recent times mostly has focused on aim (iii) listed above, while the aims (i) and (ii) listed above have progressively become less significant. Temperature characterization-based SCA techniques for HT detection, other than the ones described in this paper, have been few and far between over the years and currently do not seem to be an active sub-area of research. Path delay and EM-based SCA for HT detection currently seem to be the most active approaches of research, and recent works featuring them have focused on all the aims listed above. Backscattering-based SCA can be considered to be the most promising current line of research motivated by aim (iv). With the ever-increasing adoption of machine learning (ML) techniques in all aspects of Hardware Security, advances are expected in fulfilling aim (iii) above. ML-based techniques have contributed heavily in decreasing the impact of environmental noise and process variations, extracting relevant features, reducing data dimensions, and partly reducing the dependence on a golden IC [51].

Logic testing-based HT detection schemes aim to detect malicious behavior of fabricated IC, only when the HT is triggered and its malfunction is manifested at the output ports of the IC. Thus, they are ineffective in detecting information leakage HTs [40]. They are also of little use when the HT is free-running and do not depend on an input trigger condition to initiate the malfunction. Since the trend of HTs designed in the recent years is decisively toward sophisticated information leakage HTs which do not directly affect any discernible malfunction, we envisage logic testing methods to be most useful in a supportive role to SCA, to enhance their HT detection sensitivity, as already described above in Sect. 4.4. This would expand their applicability to the detection of a much wider range of HTs.

To analyze the historical and recent research trends through their corresponding publication counts, we systematically searched the *IEEEXplore* digital library, with the key-phrase "Hardware Trojan," and selected the works related to HT detection. Based on the results of this database search, we plotted two graphs, as shown in Figs. 11 and 12. In Fig. 11, we find a noticeable superiority in the number of research publications on SCA-based techniques for HT detection, compared to logic testing-based HT detection techniques. Further, more publications have focused on logic testing that improves SCA-based HT detection, than those which simply focuses on logic testing. Figure 12 shows that the number of research publications on application of ML for HT detection have increased considerably over the years.

From the above discussion, it should be clear that research in HT detection based on SCA is ever-increasing. The main goal of current SCA-based approaches for HT detection, is to reduce the impact of inter-die and intra-die process variations

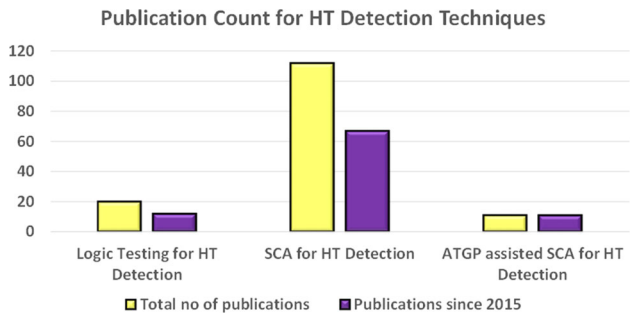

Fig. 11 Publication count trend for contemporary research on HT detection

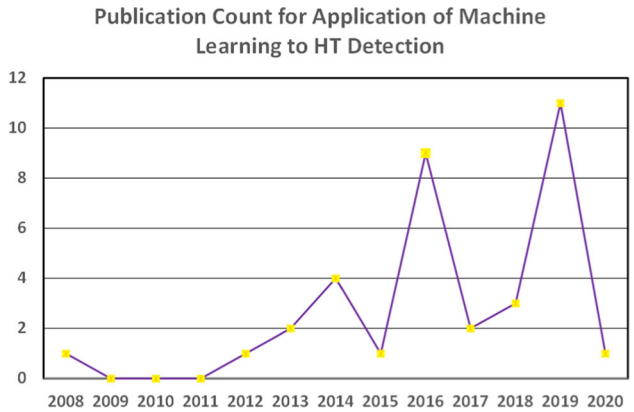

Fig. 12 Publication count trend for contemporary research on machine learning-based techniques for HT detection

on detection accuracy, which have traditionally proved to be major hindrance for these techniques. Another important motivation in these works is to remove the requirement of a golden IC sample, or its accurate simulation model. We envisage that these two will continue to be the primary focus of research, catalysed by the application of advanced machine learning in this problem domain. A parallel focus of current research (albeit of lesser intensity) is to generate effective test patterns for quicker and prominent sensitivity of ICs for detection of HTs.

10 Conclusions

We have provided a comprehensive overview of the state of the art of testing techniques for Hardware Trojan detection, including side-channel analysis-based techniques, logic testing-based techniques, as well as test pattern generation techniques that aid in side-channel analysis and design techniques to enhance circuit testability. We also provided an overview of image processing-based Hardware Trojan detection. Further, a detailed discussion is provided on current state of ML-based techniques in this domain. We have inferred that the current status of this field is far from mature, and given

the nature of the problem, it may be expected to remain an exciting field of research for many years to come, with new challenges appearing regularly. Machine learning, artificial vadjust

intelligence and novel side-channel techniques hopefully will provide the most effective tools against HTs and the intelligent adversaries designing and deploying them.

