Integration, the VLSI Journal xxx (xxxx) xxx

Contents lists available at ScienceDirect

Integration, the VLSI Journal

journal homepage: www.elsevier.com/locate/vlsi

Defense-in-depth: A recipe for logic locking to prevail

M. Tanjidur Rahman [a], M. Sazadur Rahman [a], Huanyu Wang [a], Shahin Tajik [a], Waleed Khalil [b], Farimah Farahmandi [a], Domenic Forte [a], Navid Asadizanjani [a,**], Mark Tehranipoor [a,*]

[a] Florida Institute for Cybersecurity (FICS) Research, Electrical & Computer Engineering Department, University of Florida, Gainesville, FL, USA
[b] Electrical & Computer Engineering Department, Ohio State University, OH, USA

ARTICLE INFO

Keywords:
Logic obfuscation
Tamper-proof memory
Scan chain
Oracle-guided attack
Physical attack

ABSTRACT

Logic locking/obfuscation has emerged as an auspicious solution for protecting the semiconductor intellectual property (IP) from the untrusted entities in the design and fabrication process. Logic locking disguises the implementation and functionality of the IP by implanting additional key-gates in the circuit. The right output of the locked chip is produced, once the correct key value is available at the input of the key-gates. The confidentiality of the key is imperative for the security of the locked IP as it stands as the lone barrier against IP infringement. Therefore, the logic locking is considered as a broken scheme once the key value is exposed. The logic locking techniques have shown vulnerability to different classes of attacks, such as Oracle-guided and physical attacks. Although the research community has proposed a number of countermeasures against such attacks, none of them is simultaneously unbreakable against Oracle-guided, Oracle-less, and physical attacks. Under such circumstances, a defense-in-depth mechanism can be considered as a feasible approach in addressing the vulnerabilities of logic locking. Defense-in-depth is a multilayer defense strategy where several independent countermeasures are implemented in the device to provide aggregated protection against different attack vectors.

Introducing such a multilayer shielding model in logic locking is the major contribution of this paper. With regard to this, we first identify the core components of logic locking schemes, which need to be protected. Afterwards, we categorize the vulnerabilities of core components according to potential threats for the locking key in logic locking schemes. Furthermore, we propose several defense layers and countermeasures to protect the device from those vulnerabilities. In conclusion, we believe that a logic locking technique with a layered defense mechanism can be a possible solution against IP piracy.

1. Introduction

Over the past two decades, the business model for the semiconductor industry has shifted from vertical to horizontal. In the horizontal model, the original component manufacturers (OCM) outsource different steps of the chip manufacturing process, like intellectual property (IP) design, fabrication, and design-for-test (DFT) structure insertion, to more sophisticated offshore fabrication facilities. This approach makes the manufacturing process less expensive for new technology development and scaling down the existing IPs. However, due to the number of stakeholders involved in design, integration, manufacturing, and distribution located around the globe, the OCM and IP owner/vendor have lost control over the supply chain. As a result, IP piracy, counterfeiting, reverse engineering, and hardware Trojan insertion have become eminent threats in the semiconductor industry. The conventional passive IP protection methods, e.g., patents and copyrights, provide no protection against the aforementioned threats. Researchers have proposed several hardware obfuscation techniques, such as logic locking/obfuscation [1], state space obfuscation [2], and IC camouflaging [3] as an active approach for safeguarding the IP.

Hardware obfuscation is a method of transforming the design and layout of the IP while maintaining the original functionality of it. Among the hardware obfuscation techniques, logic locking is appearing as possible solutions for establishing trust in the hardware design. In logic locking, additional combinational logic gates [1] or state spaces [2] are inserted in the design to protect the implementation and functionality of the IP from exposing. Logic Locking is a key-based hardware obfuscation approach and the inserted logic elements are generally termed as *key-gates*. The output of the chip is unlocked once the key-gates are connected to the unlocking key-sequence which configured by the IP owner or OCM

* Corresponding author.
** Corresponding author.
E-mail addresses: nasadi@ece.ufl.edu (N. Asadizanjani), tehranipoor@ece.ufl.edu (M. Tehranipoor).

https://doi.org/10.1016/j.vlsi.2019.12.007
Received 19 July 2019; Received in revised form 16 October 2019; Accepted 18 December 2019
Available online xxxx

through a non-volatile (NVM) memory after the chip is fabricated.

