Hardware Trojans: Lessons Learned after One Decade of Research

K. XIAO, ECE Department, University of Connecticut
D. FORTE, ECE Department, University of Florida
Y. JIN, EECS Department, University of Central Florida
R. KARRI, ECE Department, Polytechnic Institute of New York University
S. BHUNIA and M. TEHRANIPOOR, ECE Department, University of Florida

Given the increasing complexity of modern electronics and the cost of fabrication, entities from around the globe have become more heavily involved in all phases of the electronics supply chain. In this environment, hardware Trojans (i.e., malicious modifications or inclusions made by untrusted third parties) pose major security concerns, especially for those integrated circuits (ICs) and systems used in critical applications and cyber infrastructure. While hardware Trojans have been explored significantly in academia over the last decade, there remains room for improvement. In this article, we examine the research on hardware Trojans from the last decade and attempt to capture the lessons learned. A comprehensive adversarial model taxonomy is introduced and used to examine the current state of the art. Then the past countermeasures and publication trends are categorized based on the adversarial model and topic. Through this analysis, we identify what has been covered and the important problems that are underinvestigated. We also identify the most critical lessons for those new to the field and suggest a roadmap for future hardware Trojan research.

CCS Concepts: • **Hardware** → *Safety critical systems*

Additional Key Words and Phrases: Hardware security and trust, hardware Trojan attacks, countermeasures, attack model

ACM Reference Format:
K. Xiao, D. Forte, Y. Jin, R. Karri, S. Bhunia, and M. Tehranipoor. 2016. Hardware Trojans: Lessons learned after one decade of research. ACM Trans. Des. Autom. Electron. Syst. 22, 1, Article 6 (May 2016), 23 pages.
DOI: http://dx.doi.org/10.1145/2906147

1. INTRODUCTION

With the emergence of information technology and its critical role in our daily lives, the risk of cyber attacks is larger today than ever before. While the battle between software developers and hackers has raged since the 1980s, the underlying hardware was generally considered safe. However, in the last decade or so, the complexity of the design, fabrication, and distribution of electronics has caused a shift throughout the industry toward a global business model, thereby creating new sources of attack. In such a model, untrusted entities participate either directly or indirectly in all phases in the life of an electronic device or integrated circuit (IC). This unprecedented access

This work was supported in part by the National Science Foundation under grant CNS-1558516.
Authors' addresses: K. Xiao, 371 Fairfield Way, Unit 4157, Storrs, CT 06269-4157; email: kan.xiao@uconn.edu; D. Forte, 339E Larsen Hall, P.O. Box 116200, Gainesville, FL 32611-6200; email: dforte@ece.ufl.edu; Y. Jin, HEC 237, University of Central Florida, Orlando, FL 32816; email: jier.jin@eecs.ucf.edu; R. Karri and S. Bhunia, LAR 336A, 336A Larsen Hall Gainesville FL 32611-6200; emails: rkarri@nyu.edu, swarup@ece.ufl.edu; M. Tehranipoor, 325 Benton Hall, P.O. Box 116200, Gainesville, FL 32611-6200; email: tehranipoor@ece.ufl.edu.

DOI: http://dx.doi.org/10.1145/2906147

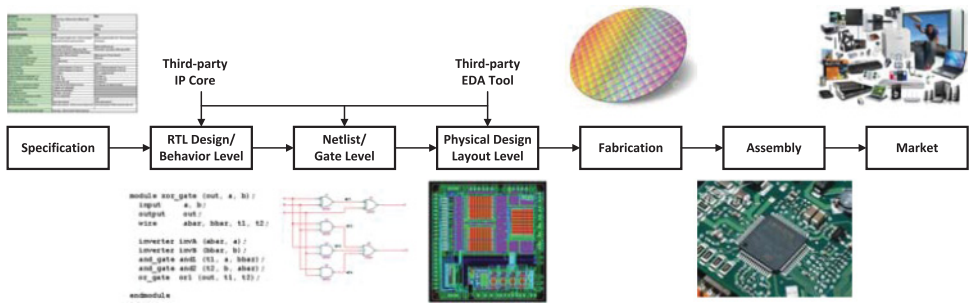


Fig. 1. Modern semiconductor supply chain.

to hardware has been a major cause for concern, resulting in very plausible conspiracy theories. In 2008, Adee [2008] reported that a critical failure in Syrian radar might have been intentionally triggered through a "back door" hidden within a commercial off-the-shelf (COTS) microprocessor. According to a U.S. defense contractor who spoke on condition of anonymity, a "European chip maker" recently built such microprocessors with remote kill switches for just such purposes. Given the dire consequences associated with such weaknesses, the so-called hardware Trojan issue has received considerable attention from academia, industry, and government over the last decade.

1.1. Vulnerability of the Integrated Circuits Supply Chain

With semiconductor scaling to very deep submicron levels, the complexity and cost of IC design and fabrication have increased dramatically. An ASIC/SoC component will typically go through a process as shown in Figure 1. The first step of the process is the translation of the specifications into a behavioral description, typically in a hardware design language (HDL) such as Verilog or VHDL. Next, synthesis is performed to transform the behavioral description into a design implementation in terms of logic gates (i.e., netlist). After implementing the netlist as a layout design, the digital GDSII files are then handed to a foundry for IC fabrication. Once the foundry produces the actual ICs, the testing step ensures their correct operations. Those ICs that pass testing are packaged by assembly, retested, sent to the market, and eventually deployed in systems.

The most advanced semiconductor technology requires prohibitive investment for each stage of the IC development procedure. As an example, the estimated cost of owning a foundry was $5 billion in 2015 [DIGITIMES 2012]. As a result, most semiconductor companies cannot afford maintaining such a long supply chain from design to packaging. In order to lower R&D cost and speed up the development cycle, they typically outsource fabrication to a third-party foundry, purchase third-party intellectual property (IP) cores, and/or use Electronic Design Automation (EDA) tools from third-party vendors. The use of untrusted (and potentially malicious) third parties increases the security concerns. Thus, the supply chain is now considered susceptible to various attacks, such as hardware Trojan insertion, reverse engineering, IP piracy, IC tampering, IC cloning, IC overproduction, and so forth. Among these, hardware Trojans are arguably the biggest concern and have garnered considerable attention.

1.2. Hardware Trojan Threat

A hardware Trojan is defined as a malicious, intentional modification of a circuit design that results in undesired behavior when the circuit is deployed [Tehranipoor and Wang 2012]. ICs that are "infected" by a hardware Trojan may experience changes to their functionality or specification, may leak sensitive information, or may experience

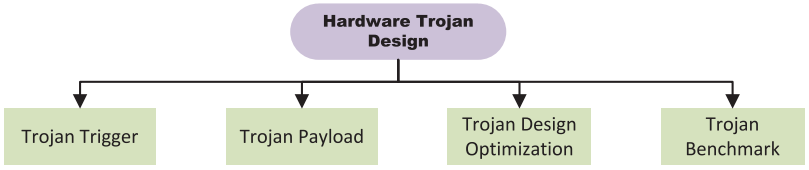


Fig. 2. Hardware Trojan design.

degraded or unreliable performance. Several previous papers have proposed detailed taxonomies to cover the wide range of potential Trojans. For instance, Karri et al. [2010] and Tehranipoor and Wang [2012] separate Trojans based on five different attributes: insertion phase, abstraction level, activation mechanism, effects, and location. Hardware Trojans are designed to be stealthy by intelligent adversaries, which is a major difference from manufacturing defects that have been extensively researched for decades. Manufacturing defects are unintentional and random, and their behavior can be reflected with stuck-at fault, delay fault, and so forth models. For hardware Trojans, it is difficult to create a model that fits all types. Additionally, defects are only produced during the manufacturing process, while hardware Trojans could be inserted at any phase of the IC development. Hence, the hardware Trojan problem is more intricate than the manifestation of manufacturing defects.

Research on hardware Trojans has grown dramatically over the past decade and is expected to continue. In this article, we reflect on the accomplishments and limitations of prior work, highlight the current trends, and discuss future directions for hardware Trojan research. In Section 2, we give a survey of prior Trojan designs and countermeasures and categorize them. Comprehensive attack models based on the current semiconductor supply chain are presented in Section 3. Section 4 discusses our observations based on the publication trends of the research community. The most dangerous hardware Trojans and attack scenarios have yet to be solved, but many new researchers are still focused on the low-hanging fruit. In Section 5, we highlight the unsolved problems that require more attention. Section 6 sets forth a roadmap to investigate more promising approaches as well as deal with new challenges in the field of hardware Trojans. Finally, Section 7 concludes the article.

2. CURRENT STATE OF THE ART

2.1. Hardware Trojan Design

The hardware Trojan domain has seen significant progress since the first paper on hardware Trojans in Agrawal et al. [2007]. To exploit the potential risks of hardware Trojans, various hardware Trojans have been developed. In general, a Trojan contains two basic parts: trigger and payload [Jin and Makris 2008]. A Trojan trigger is an optional part that monitors various signals and/or a series of events in the circuit. The payload usually taps signals from the original (Trojan-free) circuit and the output of the trigger. Once the trigger detects an expected event or condition, the payload is activated to perform malicious behavior. Typically, the trigger is expected to be activated under extremely rare conditions, so the payload remains inactive most of the time. When the payload is inactive, the IC acts like a Trojan-free circuit, making it difficult to detect the Trojan.

Existing research for hardware Trojan design can be classified into four categories, as shown in Figure 2. Because a Trojan's trigger and payload mechanisms determine the difficulty of activation and detection, this has motivated researchers to explore and evaluate novel triggers and payloads. For instance, new triggers utilize don't-care states in a design [Dunbar and Qu 2014] or silicon wearout mechanisms [Shiyanovskii et al.