A overview of HT detection approaches discussed in the paper

Paper	Detection technique	Golden reference model	Methodology	Adversary model	Years
Agrawal et al. [4]	Power-based SCA	Reverse Engineering	PCA of dynamic power side-channel signatures	Transistor Trojan	2007
Wang et al. [112]	Power-based SCA	Simulation	Multi-supply transient-current integration methodology	Transistor Trojan	2008
Banga et al. [13–15]	Power-based SCA	Simulation	Region-aware test pattern generation for increasing transient power	RTL Trojan	2008, 2009
Rad et al. [89,90]	Power-based SCA	Simulation	Region-based transient power signal analysis for Outlier detection	Transistor Trojan	2008
Li et al. [63]	Delay-based SCA	Reverse Engineering	Measurement of selected register-to-register path delays	RTL Trojan	2008
Jin et al. [127]	Delay-based SCA	Reverse Engineering	Fingerprinting using path delay information of the entire chip	RTL Trojan	2008
Jha et al. [62]	Logic Testing	N.A.	Unique probabilistic signature of IC is generated and compared with a HT-free circuit	Transistor Trojan	2008
Rai et al. [92]	Delay-based SCA	Simulation	Statistical analysis of delay signature to enhance HT detection	RTL Trojan	2009
Chakraborty et al. [30]	Logic Testing	N.A.	Heuristics to improve HT detection coverage motivated by the N-detect test	RTL Trojan	2009
Aarestad et al. [1]	Power-based SCA	Simulation	Static current consumption analysis	Transistor Trojan	2010
Narasimhan et al. [77]	Power and Delay-based SCA	Simulation	Use of the correlation between F_{max} and I_{DDT} of an IC to eliminate the impact of PV	Transistor Trojan	2010
Banga et al. in [17]	Logic Testing	N.A.	HT detection scheme enhanced by SAT solver	RTL Trojan	2010
Koushanfar et al. [64]	Multi Parameter	Simulation	Use of a combination of several side-channel parameters	RTL Trojan	2011
Narasimhan et al. [78]	Power-based SCA	Self-Referencing	Compares an IC instance's transient current signature at different time instances	RTL Trojan	2011
Zhang [132]	Logic Testing	N.A.	Formal verification and code coverage analysis of redundant and unused parts of the design are identified followed by the ATPG	RTL Trojan	2011
Li et al. [70]	Delay-based SCA	Self-Referencing	Predicted delay of a path is compared against on-chip delay fingerprint	RTL Trojan	2012

Paper	Detection technique	Golden reference model	Methodology	Adversary model	Years
Lamech et al. [66]	Delay-based SCA	Simulation	Use of delay chain to validate delay measurement	Transistor Trojan	2012
Zhang et al. [129]	Power-based SCA	Self-Referencing	Detection of outliers by comparing measured signature with other measured signatures	Transistor Trojan	2012
Cha et al. [27]	Delay-based SCA	Simulation	Selection of paths having the smallest delays as integer linear programming problem	Transistor Trojan	2013
Forte and Bao et al. [18,45]	Temperature-based SCA	Simulation	Comparison of thermal model of IC at three phases: design, test, and run-time	RTL Trojan	2013, 2015
Hu and Nomroz et al. [56,85]	Temperature-based SCA	Simulation	Multimodal characterization using thermal maps and power maps to detect and locate HT	RTL Trojan	2013, 2014
Hou et al. [55]	Power-based SCA	Simulation	Intrinsic relationship between transient current and static current to reduce PV	RTL Trojan	2014
Yoshimizu et al. [128]	Delay-based SCA	Self-Referencing	Use of symmetry in different transistor-level paths	RTL Trojan	2014
Soll et al. [106]	EM-based SCA	Simulation	EM emission of HT-infected FPGAs is compared with the golden model	RTL Trojan	2014
Liu et al. [71]	Power-based SCA	Self-Referencing	Use of on-chip process control monitors to capture PV	RTL Trojan	2014
Wilcox et al. [117]	Power-based SCA	Simulation	Chip-averaging method targeted for removing intra-die variations	Transistor Trojan	2015
Ismari et al. [59]	Delay-based SCA	Simulation	On-chip embedded test structure to reduce effects of intra-die PV	Transistor Trojan	2015
Ngo et al. [79]	Delay-based SCA	Simulation	Use of clock glitches to measure path delay fingerprints	Transistor Trojan	2015
Exurville et al. [42]	Delay-based SCA	Simulation	Clock glitches combined with statistical techniques to reduce PV	Transistor Trojan	2015
Ngo et al. [79,81]	EM-based SCA	Simulation	Golden AES encryption hardware execution EM traces and HT-infected AES hardware execution traces with the same plaintext are compared	Transistor Trojan	2015, 2016
Balasch et al. [12]	EM-based SCA	Simulation	Use of Welch's t test to determine golden fingerprint	RTL Trojan	2015
Saha et al. [99]	Logic Testing	N.A.	[30] extended using genetic algorithms and SAT solvers	RTL Trojan	2015
Jap et al. [61]	EM-based SCA	Machine Learning	Support Vector Machine (SVM)-based EM side-channel profiling	Transistor Trojan	2016
Xue et al. [125]	Power-based SCA	Machine Learning	HT detection problem is expressed as a two-class machine learning classification problem	Transistor Trojan	2016