Although logic locking appeared as a promising protection mechanism against IP piracy, the literature shows that this approach is susceptible to several Oracle-guided attacks [4], like Boolean Satisfiability (SAT) attacks [5,6], Signal Probability Skey (SPS) attacks [7] and key sensitization attacks [8]. Over the last few years, Oracle-less attacks have also proved to be successful in key extraction [9,10]. Over the past several years, the security community has focused on developing countermeasures to hinder those Oracle-guided attacks [11,12]. Although protection against the above-mentioned attacks received a lot of attention, unfortunately, the security of the key itself is still ignored. The reason for such ignorance is lying under the two common assumptions made in those aforementioned attacks. First, as the untrusted foundry does not possess the key during fabrication and has only access to the locked netlist/layout and the scan chain, implemented as DFT, only Oracle-guided are considered as the most acceptable method of key extraction. Second, the unlocking key is written into a *tamper-* and *read-proof* memory, and therefore, is protected against reverse engineering in the field. However, an adversary such as an untrusted foundry with access to most advanced failure analysis (FA) equipment, such as microprobing station, scanning electron microscope (SEM) and laser scanning microscope (LSM), should be more than capable of extracting the unlocking key from a chip by contact-based electrical [13,14] or contactless optical probing [15]. Furthermore, the literature on logic locking does not consider the threat imposed by an end user with full-blown reverse engineering capability [16] and an untrusted 3rd party design service provider [17,18] in the supply chain. The task of a reverse engineer can be made difficult through implementing physical layout obfuscation techniques like camouflage cells, dummy vias, filler cells, etc. in the chip [3,19]. However, the aforementioned layout obfuscation methods do not eliminate the threat of IP piracy by reverse engineering. Thus, key-based obfuscation techniques are less secure against physical attacks than previously thought due to the possibility/ease of key extraction. As a result, after nearly a decade of research, none of the logic locking techniques are able to provide absolute defense against IP piracy/theft and root-of-trust violation.

The security measures developed for IP protection have always been a one-to-one exercise, where a security designer deploys specific technology to counter a specific risk or attack. However, "hackers" are innovative and can bypass any security measure implemented in the chip. Therefore, developing a layered defense approach, known as *defense-in-depth*, can be a more practical approach for addressing the security challenges in the hardware security domain. The similar idea has also been implemented in the cybersecurity community to detect and prevent malicious intruders in a system. A defense-in-depth approach, as shown in Fig. 1, developed for a logic locked device, can defend the locking key value in an obscured system against any attack by deploying several independent protection layers and eventually raising the cost of all attacks to unacceptable levels. Multiple defense layers also reduce the probability of intrusion through any other backdoor which was left open unintentionally. Since multiple defense layers are used for developing defense-in-depth for logic locking, the words "defense-in-depth" and "multi-layer defense" are used interchangeably.

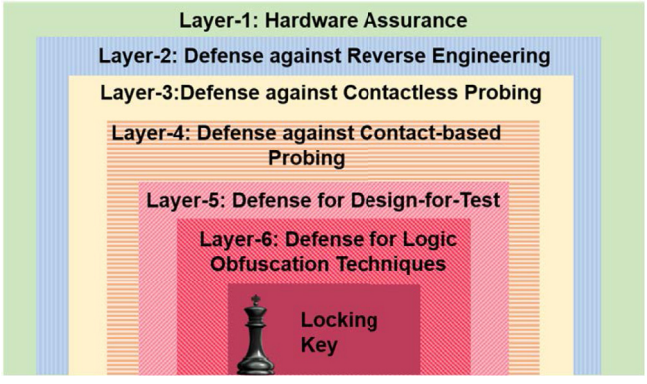


Fig. 1. Multiple protection layers in defense-in-depth implementation for logic obfuscation.

Defense-in-depth for hardware obfuscation can be commonly compared with the "castle approach" as it mirrors the layered defenses in a medieval castle to protect the "king" from an attacker. In an obfuscated hardware, the unlocking key is considered as the king in the chip. Hence, the functionality of the chip is protected by holistic and multiple layered defense scheme implemented as defense-in-depth (Fig. 1).

Contribution. The paper has three major contributions.

1. Presenting an exhaustive survey of vulnerabilities in an obfuscated chip;
2. Developing a comprehensive threat model based on the attacker's intent, capability and availability of assets;
3. Introducing a multi-layer protection approach (defense-in-depth) for the locking key against various threats.