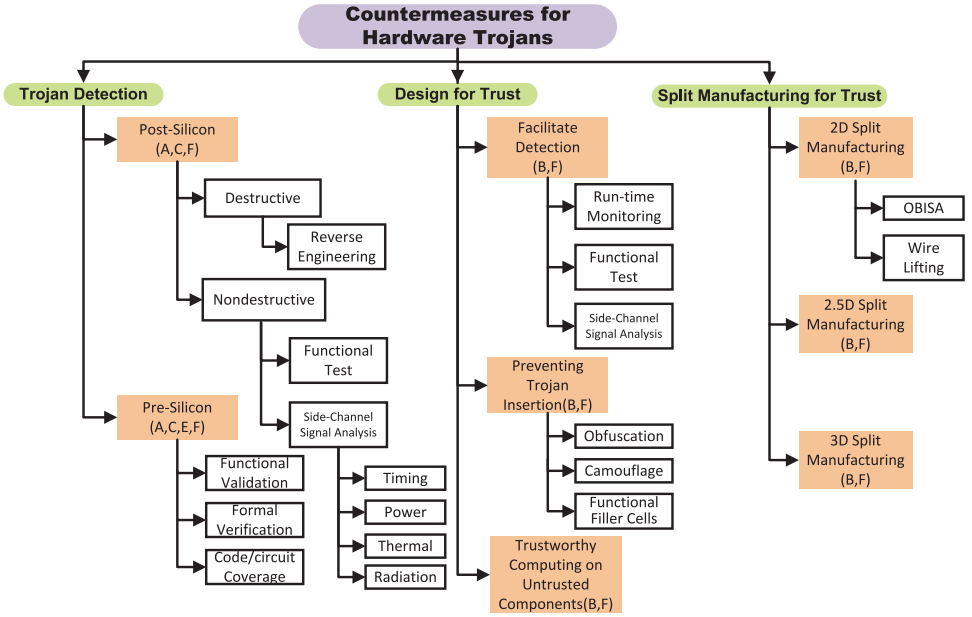


Fig. 3. The taxonomy of hardware Trojan countermeasures.

2010; Zhang et al. 2013] for Trojan activation. New payloads might generate intentional side-channel signals to leak secret information [Lin et al. 2009]. Since extra circuitry introduced by Trojan trigger and payload inevitably causes some side effects, such as additional area, timing, power, or radiation, they could be utilized by defenders for Trojan detection. Thus, to make their Trojan more stealthy and avoid being detected, methodologies have been proposed to optimize Trojan designs and minimize Trojan impact on the original design as much as possible [Cha and Gupta 2014; Tsoutsos and Maniatakos 2014]. Finally, the research community requires standard trust test vectors and benchmarks with a variety of Trojans to compare different techniques fairly. A series of standard benchmarks have been developed for different levels (RTL, gate level, layout) and Trojan types (available at Trust-hub.org [Salmani et al. 2013]).

2.2. Countermeasures Against Hardware Trojans

More research focuses on countermeasures that are able to address or mitigate potential hardware Trojan threats in the supply chain. Generally, they are classified into three broad categories, and further can be classified into several subcategories, as shown in Figure 3.

2.2.1. Trojan Detection. Trojan detection is the most straightforward and commonly used way to deal with hardware Trojans. It aims to verify the existing designs and fabricated ICs without any supplementary circuitry. They are performed either at the design stage (i.e., presilicon) to validate IC designs or after the manufacturing stage (i.e., postsilicon) to verify fabricated ICs.

Postsilicon detection techniques can be classified into destructive and nondestructive methods, as illustrated in Figure 3. Destructive methods typically use destructive reverse-engineering techniques to depackage an IC and obtain images of each layer in order to reconstruct the design-for-trust validation of the end product. Destructive reverse engineering has the potential of giving 100% assurance that any malicious modification in the IC can be detected, but it is high cost and could take

several weeks and months to do this for an IC of reasonable complexity. Additionally, at the end of this invasive process, the IC cannot be used, and we only get the information for a single IC sample. Hence, in general, destructive approaches are not considered viable for Trojan detection. However, destructive reverse engineering on a limited number of samples can be attractive in order to obtain the characteristics of a golden batch of ICs, which will be discussed in Section 5.2. Bao et al. [2014] propose to adapt a well-studied machine-learning method, the one-class support vector machine (SVM), to identify Trojan-free ICs for the golden model.

Nondestructive techniques try to authenticate fabricated ICs from untrusted foundry through functional tests or side-channel signal analysis:

—Functional tests need to activate Trojans by applying test vectors and comparing the responses with the correct results. While at first glance this is similar in spirit to manufacturing tests for detecting manufacturing defects, conventional manufacturing tests using functional/structural/random patterns perform poorly to reliably detect hardware Trojans [Bhunia et al. 2014]. Intelligent adversaries can design Trojans that are activated under very rare conditions, so they can go undetected under structural and functional tests during the manufacturing test process. Banga and Hsiao [2009] and Chakraborty et al. [2009] develop test pattern generation methods to trigger such rarely activated nets and improve the possibility of observing Trojan's effects from primary outputs. However, due to the numerous logical states in a circuit, it is impractical to enumerate all states of a real design. Additionally, instead of changing the functionality of the original circuit [Wang et al. 2008], a Trojan may transmit information (e.g., with an antenna) or modify the specification. Functional tests fail to detect these kinds of Trojans.
—Side-channel signal analysis approaches are able to detect hardware Trojans by measuring circuit parameters, such as delay [Jin and Makris 2008; Xiao et al. 2013], power (transient [Agrawal et al. 2007] and leakage power [Aarestad et al. 2010]), temperature [Forte et al. 2013], and radiation [Stellari et al. 2014; Zhou et al. 2015]. They take advantage of side effects (i.e., extra path delay, power, heat, or electromagnetic radiation) caused by additional circuits and/or activity from Trojan trigger/payload activation. However, the majority of the detection techniques assume that "golden ICs" (Trojan-free ICs) are available for comparison in order to identify Trojan-infected ICs. In addition, while side-channel analysis methods may succeed in detecting Trojans to some degree, the difficulty lies in achieving high coverage of every gate or net and in extracting the tiny, abnormal side-channel signals of hardware Trojans in the presence of process and environmental variations. As the feature size of ICs shrinks and the number of transistors grows, the increasing levels of process variations can easily mask the small side-channel signals induced by low-overhead and rarely triggered Trojans. Recently, Zhou et al. [2015] proposed a backside imaging method to produce a pattern based on filler cells placed in the IC layout, since the authors observed that fill cells are more reflective than other functional cells. Although this technique does not require golden chip, the comparison between the simulated image and measured optical image still suffers from the variations in the manufacturing process. Further, the time required to image the chips and the resolution of backside imaging are challenges.

Presilicon Trojan detection techniques are used to help SoC developers and design engineers to validate third-party IP (3PIP) cores and their final designs. Existing presilicon detection techniques can be broadly classified into functional validation, code/structural analysis, and formal verification.

—The principal idea of functional validation is the same as the functional tests described earlier. The functional validation is conducted with simulation, while functional tests have to be performed on a tester for applying input patterns and collecting output responses. Therefore, existing techniques for functional tests are also applicable to functional validation. Of course, function validation also inherits functional tests' pros and cons.
—HDL analysis can be performed on behavioral [Zhang and Tehranipoor 2011a] or structural [Hicks et al. 2010] codes to identify redundant statements or circuits that may be a part of a Trojan. Structural analysis can also employ quantitative metrics to mark signals or gates with low activation probability as suspicious [Salmani and Tehranipoor 2013; Waksman et al. 2013]. Additionally, Oya et al. [2015] attempt to identify the vulnerabilities by extracting Trojan features from several existing Trojan benchmarks. The limitations of code/structural analysis techniques are that they do not guarantee Trojan detection, and manual postprocessing is required to analyze suspicious signals or gates and determine if they are a part of a Trojan.
—Formal verification is an algorithmic-based approach to logic verification that exhaustively proves a predefined set of security properties that a design should satisfy [Zhang and Tehranipoor 2011a; Rathmair et al. 2014; Rajendran et al. 2015]. To check if a design honors these properties, one converts the target design into a proof-checking format (e.g., Coq) [Love et al. 2011]. However, formal verification techniques could fail to detect additional unexpected functionality introduced by Trojans while satisfying these properties.

2.2.2. Design-for-Trust. As described in the previous section, detecting a quiet, low-overhead hardware Trojan is still very challenging with existing techniques. A potentially more effective way is to plan for the Trojan problem in the design phase through design-for-trust. These methodologies are classified into three classes according to their objectives. The first class of design-for-trust (DfT) approaches aims to facilitate the detection approaches discussed in Section 2.2.1:

—**Facilitate Functional Test:** Triggering a Trojan from inputs and observing the Trojan effect from outputs are difficult due to the stealthy nature of Trojans. A large number of low-controllable and low-observable nets in a design significantly hinder the possibility of activating a Trojan. Salmani et al. [2012] and Zhou et al. [2014] attempt to increase controllability and observability of nodes by inserting test points into the circuit. Another approach proposes to multiplex two outputs of a DFF, Q and $\bar{Q}$, through a 2-to-1 multiplexer and select either of them. This extends the state space of the design and increases the possibility of exciting/propagating the Trojan effects to circuit outputs, making them detectable [Banga and Hsiao 2009]. These approaches are beneficial not only to functional-test-based detection techniques but also to side-channel-based methods that need partial activation of Trojan circuitry.
—**Facilitate Side-Channel Signal Analysis:** A number of design methods have been developed to increase the sensitivity of side-channel-based detection approaches. Salmani and Tehranipoor [2012] propose to minimize background side-channel signals by localizing switching activities within one region while minimizing them in other regions through a scan-cell reordering technique. Additionally, some newly developed structures or sensors are implemented in the circuit to provide a higher detection sensitivity compared to conventional measurements. Ring oscillator (RO) structures [Rajendran et al. 2011], shadow registers [Li and Lach 2008], and delay elements [Ramdas et al. 2014] on a set of selected short paths are inserted for path delay measurements. RO sensors [Zhang and Tehranipoor 2011b] and transient current sensors [Narasimhan et al. 2012; Cao et al. 2013] are able to improve sensitivity

to voltage and current fluctuations caused by Trojans, respectively. Besides, integration of process variation sensors [Cha and Gupta 2012; Liu et al. 2014a] can calibrate the model or measurement and minimize the noise induced by manufacturing variations.