Paper	Detection technique	Golden reference model	Methodology	Adversary model	Years
Zheng et al. [133]	Power-based SCA	Self-Referencing	Analyze dynamic current values in self-similar structures of IC and compared	Transistor Trojan	2016
Huang et al. [57,58]	ATPG with SCA	N.A.	Hamming-distance-based reordering and simulation-based reordering of test vectors to improve sensitivity to SCA	Transistor Trojan	2016, 2018
Esirci et al. [41]	Delay-based SCA	Simulation	Statistical analysis of delays in correlated paths	RTL Trojan	2017
He et al. [52]	EM-based SCA	Simulation	Simulation data from RTL design is used to generate the EM fingerprint	RTL Trojan	2017
Sree et al. [93,95]	ATPG with SCA	Self-Referencing	Circuit design is divided into segments and the set of input patterns are applied to extract the power signature	RTL Trojan	2017
Amelian et al. [8]	Delay-based SCA	Simulation	Comparison of delay of K shortest paths with golden IC	RTL Trojan	2018
Cui et al. [38]	Delay-based SCA	Simulation	Use of the order of path delay in path pairs	RTL Trojan	2018
Fournaris et al. [46]	Multi Parameter	Simulation	Use of combination of both logic testing and side-channel testing at different stage of operation	RTL Trojan	2018
Xue et al. [123,124]	Power-based SCA	Machine Learning	Co-training-based HT detection technique similar to [125]	RTL Trojan	2018, 2019
Xue and Ren [122]	Power-based SCA	Self-Referencing	Modeling of static power consumption by clock tree and gates for fingerprinting	RTL Trojan	2018
Cruz et al. [37]	Logic Testing	N.A.	IC is partitioned based on an inserted scan chain and model checking technique is combined with ATPG	RTL Trojan	2018
Chen et al. [31]	EM-based SCA	Simulation	Detection by analyzing the EM radiation of clock trees in an IC	RTL Trojan	2019
Nguyen et al. [82]	Backscattering SCA	Simulation	Sinusoidal EM signal is transmitted and the backscattered signal is received is compared to detect HT	RTL Trojan	2019
Lyu et al. [74]	ATPG with SCA	N.A.	GA-based test generation algorithm to increase HT detection sensitivity	RTL Trojan	2019
Lyu et al. [75]	ATPG with SCA	N.A.	SAT-based ATPG technique to assist HT detection by delay-based SCA	RTL Trojan	2020
Nigh et al. [83]	ATPG with SCA	N.A.	Trojan detection sensitivity is increased from the maximum S-RPD intra-die PV	RTL Trojan	2020