In this paper, we first identify the core components in logic locking schemes, and explain the idea of defense-in-depth. The design steps for developing a multi-layer defense to address the existing vulnerabilities of the logic obfuscation is also explained. Then, we describe the vulnerabilities of the core components in the locked chip. A comprehensive analysis of susceptibilities at different stages of the supply chain is presented as well. Such analysis facilitates the developing of threat models for different adversaries. Based on the vulnerability analysis and threat model, we propose a six-layer security architecture for developing the defense-in-depth concept. Consequently, an in-depth survey of the existing security countermeasures, best practices, and standards depending on the assets defending at each defense layer is presented. Finally, a framework for developing a multi-layered defense-in-depth for hardware obfuscation is outlined for future work.

The paper is organized as follows. In Sect. 2, we discuss the basics of hardware obfuscation and logic locking. In Sect. 3 and 4 the core components in a locked device are identified and the idea of defense-in-depth is introduced, respectively. We presented the susceptibilities of the core components in Sect. 4. Afterwards, in Sect. 6, we explore the existing vulnerabilities of the IC manufacturing process and supply chain and explain threat models for different potential adversaries. The architecture of the defense-in-depth model for the obfuscated chip is presented in Sect. 7. The available countermeasures to thwart the threat against the existing attacks at different layers of defense and security standards are reviewed in Sect. 8. The future research opportunities for developing the security of hardware obfuscation are discussed in Sect. 9.

Finally, we conclude the paper in Sect. 10.

2. Background

2.1. Hardware obfuscation

The objective of the hardware obfuscation is twofold – a) concealing the design secret, such as the algorithm and implementation, against reverse engineering and b) making the design unusable as a black-box and unintelligible for IP piracy. This obscurity can be achieved through changing certain nodes, embedding additional logic gates, altering state-transition-graph or manipulating device or interconnect layers [1–3,19]. Obfuscation methods can be classified into three categories based on the design stage at which the obfuscation is performed [20].

2.1.1. Pre-synthesis obfuscation

Pre-synthesis obfuscation is applied on register-transfer-level (RTL) IPs, which are commonly known as soft IPs. A Soft IP is usually offered in a high-level language like C++, Verilog, or VHDL form. In the case of pre-synthesis obfuscation, the IP is encrypted with well-known encryption

techniques, e.g., IEEE P1735 [21]. Obfuscating the RTL code with a finite state machine (FSM) has also been proposed, where the code later traversed with a key sequence or code-word [20]. The design house acquires a pre-synthesized IP from IP vendors and uses it in a design as a "black-box". However, protecting the obfuscating key sequence from the malicious entity in the supply chain is still appear as a challenge for the security community.

2.1.2. Post-synthesis obfuscation

Post-synthesis obfuscation is the method of hiding the true functionality of the device under attack (DUA) through structural modifications in the design.

The insertion of additional logic elements, interconnects, or modifications in the FSM are prevalent examples of structural modifications in the post-synthesis obfuscation. Combinational logic locking and FSM locking are two most researched post-synthesis obfuscation methods in the literature.

2.1.3. Physical layout obfuscation

The objective of physical layout obfuscation is to thwart the IP reverse engineering and prevent any malicious modifications in the layout. In this method, the physical characteristics of the circuit or the layout is modified to increase ambiguity in cell identification or connectivity. Several techniques have been proposed for layout obfuscation, such as doping based techniques, and dummy contact insertion in the fabrication level [22]. The layout can also be hidden at the cell level using camouflaging cells [3]. Camouflage cells alter the layout of two standard cells with different functionalities to appear identical. Camouflage cells can be developed using real and dummy contacts. As shown in Fig. 2a and 2b, 2-input NAND and NOR gates can be differentiated through analyzing the active region and metal layers. These two gates can be made looked identical (Fig. 2c and 2d) by introducing dummy vias. Inserting dummy gates, dummy filler metal or manipulating doping implant have also been used to generate camouflage cells [19,23,24]. The insertion of dummy vias and identical logic gates introduce ambiguity in image processing based reverse engineering. However, camouflage cells introduce area, power, and delay overhead in the design [25]. In the case of gate-level obfuscation, camouflage cell insertion algorithms [3] have been proposed. Camouflage connections [23]. vanishing vias [26], timing camouflaging [27], and flip-flop obfuscation [28] have also been proposed to prevent reverse engineering.