—**Runtime Monitoring:** As triggering all types and sizes of Trojans during presilicon and postsilicon tests is very difficult, runtime monitoring of critical computations can significantly increase the level of trust with respect to hardware Trojan attacks. These runtime monitoring approaches can utilize existing or supplemental on-chip structures to monitor chips' behaviors [Bloom et al. 2009; Dubeuf et al. 2013] or operating conditions, such as transient power [Narasimhan et al. 2012; Jin and Sullivan 2014] and temperature [Forte et al. 2013]. They can disable the chip upon detection of any abnormalities or bypass it to allow reliable operation, albeit with some performance overhead. Jin et al. [2012] present a design of an on-chip analog neural network that can be trained to distinguish trusted from untrusted circuit functionality based on measurements obtained via on-chip measurement acquisition sensors.

The second class of DfT consists of preventive approaches that attempt to thwart hardware Trojan insertion by attackers. To insert targeted Trojans, typically attackers need to understand the function of the design first. For attackers that are not in the design house, they usually identify circuit functionality by reverse engineering.

—**Logic Obfuscation:** Logic obfuscation attempts to hide the genuine functionality and implementation of a design by inserting built-in locking mechanisms into the original design. The locking circuits become transparent and the right function appears only when a right key is applied. The increased complexity of identifying the genuine functionality without knowing the right input vectors is able to dwarf the ability of inserting a targeted Trojan by attackers. For combinational logic obfuscation, XOR/XNOR gates could be introduced at certain locations in a design [Roy et al. 2008]. In sequential logic obfuscation, additional states are introduced in a finite state machine to conceal its functional states [Chakraborty and Bhunia 2009]. In addition, some papers [Baumgarten et al. 2010; Liu and Wang 2014; Wendt and Potkonjak 2014] propose to insert reconfigurable logics for logic obfuscation. The design is functional when the reconfigurable circuits are correctly programmed by the design house or end-user.

—**Camouflaging:** Camouflaging is a layout-level obfuscation technique to create indistinguishable layouts for different gates by adding dummy contacts and faking connections between the layers within a camouflaged logic gate [Cocchi et al. 2014; Rajendran et al. 2013]. The camouflaging technique can hinder attackers from extracting a correct gate-level netlist of a circuit from the layout through imaging different layers, so the original design is protected from insertion of targeted Trojans. Additionally, Bi et al. [2014] utilized a similar dummy contact approach and developed a set of camouflaging cells based on polarity-controllable SiNW FETs.

—**Functional Filler Cell:** Since layout design tools are typically conservative in placement, they cannot fill 100% of the area with regular standard cells in a design. The unused spaces are filled with filler cells or decap cells that do not have any functionality. Thus, the most covert way for attackers to insert Trojans in a circuit layout is replacing filler cells, because removing these nonfunctional filler cells has the smallest impact on electrical parameters. The built-in self-authentication (BISA) approach fills all white spaces with functional filler cells during layout design [Xiao and Tehranipoor 2013]. The inserted cells are then connected automatically to form a combinational circuitry that could be tested. A failure during later testing denotes that a functional filler has been replaced by a Trojan.

The third class of DfT is trustworthy computing on untrusted components. The difference between runtime monitoring and trustworthy computing is that trustworthy computing is tolerant to Trojan attacks by design. Trojan detection and recovery at runtime acting as the last line of defense is necessary, especially for mission-critical applications. Some papers employ a distributed software scheduling protocol to achieve a Trojan-activation-tolerant trustworthy computing system in a multicore processor [McIntyre et al. 2010; Liu et al. 2014b]. Concurrent Error Detection (CED) techniques can be adapted to detect malicious outputs generated by Trojans [Keren et al. 2010; Rajendran et al. 2013]. In addition, Reece et al. [2011] and Rajendran et al. [2013] propose to use a diverse set of 3PIP vendors to prevent Trojan's effects. The technique in Reece et al. [2011] verifies the integrity of a design via comparison of multiple 3PIPs with another untrusted design performing a similar function. Rajendran et al. [2013] utilize operation-to-3PIP-to-vendor allocation constraints to prevent collusions between 3PIPs from the same vendor.

For the DfT techniques that require circuitry added during the front-end design phase, the potential area and performance overheads are the chief concerns to designers. As the size of a circuit increases, the number of quiet (low controllability/observability) nets/gates will increase the complexity of processing and produce a large time/area overhead. Thus, the DfT techniques for facilitating detection are still difficult to apply to a large design that contains millions of gates. In addition, the preventive DfT techniques need to insert additional gates (logic obfuscation) or modify the original standard cells (camouflaging), which could degrade the chip performance significantly and affect their acceptability in high-end circuits. The functional filler cells also increase power leakage.

2.2.3. Split-Manufacturing-for-Trust. Split manufacturing has been proposed recently as an approach to enable use of state-of-the-art semiconductor foundries while minimizing the risks to an IC design [IARPA 2011]. Split manufacturing divides a design into Front End of Line (FEOL) and Back End of Line (BEOL) portions for fabrication by different foundries. An untrusted foundry performs (higher-cost) FEOL manufacturing, then ships wafers to a trusted foundry for (lower-cost) BEOL fabrication. The untrusted foundry does not have the access to the layers in BEOL and thus cannot identify the "safe" places within a circuit to insert Trojans.

Existing split manufacturing processes rely on either 2D integration [Vaidyanathan et al. 2014; Jagasivamani et al. 2014; Hill et al. 2013], 2.5D integration [Xie et al. 2015], or 3D integration [Valamehr et al. 2013]. The 2.5D integration first splits a design into two chips fabricated by the untrusted foundry and then inserts a silicon interposer containing interchip connections between the chip and package substrate [Xie et al. 2015]. Therefore, a portion of interconnections could be hidden in the interposer that is fabricated in the trusted foundry. In essence, it is a variant of 2D integration for split manufacturing. During the 3D integration, a design is split into two tiers fabricated by different foundries. One tier is stacked on the top of another tier, and the upper tiers are connected with vertical interconnects called TSVs. Given the manufacturing barriers to 3D in industry, 2D- and 2.5D-based split manufacturing techniques are more realistic today. Vaidyanathan et al. [2014] demonstrate the feasibility of split fabrication after metal 1 (M1) on test chips and evaluated the chip performance. Although the split after M1 attempts to hide all intercell interconnections and can obfuscate the design effectively, it leads to high manufacturing costs. Additionally, several design techniques have been proposed to enhance a design's security with split manufacturing. Imeson et al. [2013] present a k-security metric to select necessary wires to be lifted to a trusted tier (BEOL) to ensure the security when split at a higher layer. However, lifting a large number of wires in the original design will introduce large timing and power overhead

Table I. Comprehensive Attack Models

Model	Description	3PIP Vendor	SoC Developer	Foundry
A	Untrusted 3PIP vendor	Untrusted	Trusted	Trusted
B	Untrusted foundry	Trusted	Trusted	Untrusted
C	Untrusted EDA tool or rogue employee	Trusted	Untrusted	Trusted
D	Commercial off-the-shelf component	Untrusted	Untrusted	Untrusted
E	Untrusted design house	Untrusted	Untrusted	Trusted
F	Fabless SoC design house	Untrusted	Trusted	Untrusted
G	Untrusted SoC developer with trusted IPs	Trusted	Untrusted	Untrusted

and significantly impact chip performance. An obfuscated BISA (OBISA) technique can insert dummy circuits into the original design to further obfuscate the design with split manufacturing [Xiao et al. 2015].

3. LESSON #1: SPECIFY THE ATTACK MODEL

Developing and using precise attack models are critical to make progress in security research, and the research on hardware Trojans is no exception. By analyzing attack models, one can determine what's been covered by existing work and what still needs to be addressed. For example, one would not want to develop an unrealistic Trojan or countermeasure that doesn't fit a useful model. Hence, before developing a new hardware Trojan or countermeasure, the desired attack model should be considered first. Attack models can act as a guide for those new to hardware Trojans, but can also be useful even for the more experienced researchers in the community. Next, we describe comprehensive attack models that can be used to categorize current work, determine research trends, and provide insight for new directions.

3.1. Comprehensive Attack Models

Hardware Trojans can be injected at any phase during design or fabrication by different adversaries, which leads to different adversarial models. Typically, the entire design and fabrication procedure of an SoC chip can be divided into three main phases: IP core development, SoC development, and fabrication. Therefore, three types of companies, third-party IP vendors, SoC developers, and foundries, have opportunities to insert hardware Trojans. Only two attack scenarios are presented in Rostami et al. [2013], while Table I illustrates all seven possible attack models for the potential hardware Trojan threats.

Each model is described as follows:

—**Model A (i.e., untrusted 3PIP Trojan model):** With semiconductor scaling at very deep submicron levels, more functions (including digital, analog, mixed-signal, and radio-frequency) originally integrated on a board level are now being placed on a single-chip substrate (i.e., System-on-Chip or SoC). It is almost impossible for SoC developers to develop all necessary IPs in house, so they have to purchase some third-party IP (3PIP) cores, which could contain hardware Trojans. This adversarial model is very common today as SoC chips are widely used.

—**Model B (i.e., untrusted fab or fabless design house Trojan model):** Fabless design houses outsource the fabrication to offshore third-party foundries with advanced process technologies. An attacker in the foundry can insert Trojans into a design by manipulating the lithographic masks. These Trojans are in the form of addition, deletion, or modification of gates. Since the foundry has access to all layers of the design, it can inject either untargeted Trojans to produce random failures or targeted Trojans after careful reverse engineering to create intended malfunctions. This is a difficult situation for the IC design companies who not only wish to push

performance to the edge by using of-shore state-of-the-art technologies but also want to guarantee security for critical applications. The model has been discussed and studied significantly in academia in the last decade.