References

1. Aarestad, J., Acharyya, D., Rad, R., Plusquellic, J.: Detecting Trojans through leakage current analysis using multiple supply pad IDDQS. Trans. Inf. Forensic Secur. **5**(4), 893–904 (2010). https://doi.org/10.1109/TIFS.2010.2061228
2. Abramovici, M., Bradley, P.: Integrated circuit security: new threats and solutions. In: Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies, pp. 1–3 (2009)
3. Adibelli, S., Juyal, P., Nguyen, L.N., Prvulovic, M., Zajic, A.: Near field backscattering based sensing for hardware Trojan detection. IEEE Trans. Antennas Propag. **68**(12), 8082–8090 (2020)
4. Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., Sunar, B.: Trojan detection using IC fingerprinting. In: 2007 IEEE Symposium on Security and Privacy (SP '07), pp. 296–310 (2007)
5. Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P., Sunar, B.: Trojan detection using IC fingerprinting. In: 2007 IEEE Symposium on Security and Privacy (SP'07), pp. 296–310. IEEE (2007)
6. Alaql, A., Bhunia, S.: Scalable attack-resistant obfuscation of logic circuits. arXiv preprint arXiv:2010.15329 (2020)
7. Alkabani, Y., Koushanfar, F.: Consistency-based characterization for IC Trojan detection. In: 2009 IEEE/ACM International Conference on Computer-Aided Design—Digest of Technical Papers, pp. 123–127 (2009)
8. Amelian, A., Borujeni, S.E.: A side-channel analysis for hardware Trojan detection based on path delay measurement. J. Circuits Syst. Comput. **27**(09), 1850138 (2018). https://doi.org/10.1142/S0218126618501384
9. Amir, S., Shakya, B., Xu, X., Jin, Y., Bhunia, S., Tehranipoor, M., Forte, D.: Development and evaluation of hardware obfuscation benchmarks. J. Hardw. Syst. Secur. **2**(2), 142–161 (2018)
10. Amyeen, M.E., Venkataraman, S., Ojha, A., Lee, S.: Evaluation of the quality of N-detect scan ATPG patterns on a processor. In: 2004 International Conference on Test, pp. 669–678. IEEE (2004)
11. Ba, P.S., Dupuis, S., Flottes, M.L., Di Natale, G., Rouzeyre, B.: Using outliers to detect stealthy hardware Trojan triggering? In: 2016 1st IEEE International Verification and Security Workshop (IVSW), pp. 1–6. IEEE (2016)
12. Balasch, J., Gierlichs, B., Verbauwhede, I.: Electromagnetic circuit fingerprints for hardware Trojan detection. In: 2015 IEEE International Symposium on Electromagnetic Compatibility (EMC), pp. 246–251 (2015)
13. Banga, M., Hsiao, M.S.: A region based approach for the identification of hardware Trojans. In: 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 40–47 (2008)
14. Banga, M., Hsiao, M.S.: A novel sustained vector technique for the detection of hardware Trojans. In: 2009 22nd International Conference on VLSI Design, pp. 327–332 (2009)
15. Banga, M., Chandrasekar, M., Fang, L., Hsiao, M.S.: Guided test generation for isolation and detection of embedded Trojans in ICs. In: Proceedings of the 18th ACM Great Lakes Symposium on VLSI, pp. 363–366. GLSVLSI '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/13661101366196
16. Banga, M., Chandrasekar, M., Fang, L., Hsiao, M.S.: Guided test generation for isolation and detection of embedded Trojans in ICs. In: Proceedings of the 18th ACM Great Lakes Symposium on VLSI, pp. 363–366. ACM (2008)
17. Banga, M., Hsiao, M.S.: Trusted RTL: Trojan detection methodology in pre-silicon designs. In: 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 56–59. IEEE (2010)
18. Bao, C., Forte, D., Srivastava, A.: Temperature tracking: toward robust run-time detection of hardware Trojans. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **34**(10), 1577–1585 (2015)