2.2. Logic locking

Logic locking or logic obfuscation is developed to hide the functionality of an IP by inserting additional logic gates into the netlist of IP. Such protection is provided through embedding additional logic gates into the combinational or sequential parts of the design (Fig. 3). While the former approach is called combinational logic locking, the latter is called FSM locking. In the case of combinational logic locking, the extra embedded logic gates are known as *key-gates*, which are connected to primary inputs that are collectively referred to as the *key*. On the other hand, in FSM locking approaches, the functionality of the IP is obscured with additional states in the state transition graph [2]. Applying a correct sequence of the key, an authorized user can initiate the functional state of the IP/chip. In both techniques, the design provides the correct functionality only if the provided key-input values are correct. Otherwise, the IP does not reveal correct input-output behaviour. The key value is only available to the OCM and the IP owner and not available during the fabrication process. Therefore, once the chips are fabricated, they are transferred to a trusted facility for programming the key, known by the design house, into a secure and tamper-proof *key-storage element*. In the case of combinational logic locking, it has already been shown that random insertion of key-gates may not add a significant security feature to the design [29]. Therefore, several key-gate insertion algorithms, like the insertion of XOR/XNOR gates [1,29], lookup tables [30], and multiplexers [29] have been proposed. Furthermore, Shamsi et al. [31] defined the problem of locking a circuit (e.g., logic locking, camouflaging, and split-manufacturing) as a translational function to the original circuit, which is obscure without a secret key. They defined several notions of security for this translational function under different adversary models.

3. Core components in an obfuscated IC

In this section, we discuss the core components of a locked device. Each component is defined by its functions and involvement in the security of the device. An IC implemented with either combinational or sequential logic locking have five imperative components – (a) Key-storage element; (b) Key-delivery unit; (c) Interconnects; (d) Design-for-test; (e) Obfuscated hardware.

3.1. Key-storage element

In logic locking, after the fabrication, the ICs are transferred to a trusted facility for configuring the key into a secure and tamper-proof *key-storage element* (see Fig. 4). As the key is essential for the correct functionality of the device, storing the key in volatile memory is not suitable for such a purpose. In the case of a volatile key-storage, keeping the chip in a continuous power-up state to maintain the stored value is not a practical approach in terms of power consumption [32]. Therefore, non-volatile memories (NVMs) and one-time programmable (OTP) memories are the conventional choice as key storage elements.

3.2. Interconnects

Interconnects are the metal wires in the chip which connect different

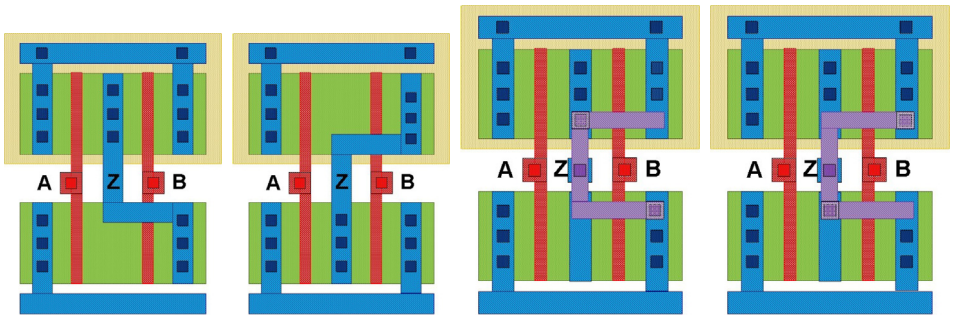


Fig. 2. The camouflage gates described in Ref. [3]. Standard NAND gate (a) and NOR gate (b). These gates could be easily differentiable by looking at the top metal layers. Camouflaged NAND gate (c) and NOR gate (d). These gates have identical top metal layers and are. therefore, harder to identify.

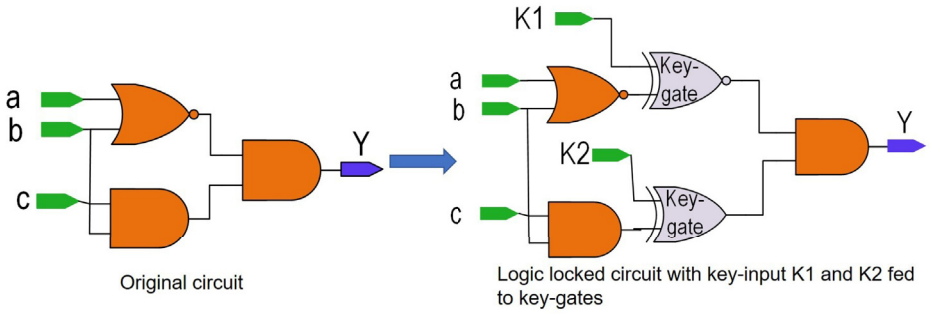


Fig. 3. Simplified example of logic locking method.