—**Model C (i.e., untrusted SoC developer Trojan model):** Since the complexity of SoC design has increased significantly, more specialized engineers and tools must be involved during SoC design. The hardware Trojan threats could be from untrusted third-party commercial EDA tools or rogue designers (also called insider threats).

—**Model D (i.e., untrusted COTS Trojan model):** An increasing number of commercial and military products make use of commercial off-the-shelf (COTS) components. COTS refers to a product available off-the-shelf and not requiring custom development before being put into a system. These components represent all electronic products tailored for specific uses and made available for sale to the general public. Generally, COTS products are typically less expensive compared to custom-designed products, readily available, and user friendly. However, in the case of COTS, none of the development stages are trusted with respect to Trojans.

—**Model E (i.e., untrusted design Trojan model):** This model assumes that the entire supply chain is untrusted except the foundry. What customers know is that the foundry has a very good reputation and the manufacturing process is dependable, but they do not trust the design company and are unsure if the design contains any hardware Trojans. For example, a product could be developed in an unfriendly foreign country. Note that this model may also be applicable to cloned ICs in the market. After reverse engineering a benign (i.e., Trojan-free) IC, a counterfeiter may insert a Trojan into the original design.

—**Model F (i.e., untrusted outsourcer Trojan model):** This model is the sum of attacks in Models A and B. It can be applied to most fabless IC design companies, such as Qualcomm, Apple, and Xilinx. They integrate some IP cores from 3PIP vendors into their SoC designs and fabricate these chips in untrusted third-party foundries.

—**Model G (i.e., untrusted system integrator and foundry Trojan model):** Several semiconductor companies also offer both application-specific integrated circuit (ASIC) design and fabrication businesses in order to satisfy demands from different clients. The clients can request using appointed IP cores for their SoC design. The chips will be shipped back to clients after fabrication, testing, and packaging. Some companies own fabrication facilities and design teams, and they also provide the specialty foundry services for chip design and manufacturing.

3.2. Relationships Between Previous Research and Attack Models

A hardware Trojan attack or countermeasure should be applicable to one or more of the aforementioned attack models/scenarios. Trojan attacks occur in untrusted parties, while countermeasures should be performed in the trusted stage to defeat them. Trojan attacks can be easily categorized into the attack models. We shall spend the remainder of the section focused on classifying countermeasure techniques into their corresponding attack models.

The relationships between countermeasures and attack models are illustrated in Figure 3 by the letters within brackets for each countermeasure. In hardware Trojan detection, presilicon detection techniques are used to help SoC developers and design engineers to validate third-party IP (3PIP) cores and their final designs, since hardware Trojans could be added into 3PIP cores by untrusted IP vendors (Model A), designs by untrusted EDA tools or rogue employees (Model C), or both (Model E). In addition, presilicon detection techniques can partially address attacks for Model F. The postsilicon Trojan detection techniques attempt to detect the existence of Trojans in ICs that are manufactured in untrusted fabrication facilities. One premise is that the design, layout, and testing steps of the design flow shown in Figure 1 are trusted, and the only

Table II. The Distribution of the Targeted Adversarial Models of the 161 Countermeasure Papers

Model	A	B	C	D	E	F	G
Paper count	48	96	22	1	0	144	0
Percentage	29.81%	59.63%	13.66%	0.66%	0%	89.44%	0%

untrusted component is the foundry. Thus, these techniques are primarily restricted to attack Model B. If SoC developers can ensure the trust of their design using presilicon techniques, the presilicon and postsilicon detection techniques can work together to address attacks in Model F. The DfT techniques try to deal with potential Trojan problems at the design phase, so they require the integrity of the design (i.e., Model B). Lastly, split manufacturing techniques also rely on the BEOL portion fabricated by trusted foundries to address untrusted foundry problems.

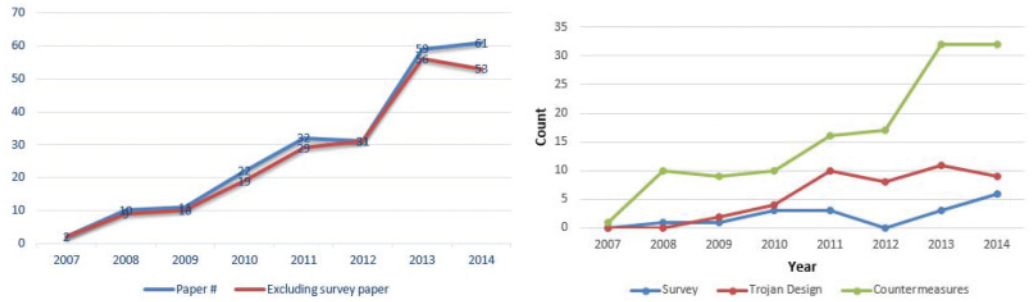
4. LESSONS #2: OBSERVE THE TRENDS

Hundreds of papers and articles have been published in conference proceedings, journals, or magazines in the last decade. In order to analyze the research trends, we have searched all publications with the keywords of "hardware Trojan" in the IEEEXplore digital library. We believe that the publications in IEEEXplore are representative of the research from the hardware security community and they can provide us a general bird's-eye-view picture about the current status. In total, 228 different papers were found from 2007 to 2014. We classify them according to the attack models, the proposed attacks, or countermeasures. We hope the observed trends can educate readers, especially new researchers, about which directions have been explored intensively and which directions are still promising.

4.1. Publications on Different Adversarial Models

Section 2.2 presented a taxonomy for the existing countermeasures against hardware Trojans. Each countermeasure targets one or more adversarial models as shown in Table I. Table II denotes the distribution of the targeted adversarial models from the 161 countermeasure papers. Since the untrusted outsourcer Trojan model (F) has 3PIP cores designed by untrusted IP venders and uses an untrusted fabrication facility for manufacturing, all the countermeasure techniques for the untrusted 3PIP Trojan model (A) and the untrusted fab Trojan model (B) can also be used for Model F. Thus, about 89.44% of papers cover the attacks in Model F. Besides Model F, the untrusted fab Trojan model (B) got the most significant attention, and 96 papers out of 161 (59.63%) target this adversarial model. The untrusted 3PIP Trojan model (A) and untrusted SoC developer Trojan model (C) have also drawn reasonable attention with percentages of 29.81% and 13.66%, respectively. The contribution percentages for these four adversarial models are reasonable, because more and more fabless semiconductor companies need to use 3PIP cores from untrusted IP vendors (Model A), third-party tools from untrusted EDA companies (Model C) and outsource fabrication to offshore foundries (Model B). The remaining adversarial models (D, E, and G) are nearly unexplored. However, we argue that all but Model D can be encapsulated by other models.

The untrusted design model (E) is similar to the untrusted SoC developer model (C). The only difference is that IP cores from third-party vendors are trusted in the untrusted SoC developer model, while 3PIPs are untrusted for the untrusted design model (E). Since the system integrator is untrusted for both threat models, trusted IPs in Model C could still be infected with hardware Trojans during system integration. Actually, these two adversarial models can be merged and considered as one threat model. In the model, all design information is available to a trusted foundry before manufacturing, which provides us opportunities to inspect the design and detect



(a) Overall Publication Trend (b) Classified Publication Trend

Fig. 4. Publication trend.

malicious functionality. Moreover, the untrusted system integrator and foundry model (G) is similar to the COTS model (D). For the same reason that Trojans can be potentially inserted during the system integration process at an untrusted SoC design house, trusted IPs are not trusted anymore after the system integration. Thus, techniques for the COTS model could also be used for the untrusted system integrator and foundry model. To summarize, for these three unexplored threat models, only the COTS threat model (D) really deserves more attention. Thus, it will be further discussed in Section 5.1.

4.2. Publication Trend

In this section, we classify all publications into three categories: survey (that summarize existing techniques), Trojan design, and countermeasures. Figure 4(a) plots the published paper count by year. The blue curve denotes the overall publications for hardware Trojan, which has gone up steadily since 2007. The publication count grows significantly in 2010, 2011, and 2013. It is also reasonable to exclude survey papers from the total publications in order to capture different trends. The new overall publication trend (the red curve) clearly shows that the paper count actually decreased by three in 2014. It is possible that research on hardware Trojan has saturated. Therefore, new research directions should be explored in the hardware security research community.

If we further analyze the publication trend for each category, the papers about countermeasures are far more than the other two types, as shown in Figure 4(b). Since 2011, the countermeasure paper count has almost doubled, but the Trojan design paper count does not grow and even drops a little bit. This might indicate two points: (1) the design of Trojan triggers or payloads has been explored so much that developing novel Trojan attacks has become too challenging, or (2) there are many types of functional Trojans with various triggers or payloads that are still hard to detect, so researchers realize that more effort should be spent on the development of countermeasures. In Figure 4(b), we can see that countermeasure techniques have grown faster in the last 4 years and the paper count has almost doubled. More research resources have been devoted to overcoming the hardware Trojans, so different types of countermeasures have been developed in the last decade. Figure 5 illustrates a timeline about the first appearance of different Trojan countermeasures. In this figure, we can see the general trend shifts from Trojan detection to design-for-trust and split-manufacturing approaches in recent years.