19. Bao, C., Forte, D., Srivastava, A.: On application of one-class SVM to reverse engineering-based hardware Trojan detection. In: Fifteenth International Symposium on Quality Electronic Design, pp. 47–54 (2014). https://doi.org/10.1109/ISQED.2014.6783305
20. Bao, C., Forte, D., Srivastava, A.: On reverse engineering-based hardware Trojan detection. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **35**(1), 49–57 (2016). https://doi.org/10.1109/TCAD.2015.2488495
21. Bhunia, S., Abramovici, M., Agrawal, D., Bradley, P., Hsiao, M.S., Plusquellic, J., Tehranipoor, M.: Protection against hardware Trojan attacks: towards a comprehensive solution. IEEE Des. Test **30**(3), 6–17 (2013)
22. Bhunia, S., Hsiao, M.S., Banga, M., Narasimhan, S.: Hardware Trojan attacks: threat, analysis and countermeasures. Proc. IEEE **102**(8), 1229–1247 (2014)
23. Bhunia, S., Tehranipoor, M.M.: The Hardware Trojan War: Attacks, Myths, and Defenses, 1st edn. Springer, Berlin (2017)
24. Bloom, G., Narahari, B., Simha, R.: OS support for detecting Trojan circuit attacks. In: 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 100–103. IEEE (2009)
25. Bloom, G., Narahari, B., Simha, R., Zambreno, J.: Providing Secure execution environments with a last line of defense against Trojan circuit attacks. Comput. Secur. **28**(7), 660–669 (2009)
26. Bright, P.: Meltdown and spectre: here's what Intel, Apple, Microsoft, others are doing about it. Ars Technica. 5 January (2018)
27. Cha, B., Gupta, S.K.: Trojan detection via delay measurements: a new approach to select paths and vectors to maximize effectiveness and minimize cost. In: 2013 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 1265–1270 (2013)
28. Chakraborty, R.S., Narasimhan, S., Bhunia, S.: Hardware Trojan: threats and emerging solutions. In: 2009 IEEE International high level design validation and test workshop, pp. 166–171. IEEE (2009)
29. Chakraborty, R.S., Bhunia, S.: Security against hardware Trojan through a novel application of design obfuscation. In: 2009 IEEE/ACM International Conference on Computer-Aided Design-Digest of Technical Papers, pp. 113–116. IEEE (2009)
30. Chakraborty, R.S., Wolff, F., Paul, S., Papachristou, C., Bhunia, S.: MERO: a statistical approach for hardware Trojan detection. In: International Workshop on Cryptographic Hardware and Embedded Systems, pp. 396–410. Springer (2009)
31. Chen, Z., Guo, S., Wang, J., Li, Y., Lu, Z.: Toward FPGA security in IoT: a new detection technique for hardware Trojans. IEEE Internet Things J. **6**(4), 7061–7068 (2019)
32. Courbon, F., Loubet-Moundi, P., Fournier, J.J., Tria, A.: A High efficiency hardware Trojan detection technique based on fast SEM imaging. In: 2015 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 788–793 (2015). https://doi.org/10.7873/DATE.2015.1104
33. Courbon, F., Loubet-Moundi, P., Fournier, J.J., Tria, A.: SEMBA: a SEM based acquisition technique for fast invasive hardware Trojan detection. In: 2015 European Conference on Circuit Theory and Design (ECCTD), pp. 1–4 (2015). https://doi.org/10.1109/ECCTD.2015.7300097
34. Cozzi, M., Galliere, J.M., Maurine, P.: Exploiting phase information in thermal scans for stealthy Trojan detection. In: 2018 21st Euromicro Conference on Digital System Design (DSD), pp. 573–576 (2018). https://doi.org/10.1109/DSD.2018.00100
35. Cozzi, M., Galliere, J.M., Maurine, P.: Thermal scans for detecting hardware Trojans. In: Fan, J., Gierlichs, B. (eds.) Constructive Side-Channel Analysis and Secure Design, pp. 117–132. Springer International Publishing, Cham (2018)

36. Cozzi, M., Galliere, J.M., Maurine, P.: Statistical lock-in thermography to improve contrast and detectivity of ICs thermal maps. In: 2019 25th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC), pp. 1–6 (2019). https://doi.org/10.1109/THERMINIC.2019.8923769
37. Cruz, J., Farahmandi, F., Ahmed, A., Mishra, P.: Hardware Trojan detection using ATPG and model checking. In: 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), pp. 91–96. IEEE (2018)
38. Cui, X., Koopahi, E., Wu, K., Karri, R.: Hardware Trojan detection using the order of path delay. J. Emerg. Technol. Comput. Syst. **14**(3), 1–23 (2018). https://doi.org/10.1145/3229050
39. Dupuis, S., Ba, P.S., Flottes, M.L., Di Natale, G., Rouzeyre, B.: New testing procedure for finding insertion sites of stealthy hardware Trojans. In: 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 776–781. IEEE (2015)
40. Dupuis, S., Flottes, M.L., Di Natale, G., Rouzeyre, B.: Protection against hardware Trojans with logic testing: proposed solutions and challenges ahead. IEEE Des. Test **35**(2), 73–90 (2017)
41. Esirci, F.N., Bayrakci, A.A.: Hardware Trojan detection based on correlated path delays in defiance of variations with spatial correlations. In: Design, Automation Test in Europe Conference Exhibition (DATE), 2017, pp. 163–168 (2017)
42. Exurville, I., Zussa, L., Rigaud, J., Robisson, B.: Resilient hardware Trojans detection based on path delay measurements. In: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 151–156 (2015)
43. Faezi, S., Yasaei, R., Barua, A., Faruque, M.A.A.: Brain-inspired golden chip free hardware Trojan detection. IEEE Trans. Inf. Forensics Secur. **16**, 2697–2708 (2021). https://doi.org/10.1109/TIFS.2021.3062989
44. Ferraiuolo, A., Zhang, X., Tehranipoor, M.: Experimental analysis of a ring oscillator network for hardware Trojan detection in a 90 nm ASIC. In: 2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 37–42 (2012)
45. Forte, D., Bao, C., Srivastava, A.: Temperature tracking: an innovative run-time approach for hardware Trojan detection. In: 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 532–539 (2013)
46. Fournaris, A.P., Pyrgas, L., Kitsos, P.: An FPGA hardware Trojan detection approach based on multiple parameter analysis. In: 2018 21st Euromicro Conference on Digital System Design (DSD), pp. 516–522 (2018)
47. Guin, U., Huang, K., DiMase, D., Carulli, J.M., Tehranipoor, M., Makris, Y.: Counterfeit integrated circuits: a rising threat in the global semiconductor supply chain. Proc. IEEE **102**(8), 1207–1228 (2014)
48. Haider, S.K., Jin, C., Ahmad, M., Shila, D.M., Khan, O., van Dijk, M.: Advancing the state-of-the-art in hardware Trojans detection. IEEE Trans. Dependable Secur. Comput. **16**(1), 18–32 (2017)
49. Haider, S.K., Jin, C., van Dijk, M.: Advancing the state-of-the-art in hardware Trojans design. In: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 823–826. IEEE (2017)
50. Harada, L.L.: Semiconductor technology and U.S. National Security. Technical report, ARMY WAR COLL CARLISLE BARRACKS PA (2010)
51. Hasegawa, K., Shi, Y., Togawa, N.: Hardware Trojan detection utilizing machine learning approaches. In: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science And Engineering (TrustCom/BigDataSE), pp. 1891–1896 (2018)
52. He, J., Zhao, Y., Guo, X., Jin, Y.: Hardware Trojan detection through chip-free electromagnetic side-channel statistical analysis. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **25**(10), 2939–2948 (2017)
53. Hicks, M., Finnicum, M., King, S.T., Martin, M.M., Smith, J.M.: Overcoming an untrusted computing base: detecting and removing malicious hardware automatically. In: 2010 IEEE Symposium on Security and Privacy, pp. 159–172. IEEE (2010)
54. Schneider, W., Chairman, F.: Defense Science Board Task Force on High Performance Microchip Supply. Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, Washington (2005)
55. Hou, B., He, C., Wang, L., En, Y., Xie, S.: Hardware Trojan detection via current measurement: a method immune to process variation effects. In: 2014 10th International Conference on Reliability, Maintainability and Safety (ICRMS), pp. 1039–1042 (2014)
56. Hu, K., Nowroz, A.N., Reda, S., Koushanfar, F.: High-sensitivity hardware Trojan detection using multimodal characterization. In: 2013 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 1271–1276 (2013)
57. Huang, Y., Bhunia, S., Mishra, P.: Scalable test generation for Trojan detection using side channel analysis. IEEE Trans. Inf. Forensics Secur. **13**(11), 2746–2760 (2018)
58. Huang, Y., Bhunia, S., Mishra, P.: MERS: statistical test generation for side-channel analysis based Trojan detection. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 130–141. CCS '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2976749.2978396
59. Ismari, D., Plusquellic, J., Lamech, C., Bhunia, S., Saqib, F.: On detecting delay anomalies introduced by hardware Trojans. In: 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1–7 (2016)
60. Jacob, N., Merli, D., Heyszl, J., Sigl, G.: Hardware Trojans: current challenges and approaches. IET Comput. Digit. Tech. **8**(6), 264–273 (2014)
61. Jap, D., Wei He, Bhasin, S.: Supervised and unsupervised machine learning for side-channel based Trojan detection. In: 2016 IEEE 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP), pp. 17–24 (2016)
62. Jha, S., Jha, S.K.: Randomization based probabilistic approach to detect Trojan circuits. In: 2008 11th IEEE High Assurance Systems Engineering Symposium, pp. 117–124. IEEE (2008)
63. Jie Li, Lach, J.: At-speed delay characterization for IC authentication and Trojan horse detection. In: 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 8–14 (2008)
64. Koushanfar, F., Mirhoseini, A.: A unified framework for multimodal submodular integrated circuits Trojan detection. IEEE Trans. Inf. Forensics Secur. **6**(1), 162–174 (2011)
65. Krieg, C., Rathmair, M., Schupfer, F.: A process for the detection of design-level hardware Trojans using verification methods. In: 2014 IEEE International Conference on High Performance Computing and Communications, 2014 IEEE 6th International Symposium on Cyberspace Safety and Security, 2014 IEEE 11th International Conference on Embedded Software and System (HPCC, CSS, ICESS), pp. 729–734. IEEE (2014)
66. Lamech, C., Plusquellic, J.: Trojan detection based on delay variations measured using a high-precision, low-overhead embedded test structure. In: 2012 IEEE International Symposium on Hardware-Oriented Security and Trust, pp. 75–82 (2012)
67. Lecomte, M., Fournier, J., Maurine, P.: An on-chip technique to detect hardware Trojans and assist counterfeit identification. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **25**(12), 3317–3330 (2017)
68. Li, H., Liu, Q.: Hardware Trojan detection acceleration based on word-level statistical properties management. In: 2014 Inter-