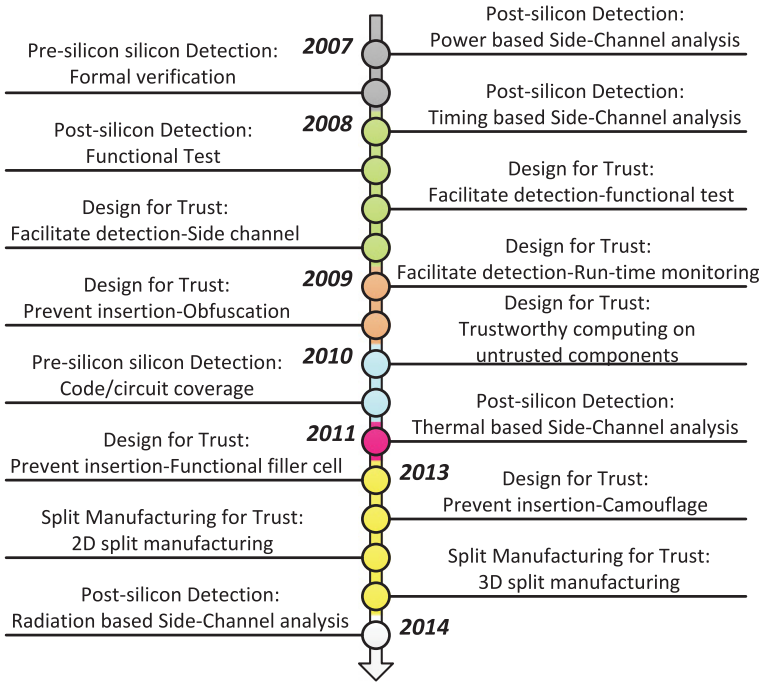
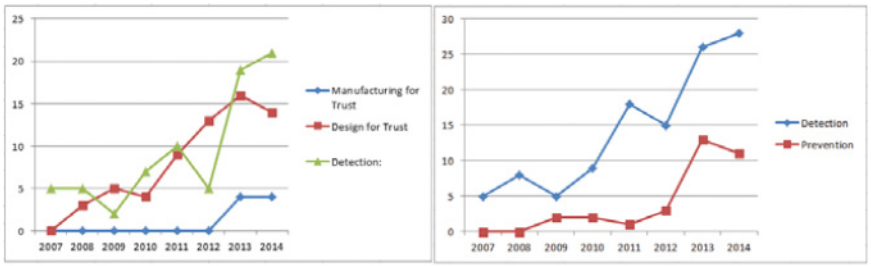


Fig. 5. Timeline for hardware countermeasure techniques.



(a) Trend for different countermeasures (b) Detection vs. Prevention

Fig. 6. The trend for countermeasure techniques.

4.3. Trojan Detection Versus Trojan Prevention

Trojan detection and Trojan prevention attempt to address the hardware Trojan issues from two different perspectives. Trojan detection techniques contain all typical approaches shown in the first column in Figure 3 and a portion of design methodologies in the second column that facilitate Trojan detection. The rest of the countermeasures are trying to either prevent hardware Trojan insertion or prevent a Trojan's malicious behavior, including split-manufacturing-for-trust approaches. Figure 6(b) plots the publication trend for Trojan detection and Trojan prevention. Few papers about Trojan prevention were published before 2013, but prevention methodology got sufficient attention in the last two years (2013 and 2014). Researchers have explored various detection methods and realized that detecting a tiny, quiet, and low-overhead Trojan is still very challenging. Additionally, most of the hardware Trojan detection techniques are still developed based on the available golden model or IC. Obtaining

a golden model or IC in the semiconductor supply chain is extremely difficult or, in other words, practically impossible. Therefore, hardware Trojan prevention might be a more effective and practical way to overcome the hardware Trojan threat. However, as shown in Figure 6(b), newly published detection techniques are still two times more than those prevention techniques in the categories of design-for-trust and split-manufacturing-for-trust. Prevention techniques deserve more attention in the near future.

5. LESSON #3: BE MORE VIGILANT AGAINST THE UNSOLVED PROBLEMS

The previous sections discussed the past research, current and future Trojan attack models, and the trends of current hardware Trojan research. A major takeaway from the previous discussion is just how much is still left unsolved or completely unexplored in the past decade. Here, we elaborate on this more clearly.

5.1. Hardware Trojan Detection for Commercial Off-the-Shelf (COTS) Components

The adversarial model for the COTS component (Model D) is presented in Section 3. To maintain a capability edge, COTS components have also been deployed in many critical systems for military, financial, and transportation applications. As governments around the globe look to cut costs, defense budgets are under fire. This pressure has manifested in the specification and procurement of components for electronic systems [Chidley 2014]. One article from CISCO Systems reports that "the momentum behind using COTS components, rather than highly specialized military equipment, has intensified" [Cisco 2005]. It also mentions some recent examples of acquisitions leveraging COTS in support of military operations. In 2004, the USS Pinckney became the first Aegis class destroyer to be completely outfitted with COTS-based technology, replacing all military-specific computers used previously [Cisco 2005]. The Royal Netherlands Army was the first to employ the Theatre Independent Tactical Army and Air Force Network (TITAAN) that is completely based on COTS software and hardware components [Cisco 2005]. Koch and Rodosek [2012] highlight not only the tendency of using COTS products in recent armament projects but also the security of COTS components including insertion of Trojans. Beaumont et al. [2012] report that the Australian Military must procure and use a large number of COTS electronic components within their systems. This situation also happens in other countries. Ten years ago, only 20% of components in a military system are COTS, but nowadays, the figures have reversed. Currently around 80% of components are COTS, and the percentage could potentially approach 100% in the future [Herr 2015]. The COTS components are widely used in today's systems for several reasons:

—They are typically lower cost because of massive production.
—They are higher in quality and performance due to pressure of competition in the market.
—They are often viewed as more reliable when compared to custom-built chips due to their wide use.
—They are easier to replace due to their availability in the marketplace.

Although COTS components have these advantages and they can pass rigorous qualification processes and extended test cycles at extreme parametric limits and under harsh environmental conditions, trustworthiness of the COTS component is still a major concern, especially for security-critical applications. COTS components are typically procured in a global market from a large array of vendors, and their design, implementation, and fabrication details are largely untraceable. Therefore, COTS consumers must take into account the risks of hardware Trojan attacks if they incorporate COTS components into their critical systems.

While the importance of authenticating COTS components has increased significantly, little work has investigated the hardware Trojan issue in the COTS threat model, as shown in Table II. One approach, called SAFER PATH, attempts to achieve computational integrity by multiple processing elements (PEs) simultaneously voting on a computer program's execution [Beaumont et al. 2012]. Rather than develop and accredit a single trusted processor, the authors augment these untrusted COTS processors with a small subset of trusted logic. This combination can then be used to do the job of a trusted processor, avoiding the unwanted effects of any resident hardware Trojans. This approach saves considerable effort in the accreditation process and allows the use of the latest COTS components to track technological advances. However, this technique has a couple of limitations: (1) It requires physical variability of PEs to avoid the possibility of inserting the same or colluding hardware Trojans into the variant PEs. The authors mentioned that this can be achieved by utilizing unique RTL descriptions of the same PE specification created by independent design vendors using different sets of design tools. These descriptions can then be fabricated at independent facilities, utilizing different processes, geometries, and cell libraries. It is very difficult to meet this requirement for untraceable COTS components from the market. (2) It counters hardware Trojans within processing components. The architecture does not protect other system elements such as memory and data bus from hardware Trojan attacks. Efficient and comprehensive solutions to authenticate COTS and/or achieve secure operations using untrusted COTS components are therefore still needed.

5.2. Hardware Trojan Detection Without Golden Model

Almost all the Trojan detection methodologies rely on the existence of the golden model. Typically, there are two kinds of golden models required for the existing detection methods: golden design or golden IC. Generally, golden designs are needed for presilicon Trojan detection approaches to validate RTL/netlist of IP cores or SoC designs. To verify and authenticate 3PIP cores, a golden functionality or property must be available for hardware Trojan defenders. Moreover, a portion of postsilicon detection approaches are able to detect hardware Trojans based on the existence of golden designs, either at gate or layout levels. The destructive reverse-engineering-based method needs a golden netlist or layout for the comparison. Functional tests also need golden designs to generate test patterns and correct responses. Whether we can obtain a golden design is dependent on three factors: 3PIP core supplier, SoC developer, and third-party development tools they use. Except for Model B, all other adversarial models (A, C, D, E, F, and G) contain untrusted parties that are involved in the design development procedure. Having a golden design available is therefore very unrealistic in most scenarios. On the other hand, since SoC developers are trusted in Models A and F, they can produce a golden SoC design only when 3PIP cores are trusted or can be verified. If the SoC developer is untrusted, it is theoretically impossible to generate a golden design because they are close to the end of the design phase. Therefore, we can claim that a golden design is definitely not available for Models C, D, E, and G.

A golden IC is a fabricated chip with genuine functionality. Golden ICs are required for most postsilicon detection techniques, specifically side-channel methods. Most side-channel techniques require golden ICs as golden references for comparing various side-channel information, including delay, power, temperature, electromagnetic, and so forth. One premise for the assumption of golden IC is that the design sent for fabrication must be trusted. This only occurs for Models B and F. If we have a golden design, a few methods could be able to create golden ICs. The most straightforward way is doing a complete reverse engineering for a batch of manufactured chips to identify golden ICs based on our knowledge of golden design. Both nondestructive and destructive RE techniques presented in Section 2.2.1 could be helpful. Nondestructive

RE does not destroy the chip under investigation, while destructive RE could incur a better resolution. Regardless of nondestructive and destructive RE, the process of RE is an expensive and time-consuming procedure, which incurs prohibitive cost. Another approach is manufacturing a small number of chips in another foundry that is trusted. These chips can be considered as golden ICs. However, the design could be changed if it is fabricated in a different foundry because of a different standard cell library. Two different designs definitely result in two different side-channel signals. Moreover, even for the same design, different fabrication facilities use different process technologies that could lead to variabilities in physical characteristics. Therefore, the separately fabricated ICs are hard to be used as golden ICs for side-channel-based detection.