national Conference on Field-Programmable Technology (FPT), pp. 153–160. IEEE (2014)
69. Li, H., Liu, Q., Zhang, J.: A survey of hardware Trojan threat and defense. Integration **55**, 426–437 (2016)
70. Li, M., Davoodi, A., Tehranipoor, M.: A sensor-assisted self-authentication framework for hardware Trojan detection. In: 2012 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 1331–1336 (2012)
71. Liu, Y., Huang, K., Makris, Y.: Hardware Trojan detection through golden chip-free statistical side-channel fingerprinting. In: 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6 (2014)
72. Lodhi, F.K., Abbasi, I., Khalid, F., Hasan, O., Awwad, F., Hasan, S.R.: A self-learning framework to detect the intruded integrated circuits. In: 2016 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1702–1705 (2016). https://doi.org/10.1109/ISCAS.2016.7538895
73. Lodhi, F.K., Hasan, S.R., Hasan, O., Awwadl, F.: Power profiling of microcontroller's instruction set for runtime hardware Trojans detection without golden circuit models. In: Design, Automation Test in Europe Conference Exhibition (DATE), 2017, pp. 294–297 (2017). https://doi.org/10.23919/DATE.2017.7927002
74. Lyu, Y., Mishra, P.: Efficient test generation for Trojan detection using side channel analysis. In: 2019 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 408–413 (2019)
75. Lyu, Y., Mishra, P.: Automated test generation for Trojan detection using delay-based side channel analysis. In: 2020 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 1031–1036 (2020)
76. Mishra, P., Bhunia, S., Tehranipoor, M.: Hardware IP Security and Trust. Springer, Berlin (2017)
77. Narasimhan, S., Du, D., Chakraborty, R.S., Paul, S., Wolff1, F., Papachristou, C., Roy, K., Bhunia, S.: Multiple-parameter side-channel analysis: a non-invasive hardware Trojan detection approach. In: 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 13–18 (2010)
78. Narasimhan, S., Wang, X., Du, D., Chakraborty, R.S., Bhunia, S.: TeSR: a robust temporal self-referencing approach for hardware Trojan detection. In: 2011 IEEE International Symposium on Hardware-Oriented Security and Trust, pp. 71–74 (2011)
79. Ngo, X., Exurville, I., Bhasin, S., Danger, J., Guilley, S., Najm, Z., Rigaud, J., Robisson, B.: Hardware Trojan detection by delay and electromagnetic measurements. In: 2015 Design, Automation Test in Europe Conference Exhibition (DATE), pp. 782–787 (2015)
80. Ngo, X.T., Bhasin, S., Danger, J.L., Guilley, S., Najm, Z.: Linear complementary dual code improvement to strengthen encoded circuit against hardware Trojan horses. In: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 82–87. IEEE (2015)
81. Ngo, X.T., Najm, Z., Bhasin, S., Guilley, S., Danger, J.L.: Method taking into account process dispersion to detect hardware Trojan horse by side-channel analysis. J. Cryptogr. Eng. **6**(3), 239–247 (2016). https://doi.org/10.1007/s13389-016-0129-2
82. Nguyen, L.N., Cheng, C., Prvulovic, M., Zajić, A.: Creating a backscattering side channel to enable detection of dormant hardware Trojans. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **27**(7), 1561–1574 (2019)
83. Nigh, C., Orailoglu, A.: Test pattern superposition to detect hardware Trojans. In: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 25–30. IEEE (2020)
84. Nigh, C., Orailoglu, A.: AdaTrust: combinational hardware Trojan detection through adaptive test pattern construction. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **29**(3), 544–557 (2021). https://doi.org/10.1109/TVLSI.2021.3053553
85. Nowroz, A.N., Hu, K., Koushanfar, F., Reda, S.: Novel techniques for high-sensitivity hardware Trojan detection using thermal and power maps. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **33**(12), 1792–1805 (2014)
86. Pan, Z., Mishra, P.: Automated test generation for hardware Trojan detection using reinforcement learning. In: 2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 408–413 (2021)
87. Potkonjak, M., Nahapetian, A., Nelson, M., Massey, T.: Hardware Trojan horse detection using gate-level characterization. In: 2009 46th ACM/IEEE Design Automation Conference, pp. 688–693 (2009)
88. Quadir, S.E., Chen, J., Forte, D., Asadizanjani, N., Shahbazmohamadi, S., Wang, L., Chandy, J., Tehranipoor, M.: A survey on chip to system reverse engineering. J. Emerg. Technol. Comput. Syst. **13**(1), 1–34 (2016). https://doi.org/10.1145/2755563
89. Rad, R., Plusquellic, J., Tehranipoor, M.: Sensitivity analysis to hardware Trojans using power supply transient signals. In: 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 3–7 (2008)
90. Rad, R.M., Wang, X., Tehranipoor, M., Plusquellic, J.: Power supply signal calibration techniques for improving detection resolution to hardware Trojans. In: 2008 IEEE/ACM International Conference on Computer-Aided Design, pp. 632–639 (2008)
91. Rad, R.M., Wang, X., Tehranipoor, M., Plusquellic, J.: Power Supply signal calibration techniques for improving detection resolution to hardware Trojans. In: 2008 IEEE/ACM International Conference on Computer-Aided Design, pp. 632–639. IEEE (2008)
92. Rai, D., Lach, J.: Performance of delay-based Trojan detection techniques under parameter variations. In: 2009 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 58–65 (2009)
93. Ranjani, R.S., Devi, M.N.: Golden-chip free power metric based hardware Trojan detection and diagnosis. Far East J. Electron. Commun. **17**, 517–530 (2017)
94. Ranjani, R.S., Devi, M.N.: Malicious hardware detection and design for trust: an analysis. Elektrotehniski Vestnik **84**(1/2), 7 (2017)
95. Ranjani, R.S., Maneesh, P., Devi, M.N.: Golden chip free HT detection and diagnosis using power signature analysis. Presented at the 7th IEEE International Workshop on Reliability Aware System Design and Test (RASDAT) (2016)
96. Rathmair, M., Schupfer, F., Krieg, C.: Applied formal methods for hardware Trojan detection. In: 2014 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 169–172. IEEE (2014)
97. Rilling, J., Graziano, D., Hitchcock, J., Meyer, T., Wang, X., Jones, P., Zambreno, J.: Circumventing a ring oscillator approach to FPGA-based hardware Trojan detection. In: 2011 IEEE 29th International Conference on Computer Design (ICCD), pp. 289–292 (2011)
98. Sabri, M., Shabani, A., Alizadeh, B.: SAT-based integrated hardware Trojan detection and localization approach through path-delay analysis. IEEE Trans. Circuits Syst. II Express Briefs **68**, 2850–2854 (2021). https://doi.org/10.1109/TCSII.2021.3074549
99. Saha, S., Chakraborty, R.S., Nuthakki, S.S., Mukhopadhyay, D., et al.: Improved test pattern generation for hardware Trojan detection using genetic algorithm and Boolean satisfiability. In: International Workshop on Cryptographic Hardware and Embedded Systems, pp. 577–596. Springer (2015)
100. Salmani, H., Tehranipoor, M., Karri, R.: On design vulnerability analysis and trust benchmarks development. In: 2013 IEEE 31st International Conference on Computer Design (ICCD), pp. 471–474. IEEE (2013)
101. Saran, T., Ranjani, R.S., Devi, M.N.: A region based fingerprinting for hardware Trojan detection and diagnosis. In: 2017 4th Interna-

tional Conference on Signal Processing and Integrated Networks (SPIN), pp. 166–172. IEEE (2017)
102. Shakya, B., He, T., Salmani, H., Forte, D., Bhunia, S., Tehranipoor, M.: Benchmarking of hardware Trojans and maliciously affected circuits. J. Hardw. Syst. Secur. **1**(1), 85–102 (2017)
103. Shi, Q., Vashistha, N., Lu, H., Shen, H., Tehranipoor, B., Woodard, D.L., Asadizanjani, N.: Golden gates: a new hybrid approach for rapid hardware Trojan detection using testing and imaging. In: 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 61–71 (2019). https://doi.org/10.1109/HST.2019.8741031
104. Stern, A., Mehta, D., Tajik, S., Farahmandi, F., Tehranipoor, M.: SPARTA: a laser probing approach for Trojan detection. In: 2020 IEEE International Test Conference (ITC), pp. 1–10 (2020). https://doi.org/10.1109/ITC44778.2020.9325222
105. Stern, A., Mehta, D., Tajik, S., Guin, U., Farahmandi, F., Tehranipoor, M.: SPARTA-COTS: a laser probing approach for sequential Trojan detection in COTS integrated circuits. In: 2020 IEEE Physical Assurance and Inspection of Electronics (PAINE), pp. 1–6 (2020). https://doi.org/10.1109/PAINE49178.2020.9337728
106. Söll, O., Korak, T., Muehlberghuber, M., Hutter, M.: EM-based detection of hardware Trojans on FPGAs. In: 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 84–87 (2014)
107. Tehranipoor, M., Koushanfar, F.: A survey of hardware Trojan taxonomy and detection. IEEE Des. Test Comput. **27**(1), 10–25 (2010)
108. Trust-hub.org. https://www.trust-hub.org/benchmarks/chip-level-Trojan
109. Vashistha, N., Rahman, M.T., Shen, H., Woodard, D.L., Asadizanjani, N., Tehranipoor, M.: Detecting hardware Trojans inserted by untrusted foundry using physical inspection and advanced image processing. J. Hardw. Syst. Secur. **2**(4), 333–344 (2018). https://doi.org/10.1007/s41635-018-0055-0
110. Voyiatzis, A.G., Stefanidis, K.G., Kitsos, P.: Efficient triggering of Trojan hardware logic. In: 2016 IEEE 19th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), pp. 1–6. IEEE (2016)
111. Waksman, A., Suozzo, M., Sethumadhavan, S.: FANCI: identification of stealthy malicious logic using Boolean functional analysis. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, pp. 697–708 (2013)
112. Wang, X., Salmani, H., Tehranipoor, M., Plusquellic, J.: Hardware Trojan detection and isolation using current integration and localized current analysis. In: 2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems, pp. 87–95 (2008)
113. Wei, S., Potkonjak, M.: Scalable consistency-based hardware Trojan detection and diagnosis. In: 2011 5th International Conference on Network and System Security, pp. 176–183 (2011)
114. Wei, S., Potkonjak, M.: Scalable hardware Trojan diagnosis. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **20**(6), 1049–1057 (2012)
115. Wei, S., Potkonjak, M.: Self-consistency and consistency-based detection and diagnosis of malicious circuitry. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **22**(9), 1845–1853 (2014)
116. Wen, Y., Yu, W.: Combining thermal maps with inception neural networks for hardware Trojan detection. IEEE Embed. Syst. Lett. **13**(2), 45–48 (2021). https://doi.org/10.1109/LES.2020.3000008
117. Wilcox, I., Saqib, F., Plusquellic, J.: GDS-II Trojan detection using multiple supply pad VDD and GND IDDQs in ASIC functional units. In: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp. 144–150 (2015)
118. Wolff, F., Papachristou, C., Bhunia, S., Chakraborty, R.S.: Towards Trojan-free trusted ICs: problem analysis and detection scheme. In: 2008 Design, Automation and Test in Europe, pp. 1362–1365. IEEE (2008)
119. Wolff, F., Papachristou, C., Bhunia, S., Chakraborty, R.S.: Towards Trojan-free trusted ICs: problem analysis and detection scheme. In: Proceedings of the Conference on Design, Automation and Test in Europe, pp. 1362–1365. ACM (2008)
120. Xiao, K., Forte, D., Jin, Y., Karri, R., Bhunia, S., Tehranipoor, M.: Hardware Trojans: lessons learned after one decade of research. ACM Trans. Des. Autom. Electron. Syst. **22**(1), 1–23 (2016). https://doi.org/10.1145/2906147
121. Xiao, K., Forte, D., Jin, Y., Karri, R., Bhunia, S., Tehranipoor, M.: Hardware Trojans: lessons learned after one decade of research. ACM Trans. Des. Autom. Electron. Syst. (TODAES) **22**(1), 1–23 (2016)
122. Xue, H., Ren, S.: Self-reference-based hardware Trojan detection. IEEE Trans. Semicond. Manuf. **31**(1), 2–11 (2018)
123. Xue, M., Bian, R., Wang, J., Liu, W.: A co-training based hardware Trojan detection technique by exploiting unlabeled ICs and inaccurate simulation models. In: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science And Engineering (TrustCom/BigDataSE), pp. 1452–1457 (2018)
124. Xue, M., Bian, R., Wang, J., Liu, W.: Building an accurate hardware Trojan detection technique from inaccurate simulation models and unlabelled ICs. IET Comput. Dig. Tech. **13**(4), 348–359 (2019)
125. Xue, M., Wang, J., Hu, A.: An enhanced classification-based golden chips-free hardware Trojan detection technique. In: 2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST), pp. 1–6 (2016)
126. Yang, L., Li, X., Li, H.: Hardware Trojan detection method based on time feature of chip temperature. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1029–1032 (2020). https://doi.org/10.1109/CCWC47524.2020.9031281
127. Yier Jin, Makris, Y.: Hardware Trojan detection using path delay fingerprint. In: 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 51–57 (2008)
128. Yoshimizu, N.: Hardware Trojan detection by symmetry breaking in path delays. In: 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 107–111 (2014)
129. Zhang, J., Yu, H., Xu, Q.: HTOutlier: hardware Trojan detection with side-channel signature outlier identification. In: 2012 IEEE International Symposium on Hardware-Oriented Security and Trust, pp. 55–58 (2012)
130. Zhang, X., Tehranipoor, M.: RON: an on-chip ring oscillator network for hardware Trojan detection. In: 2011 Design, Automation Test in Europe, pp. 1–6 (2011)
131. Zhang, X., Salmani, H.: Integrated circuit authentication: hardware Trojans and counterfeit detection (2014)
132. Zhang, X., Tehranipoor, M.: Case study: detecting hardware Trojans in third-party digital IP cores. In: 2011 IEEE International Symposium on Hardware-Oriented Security and Trust, pp. 67–70. IEEE (2011)
133. Zheng, Y., Yang, S., Bhunia, S.: SeMIA: self-similarity-based IC integrity analysis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **35**(1), 37–48 (2016)
134. Zhou, B., Adato, R., Zangeneh, M., Yang, T., Uyar, A., Goldberg, B., Unlu, S., Joshi, A.: Detecting hardware Trojans using backside optical imaging of embedded watermarks. In: 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6 (2015). https://doi.org/10.1145/2744769.2744822

135. Zhou, B., Aksoylar, A., Vigil, K., Adato, R., Tan, J., Goldberg, B., Ünlü, M.S., Joshi, A.: Hardware Trojan detection using backside optical imaging. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **40**(1), 24–37 (2021). https://doi.org/10.1109/TCAD.2020.2991680

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.