A few detection techniques without the requirement of the golden model have been developed. Narasimhan et al. [2011] propose a temporal self-referencing approach that compares the current signature of a chip at two different time windows to completely eliminate the effect of process noise, but this technique has a couple of weaknesses. It only works for sequential Trojans that have different states in their FSMs, and changing the Trojan's state is another challenge during test time. Liu et al. [2014a] utilize on-chip process control monitors to capture process variations for each chip and then statistically construct a trusted region for side-channel-based detection. Zhang et al. [2013] try to establish a relationship among side-channel signals in a chip using gate-level characterization and then calculate an estimated side-channel signal value from other measured signals. By comparing the estimated value with the real measured value, the side-channel outliner could be identified. Although these techniques eliminate the requirement of golden IC by modeling, the effectiveness is highly dependent on the accuracy of the model and thus impacts the confidence level of detection. Therefore, the golden model is still a great challenge for detection techniques.

5.3. Hardware Trojans in 3D ICs

As demands accelerate for increasing density, higher bandwidths, and lower power, many IC design houses are gradually adopting 3D ICs with through-silicon vias (TSVs). 3D ICs promise "more than Moore" integration by packing a great deal of functionality into small form factors while improving performance and reducing costs. Three-dimensional IC packages may accommodate multiple heterogeneous dies at different process nodes, which potentially postpones an expensive move to a new process node for all of the same functionality [Cadence 2011].

From a security standpoint, the new development flow for 3D ICs requires a new supply chain ecosystem, which also provides new opportunities for hardware Trojan attacks. Since multiple dies fabricated at different foundries are integrated into one package, trusted and untrusted foundries are involved in the 3D IC manufacturing process. As a result, there are new adversarial models for 3D IC Trojan insertion: some dies are from trusted foundries, while some are not. A complete adversarial model for hardware Trojan attack is required to include the Trojan insertion in 3D ICs. Additionally, the integration process of multiple dies introduces many more intermediate steps, such as die stacking and TVS bonding, compared to conventional single-die IC fabrication. This also provides new opportunities for an attacker to implement Trojans. For example, TSVs could be attacked with malicious modifications. Recently, Hasan et al. [2015] proposed a kind of hardware Trojan that utilizes the unique structure of 3D ICs. Three-dimensional ICs suffer from high temperatures in their middle tiers due to a long heat dissipation path, which could result in large violations in delays that produce a glitch for the Trojan trigger. The proposed technique just leverages the thermal effect of middle tiers in 3D ICs to trigger a Trojan, and it can be eliminated with the progress of heat dissipation in 3D ICs. Apart from this Trojan trigger, more research on hardware Trojans in 3D ICs is needed.

6. HARDWARE TROJAN RESEARCH ROADMAP

Given the aforementioned lessons, trends, and so forth, in this section, we set forth a roadmap for the hardware Trojan research community.

6.1. Authenticate COTS Components

Section 5.1 presents that the COTS component becomes a significant threat to many critical systems, but unfortunately, very few papers focus on this issue. Basically, there might be two possible ways to build up a secure system based on untrusted COTS components. The first class of solutions is to validate a COTS component and make sure it is free of Trojans before deployment. It is very challenging to authenticate COTS components because COTS components are not traceable and their internal detailed design information is not available to defenders. What is available for COTS components is limited to their public documentations, such as datasheets and specifications. Thus, defenders need to verify that the COTS component matches the documented functionality and specifications, nothing more and nothing less. For a COTS component, one might perform numerous and various functional and parametric tests in order to verify whether a COTS component satisfies all the requirements. However, testing a black-box component is very difficult and time-consuming. Additionally, it is impractical to do such an exhaustive test for a large and complex design. Destructive and nondestructive reverse engineering could be used to extract a complete netlist and reveal the internal design information. Simulations on the generated netlist can accelerate the validation process. Another way is trying to get to know the internal details from pins by performing structural and functional analyses. Some functionality could be identified if the design has been explored significantly by a combination of pattern mining from input-output traces and model checking [Li et al. 2012]. It is possible to predict the output of the COTS component. If the predicted output is different from its real output, this mismatch could be caused by a hardware Trojan. The second class employs a secure architecture that can realize trusted computations based on untrusted COTS components that could contain hardware Trojans (e.g., see the SAFER PATH structure [Beaumont et al. 2012]). Besides this, a number of trustworthy computing methods have been developed to address the untrusted 3PIP issue. Some improvements have to be made in order to extend them for the issue of COTS authentication.

6.2. Vulnerability Analysis

As described in Section 2.2, the existing detection techniques are not effective enough. DfT approaches typically require extra circuitry and thus inevitably introduce negative impacts on delay and power. Although many different methods have been developed, designers still need to decide which countermeasure is more effective and economic for one specific design and will be applied first. To answer this question, design-time considerations (e.g., incorporation of DfT features) are becoming essential. If we can get some ideas about the most possible hardware Trojans that could be inserted into the design at some stages, it will be very helpful to guide designers to improve their designs and incorporate appropriate DfT techniques for detecting or preventing hardware Trojans. Instead of deploying DfT features that are too conservative and monitoring every signal/net or gate in a design, the security of a design could be obtained with lower overhead. However, until today, there has been no comprehensive metric available to assess the vulnerabilities of a design to hardware Trojan attacks. Salmani et al. develop a couple of methodologies to evaluate the testability of internal signals and determine a circuit's susceptibility to Trojan insertion at the behavioral level [Salmani and Tehranipoor 2013] and gate level [Salmani et al. 2013]. The analysis quantifies the difficulty of activating each line of a code/signal and observing internal signals and

primary inputs through primary outputs. By revising the design and eliminating low-testable nets, malicious behaviors caused by hardware Trojans might be detected with higher confidence. However, these analyses are not adequate for an RTL or gate-level design since they do not take the circuit's function into account during the analysis. For some security-critical modules, such as encryption/decryption blocks, it is necessary to increase their security level at the design phase. Moreover, the hardware Trojan vulnerability analysis can be extended to the system and layout level. To the system level, DfT techniques for trustworthy computing can be employed to ensure the security of critical computation blocks. Designers are able to decide whether to sacrifice some die area and circuit performance to make the system architecture immune to hardware Trojan attacks. It is very helpful to do vulnerability analysis at the layout level, because it is the last step at the design house and we could make untrusted foundries (attack Model B) have limited spaces for hardware Trojan injection (e.g., through BISA [Xiao and Tehranipoor 2013]).

6.3. Trojan-Resistant or Trojan-Tolerant Design

Because it is very hard to detect or prevent the presence of hardware Trojans in a design or an IP, Trojan-resistant or Trojan-tolerant design methodologies are another way to protect designs from Trojan attacks. Trojan-resistant design mainly employs three strategies: The first way is trying to eliminate Trojan's behaviors. Section 2.2.2 includes a couple of Trojan-tolerant designs or structures that are able to prevent malicious effects even if Trojans are present in a design. For example, trustworthy computing is achieved by incorporating diverse IPs for one task. Another strategy is preventing hardware Trojan from triggering. Since most Trojans are activated conditionally, it also provides us an opportunity to achieve reliable and trusted operations on the hardware platform with Trojans by avoiding triggering these Trojans. The last approach is to prevent hardware Trojan insertion. Several techniques in Section 2.2.2 aim to hamper reverse engineering the design function by attackers at an untrusted foundry so as to prevent targeted hardware Trojan attacks. Those techniques mainly focus on the attack Model B, while most other models have received little if any attention.

6.4. Emerging Hardware Trojans

According to the hardware Trojan taxonomy [Tehranipoor and Wang 2012], hardware Trojans could be inserted at any development phase from specification to assembly and package. Besides the third-party IP vendor and foundry, the third-party testing facility and assembly could also take part in the IC development process shown in Figure 1. However, nearly all hardware Trojans and countermeasures discuss the Trojans that are injected either at the design or fabrication phase. There are few papers that discuss potential threats about Trojans inserted during the specification design, EDA tool design, postmanufacturing test, and packaging process.

Traditional hardware Trojans typically are inserted to perform malicious behaviors under specific conditions. Several novel Trojan triggers and payloads have been proposed, as discussed in Section 2.1. For example, reliability Trojans are created to radically accelerate device aging [Shiyanovskii et al. 2010]. Doping Trojans make hardware Trojans stealthier because they do not introduce additional circuitry into the original design. Thus, hardware Trojan attacks can be made with emerging nanoscale devices. Furthermore, the desire to hold more transistors and integrate more functions into a chip is leading to the adoption of 3D integrated circuit technology. The emerging 3D ICs can change the current circuit architecture and IC supply chain, and could result in new hardware Trojan attacks. Thus, additional attack models (similar in spirit to those presented in Section III) should be developed for 3D ICs. Regarding countermeasures, original techniques for 2D ICs might be modified, in order to apply for 3D ICs. For

example, the Trojan detection approaches based on functional tests must be considered at two levels: prebond and postbond tests. Existing functional test approaches are applicable to prebond tests (wafer test for the silicon die). Postbond tests (package test after die assembly into the package) are needed to target potential hardware Trojans.

7. SUMMARY

Hardware Trojan is a growing research topic that has gained considerable attention over the last decade. Researchers have made significant progress in this domain. In this article, we elaborated on the current status of research in the area of hardware Trojans. By analyzing comprehensive Trojan threat models and all previous research, several open problems were identified and a roadmap for hardware Trojan research was proposed for the community.

REFERENCES

J. Aarestad, D. Acharyya, R. Rad, and J. Plusquellic. 2010. Detecting Trojans through leakage current analysis using multiple supply pad IDDQ. *IEEE Transactions on Information Forensics and Security* 5, 4 (Dec. 2010), 893–904. DOI:http://dx.doi.org/10.1109/TIFS.2010.2061228

S. Adee. 2008. The hunt for the kill switch. *IEEE Spectrum* 45, 5 (May 2008), 34–39. DOI:http://dx.doi.org/10.1109/MSPEC.2008.4505310

D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. 2007. Trojan detection using IC fingerprinting. In *Proceedings of the IEEE Symposium on Security and Privacy, 2007 (SP'07)*. 296–310. DOI:http://dx.doi.org/10.1109/SP.2007.36

M. Banga and M. S. Hsiao. 2009. A novel sustained vector technique for the detection of hardware Trojans. In *Proceedings of the 2009 22nd International Conference on VLSI Design*. 327–332. DOI:http://dx.doi.org/10.1109/VLSI.Design.2009.22

C. Bao, D. Forte, and A. Srivastava. 2014. On application of one-class SVM to reverse engineering-based hardware Trojan detection. In *Proceedings of the 2014 15th International Symposium on Quality Electronic Design (ISQED'14)*. 47–54. DOI:http://dx.doi.org/10.1109/ISQED.2014.6783305

A. Baumgarten, A. Tyagi, and J. Zambreno. 2010. Preventing IC piracy using reconfigurable logic barriers. *IEEE Design Test of Computers* 27, 1 (Jan. 2010), 66–75. DOI:http://dx.doi.org/10.1109/MDT.2010.24

M. Beaumont, B. Hopkins, and T. Newby. 2012. SAFER PATH: Security architecture using fragmented execution and replication for protection against Trojaned hardware. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'12)*. 1000–1005. DOI:http://dx.doi.org/10.1109/DATE.2012.6176642

S. Bhunia, M. S. Hsiao, M. Banga, and S. Narasimhan. 2014. Hardware Trojan attacks: Threat analysis and countermeasures. *Proceedings of the IEEE* 102, 8 (Aug. 2014), 1229–1247. DOI:http://dx.doi.org/10.1109/JPROC.2014.2334493

Y. Bi, P.-E. Gaillardon, X. S. Hu, M. Niemier, J.-S. Yuan, and Y. Jin. 2014. Leveraging emerging technology for hardware security - case study on silicon nanowire FETs and graphene SymFETs. In *Proceedings of the 2014 IEEE 23rd Asian Test Symposium (ATS'14)*. 342–347. DOI:http://dx.doi.org/10.1109/ATS.2014.69

G. Bloom, B. Narahari, and R. Simha. 2009. OS support for detecting Trojan circuit attacks. In *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust, 2009 (HOST'09)*. 100–103. DOI:http://dx.doi.org/10.1109/HST.2009.5224959

Cadence. 2011. 3D ICs with TSVs - design challenges and requirements. Retrieved from http://www.europractice.stfc.ac.uk/vendors/cadence_3DIC_wp.pdf.

Y. Cao, C.-H. Chang, and S. Chen. 2013. Cluster-based distributed active current timer for hardware Trojan detection. In *Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS'13)*. 1010–1013. DOI:http://dx.doi.org/10.1109/ISCAS.2013.6572020

B. Cha and S. K. Gupta. 2012. Efficient Trojan detection via calibration of process variations. In *Proceedings of the 2012 IEEE 21st Asian Test Symposium (ATS'12)*. 355–361. DOI:http://dx.doi.org/10.1109/ATS.2012.64

B. Cha and S. K. Gupta. 2014. A resizing method to minimize effects of hardware Trojans. In *Proceedings of the 2014 IEEE 23rd Asian Test Symposium (ATS'14)*. 192–199. DOI:http://dx.doi.org/10.1109/ATS.2014.44

R. S. Chakraborty and S. Bhunia. 2009. Security against hardware Trojan through a novel application of design obfuscation. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, 2009 (ICCAD'09)*. 113–116.

R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia. 2009. MERO: A statistical approach for hardware Trojan detection. In *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'09)*. Springer-Verlag, Berlin, 396–410. DOI: http://dx.doi.org/10.1007/978-3-642-04138-9_28

A. Chidley. 2014. Use COTS parts to cut costs in military and aerospace systems. *Electronic Design Magazine* Retrieved from http://electronicdesign.com/components/use-cots-parts-cut-costs-military-and-aerospace-systems.

Cisco. 2005. Defense agencies meet readiness challenges with commercial off the shelf (COTS)-based systems. Retrieved from http://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/space_COTS_v2.pdf.

R. P. Cocchi, J. P. Baukus, L. W. Chow, and B. J. Wang. 2014. Circuit camouflage integration for hardware IP protection. In *Proceedings of the 2014 51st ACM / EDAC / IEEE Design Automation Conference (DAC'14)*. 1–5. DOI: http://dx.doi.org/10.1145/2593069.2602554

DIGITIMES. 2012. Trends in the global IC design service market. Retrieved from http://www.digitimes.com/news/a20120313RS400.html?chid=2.

J. Dubeuf, D. Hely, and R. Karri. 2013. Run-time detection of hardware Trojans: The processor protection unit. In *Proceedings of the 2013 18th IEEE European Test Symposium (ETS'13)*. 1–6. DOI: http://dx.doi.org/10.1109/ETS.2013.6569378

C. Dunbar and G. Qu. 2014. Designing trusted embedded systems from finite state machines. *ACM Transactions on Embedded Computing Systems* 13, 5s, Article 153 (Oct. 2014), 20 pages. DOI: http://dx.doi.org/10.1145/2638555

D. Forte, Chongxi Bao, and A. Srivastava. 2013. Temperature tracking: An innovative run-time approach for hardware Trojan detection. In *Proceedings of the 2013 IEEE / ACM International Conference on Computer-Aided Design (ICCAD'13)*. 532–539. DOI: http://dx.doi.org/10.1109/ICCAD.2013.6691167

S. R. Hasan, S. F. Mossa, O. S. A. Elkeelany, and F. Awwad. 2015. Tenacious hardware Trojans due to high temperature in middle tiers of 3-D ICs. In *Proceedings of the 2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS'15)*. 1–4. DOI: http://dx.doi.org/10.1109/MWSCAS.2015.7282148

W. Herr. 2015. Keynote talk: Is it safe? In *Proceedings of the 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST'15)*.

M. Hicks, M. Finnicum, S. T. King, M. Martin, and J. M. Smith. 2010. Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy (SP'10)*. 159–172. DOI: http://dx.doi.org/10.1109/SP.2010.18

B. Hill, R. Karmazin, C. T. O. Otero, J. Tse, and R. Manohar. 2013. A split-foundry asynchronous FPGA. In *Proceedings of the 2013 IEEE Custom Integrated Circuits Conference (CICC'13)*. 1–4. DOI: http://dx.doi.org/10.1109/CICC.2013.6658536

IARPA. 2011. Trusted integrated circuits (TIC) program announcement. Retrieved from http://www.fbo.gov.

F. Imeson, A. Emtenan, S. Garg, and M. V. Tripunitara. 2013. Securing computer hardware using 3D integrated circuit (IC) technology and split manufacturing for obfuscation. In *Proceedings of the 22nd USENIX Conference on Security (SEC'13)*. USENIX Association, Berkeley, CA, 495–510.

M. Jagasivamani, P. Gadfort, M. Sika, M. Bajura, and M. Fritze. 2014. Split-fabrication obfuscation: Metrics and techniques. In *Proceedings of the 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'14)*. 7–12. DOI: http://dx.doi.org/10.1109/HST.2014.6855560

Y. Jin and Y. Makris. 2008. Hardware Trojan detection using path delay fingerprint. In *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust, 2008 (HOST'08)*. 51–57. DOI: http://dx.doi.org/10.1109/HST.2008.4559049

Y. Jin, D. Maliuk, and Y. Makris. 2012. Post-deployment trust evaluation in wireless cryptographic ICs. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'12)*. 965–970. DOI: http://dx.doi.org/10.1109/DATE.2012.6176636

Y. Jin and D. Sullivan. 2014. Real-time trust evaluation in integrated circuits. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'14)*. 1–6. DOI: http://dx.doi.org/10.7873/DATE.2014.104

R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor. 2010. Trustworthy hardware: Identifying and classifying hardware Trojans. *Computer* 43, 10 (Oct. 2010), 39–46. DOI: http://dx.doi.org/10.1109/MC.2010.299

O. Keren, I. Levin, and M. Karpovsky. 2010. Duplication based one-to-many coding for Trojan HW detection. In *Proceedings of the 2010 IEEE 25th International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'10)*. 160–166. DOI: http://dx.doi.org/10.1109/DFT.2010.26

R. Koch and G. D. Rodosek. 2012. The role of COTS products for high security systems. In *Proceedings of the 2012 4th International Conference on Cyber Conflict (CYCON'12)*. 1–14.

J. Li and J. Lach. 2008. At-speed delay characterization for IC authentication and Trojan horse detection. In *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust, 2008 (HOST'08)*. 8–14. DOI:http://dx.doi.org/10.1109/HST.2008.4559038

W. Li, Z. Wasson, and S. A. Seshia. 2012. Reverse engineering circuits using behavioral pattern mining. In *Proceedings of the 2012 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'12)*. 83–88. DOI:http://dx.doi.org/10.1109/HST.2012.6224325

L. Lin, W. Burleson, and C. Paar. 2009. MOLES: Malicious off-chip leakage enabled by side-channels. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, 2009 (ICCAD'09)*. 117–122.

B. Liu and B. Wang. 2014. Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE'14)*. 1–6. DOI:http://dx.doi.org/10.7873/DATE.2014.256

C. Liu, J. Rajendran, C. Yang, and R. Karri. 2014b. Shielding heterogeneous MPSoCs from untrustworthy 3PIPs through security- driven task scheduling. *IEEE Transactions on Emerging Topics in Computing* 2, 4 (Dec. 2014), 461–472. DOI:http://dx.doi.org/10.1109/TETC.2014.2348182

Y. Liu, K. Huang, and Y. Makris. 2014a. Hardware Trojan detection through golden chip-free statistical side-channel fingerprinting. In *Proceedings of the 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC'14)*. 1–6.

E. Love, Y. Jin, and Y. Makris. 2011. Enhancing security via provably trustworthy hardware intellectual property. In *Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'11)*. 12–17. DOI:http://dx.doi.org/10.1109/HST.2011.5954988

D. McIntyre, F. Wolff, C. Papachristou, and S. Bhunia. 2010. Trustworthy computing in a multi-core system using distributed scheduling. In *Proceedings of the 2010 IEEE 16th International On-Line Testing Symposium (IOLTS'10)*. 211–213. DOI:http://dx.doi.org/10.1109/IOLTS.2010.5560200

S. Narasimhan, X. Wang, D. Du, R. S. Chakraborty, and S. Bhunia. 2011. TeSR: A robust temporal self-referencing approach for hardware Trojan detection. In *Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'11)*. 71–74. DOI:http://dx.doi.org/10.1109/HST.2011.5954999

S. Narasimhan, W. Yueh, X. Wang, S. Mukhopadhyay, and S. Bhunia. 2012. Improving IC security against Trojan attacks through integration of security monitors. *IEEE Design Test of Computers* 29, 5 (Oct. 2012), 37–46. DOI:http://dx.doi.org/10.1109/MDT.2012.2210183

M. Oya, Youhua Shi, M. Yanagisawa, and N. Togawa. 2015. A score-based classification method for identifying hardware-Trojans at gate-level netlists. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'15)*. 465–470.

J. Rajendran, V. Jyothi, O. Sinanoglu, and R. Karri. 2011. Design and analysis of ring oscillator based design-for-trust technique. In *Proceedings of the 2011 IEEE 29th VLSI Test Symposium (VTS'11)*. 105–110. DOI:http://dx.doi.org/10.1109/VTS.2011.5783766

J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri. 2013. Security analysis of integrated circuit camouflaging. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS'13)*. ACM, New York, NY, 709–720. DOI:http://dx.doi.org/10.1145/2508859.2516656

J. Rajendran, V. Vedula, and R. Karri. 2015. Detecting malicious modifications of data in third-party intellectual property cores. In *Proceedings of the 52nd Annual Design Automation Conference (DAC'15)*. ACM, New York, NY, Article 112, 6 pages. DOI:http://dx.doi.org/10.1145/2744769.2744823

J. Rajendran, H. Zhang, O. Sinanoglu, and R. Karri. 2013. High-level synthesis for security and trust. In *Proceedings of the 2013 IEEE 19th International On-Line Testing Symposium (IOLTS'13)*. 232–233. DOI:http://dx.doi.org/10.1109/IOLTS.2013.6604087

A. Ramdas, S. M. Saeed, and O. Sinanoglu. 2014. Slack removal for enhanced reliability and trust. In *Proceedings of the 2014 9th IEEE International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS'14)*. 1–4. DOI:http://dx.doi.org/10.1109/DTIS.2014.6850660

M. Rathmair, F. Schupfer, and C. Krieg. 2014. Applied formal methods for hardware Trojan detection. In *Proceedings of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS'14)*. 169–172. DOI:http://dx.doi.org/10.1109/ISCAS.2014.6865092

T. Reece, D. B. Limbrick, and W. H. Robinson. 2011. Design comparison to identify malicious hardware in external intellectual property. In *Proceedings of the 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11)*. 639–646. DOI:http://dx.doi.org/10.1109/TrustCom.2011.82

M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri. 2013. Hardware security: Threat models and metrics. In *Proceedings of the International Conference on Computer-Aided Design (ICCAD'13)*. IEEE Press, Piscataway, NJ, 819–823.

J. A. Roy, F. Koushanfar, and I. L. Markov. 2008. EPIC: Ending piracy of integrated circuits. In *Design, Automation and Test in Europe, 2008 (DATE'08)*. 1069–1074. DOI:http://dx.doi.org/10.1109/DATE.2008.4484823

H. Salmani and M. Tehranipoor. 2012. Layout-aware switching activity localization to enhance hardware Trojan detection. *IEEE Transactions on Information Forensics and Security* 7, 1 (Feb. 2012), 76–87. DOI:http://dx.doi.org/10.1109/TIFS.2011.2164908

H. Salmani and M. Tehranipoor. 2013. Analyzing circuit vulnerability to hardware Trojan insertion at the behavioral level. In *Proceedings of the 2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT'13)*. 190–195. DOI:http://dx.doi.org/10.1109/DFT.2013.6653605

H. Salmani, M. Tehranipoor, and R. Karri. 2013. On design vulnerability analysis and trust benchmarks development. In *Proceedings of the 2013 IEEE 31st International Conference on Computer Design (ICCD'13)*. 471–474. DOI:http://dx.doi.org/10.1109/ICCD.2013.6657085

H. Salmani, M. Tehranipoor, and J. Plusquellic. 2012. A novel technique for improving hardware Trojan detection and reducing Trojan activation time. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 20, 1 (Jan. 2012), 112–125. DOI:http://dx.doi.org/10.1109/TVLSI.2010.2093547

Y. Shiyanovskii, F. Wolff, A. Rajendran, C. Papachristou, D. Weyer, and W. Clay. 2010. Process reliability based Trojans through NBTI and HCI effects. In *Proceedings of the 2010 NASA/ESA Conference on Adaptive Hardware and Systems (AHS'10)*. 215–222. DOI:http://dx.doi.org/10.1109/AHS.2010.5546257

F. Stellari, Peilin Song, A. J. Weger, J. Culp, A. Herbert, and D. Pfeiffer. 2014. Verification of untrusted chips using trusted layout and emission measurements. In *Proceedings of the 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'14)*. 19–24. DOI:http://dx.doi.org/10.1109/HST.2014.6855562

M. Tehranipoor and C. Wang. 2012. *Introduction to Hardware Security and Trust*. Springer.

N. G. Tsoutsos and M. Maniatakos. 2014. Fabrication attacks: Zero-overhead malicious modifications enabling modern microprocessor privilege escalation. *IEEE Transactions on Emerging Topics in Computing* 2, 1 (March 2014), 81–93. DOI:http://dx.doi.org/10.1109/TETC.2013.2287186

K. Vaidyanathan, B. P. Das, E. Sumbul, R. Liu, and L. Pileggi. 2014. Building trusted ICs using split fabrication. In *Proceedings of the 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'14)*. 1–6. DOI:http://dx.doi.org/10.1109/HST.2014.6855559

J. Valamehr, T. Sherwood, R. Kastner, D. Marangoni-Simonsen, T. Huffmire, C. Irvine, and T. Levin. 2013. A 3-d split manufacturing approach to trustworthy system development. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 32, 4 (April 2013), 611–615. DOI:http://dx.doi.org/10.1109/TCAD.2012.2227257

A. Waksman, M. Suozzo, and S. Sethumadhavan. 2013. FANCI: Identification of stealthy malicious logic using boolean functional analysis. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS'13)*. ACM, New York, NY, 697–708. DOI:http://dx.doi.org/10.1145/2508859.2516654

X. Wang, M. Tehranipoor, and J. Plusquellic. 2008. Detecting malicious inclusions in secure hardware: Challenges and solutions. In *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust, 2008 (HOST'08)*. 15–19. DOI:http://dx.doi.org/10.1109/HST.2008.4559039

J. B. Wendt and M. Potkonjak. 2014. Hardware obfuscation using PUF-based logic. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD'14)*. 270–271. DOI:http://dx.doi.org/10.1109/ICCAD.2014.7001362

K. Xiao, D. Forte, and M. M. Tehranipoor. 2015. Efficient and secure split manufacturing via obfuscated built-in self-authentication. In *Proceedings of the 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST'15)*. 14–19. DOI:http://dx.doi.org/10.1109/HST.2015.7140229

K. Xiao and M. Tehranipoor. 2013. BISA: Built-in self-authentication for preventing hardware Trojan insertion. In *Proceedings of the 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'13)*. 45–50. DOI:http://dx.doi.org/10.1109/HST.2013.6581564

K. Xiao, X. Zhang, and M. Tehranipoor. 2013. A clock sweeping technique for detecting hardware Trojans impacting circuits delay. *IEEE Design Test* 30, 2 (April 2013), 26–34. DOI:http://dx.doi.org/10.1109/MDAT.2013.2249555

Y. Xie, C. Bao, and A. Srivastava. 2015. Security-aware design flow for 2.5d IC technology. In *Proceedings of the 5th International Workshop on Trustworthy Embedded Devices (TrustED'15)*. ACM, New York, NY, 31–38. DOI:http://dx.doi.org/10.1145/2808414.2808420

X. Zhang and M. Tehranipoor. 2011a. Case study: Detecting hardware Trojans in third-party digital IP cores. In *Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST'11)*. 67–70. DOI:http://dx.doi.org/10.1109/HST.2011.5954998

X. Zhang and M. Tehranipoor. 2011b. RON: An on-chip ring oscillator network for hardware Trojan detection. In *Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE'11)*. 1–6. DOI:http://dx.doi.org/10.1109/DATE.2011.5763260

X. Zhang, K. Xiao, M. Tehranipoor, J. Rajendran, and R. Karri. 2013. A study on the effectiveness of Trojan detection techniques using a red team blue team approach. In *Proceedings of the 2013 IEEE 31st VLSI Test Symposium (VTS'13)*. 1–3. DOI:http://dx.doi.org/10.1109/VTS.2013.6548922

B. Zhou, R. Adato, M. Zangeneh, T. Yang, A. Uyar, B. Goldberg, S. Unlu, and A. Joshi. 2015. Detecting hardware Trojans using backside optical imaging of embedded watermarks. In *Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC'15)*. 1–6. DOI:http://dx.doi.org/10.1145/2744769.2744822

B. Zhou, W. Zhang, S. Thambipillai, and J. K. J. Teo. 2014. A low cost acceleration method for hardware Trojan detection based on fan-out cone analysis. In *Proceedings of the 2014 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'14)*. 1–10. DOI:http://dx.doi.org/10.1145/2656075.2656077

Received September 2015; revised January 2016; accepted March 2016