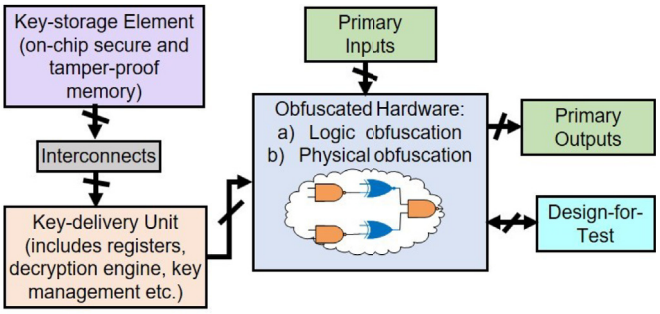


Fig. 4. Core components in an IC implemented with hardware obfuscation.

elements, like transistors, capacitors, etc. and naturally more complex modules, such as memory, processors, cache, etc. in the chip. Depending on the functionality and complexity of the IC, the number of interconnect layers may vary. All devices exchange confidential data between memory and other operational units in the chip through interconnects. For example, The obfuscation keys and other security-critical assets, such as encryption keys, device configuration, and manufacturer firmware are typically stored in a key-storage memory cells. Therefore, these memory cells storing the assets are the root of the security for the design, which needs exclusive protections, such as memory encryption techniques. However, to process the assets in the logic, they have to be transmitted to the logic parts of the chip through chip interconnects. Hence, protecting the interconnects against potential vulnerabilities, such as probing and bus snooping, is equally important in logic obfuscation schemes.

3.3. Key-delivery unit

The key value is compulsory for the operation of the corresponding key-based obfuscated IP. Hence, initialization of any IP must include reading the locking key from the key-storage element. Thereafter, the key must be fed to the key-gates through registers connected to those key-gates [16,33]. These registers, which can be termed as key-registers, should be privileged registers to prevent any inadvertent manipulation of key values and should maintain the stored data during the entire operating period of the IP/chip. The key can be fed directly to the key-gates from the key-storage. However, it does not eliminate the requirement for a read-circuitry, which is also considered in the key-delivery unit, connected to the key-storage. In addition, the confidentiality of the locking key requires to be maintained by digital right management (DRM) policies [34,35]. Moreover, the unlocking key can also be stored in an encrypted format in the key-storage [1]. The encrypted key must be decrypted before fed to the key-gates. This implies the involvement of a decryption engine. Furthermore, reading the key from secured storage may include key-management logic in the chip for cryptomodules as described in Ref. [36]. All the key-read circuitry, key-registers, and key-management logic establish the *key delivery* unit for a locked device and should be protected against asset leakage.

3.4. Design-for-Test

Design-for-Test techniques are widely used in modern system-on-chips (SoCs) to ensure testability of internal circuit elements for monitoring the reliability of the hardware design. This added feature makes it easier to perform structural tests in the hardware design. The manufacturing process is not perfect, making post-silicon validation of designed hardware a vital one. The purpose of functional tests is to verify the correct functionality of the hardware design. However, functional tests are very expensive and the complexity of applying them is too high to realize. To circumvent this obstacle, additional DFT logic is added in the circuit to overcome the difficulty of functional testing in a divide and conquer fashion. For all these obvious reasons, we are considering DFT as a core component in an obfuscated IC. Design-for-test can be inserted in the design by replacing sequential memory elements with scan cells and converting a sequential design into a combinational one to facilitate the structural testing process. However, these scan cells can be used to attack obfuscated hardware designs to extract keys, e.g., key sensitization and Oracle-based attacks.

3.5. Obfuscated hardware

The last core element for the security of the chip is the *obfuscated hardware*. The functionality and layout of the chip can be concealed from an adversary by implementing different logic locking and physical obfuscation techniques. Depending on the objective of the hardware obfuscation, the obfuscation techniques can be applied in three ways:

3.5.1. Device-level hardware obfuscation

At the device level, the layout of the device is disguised by introducing stuck-at-fault or delay manipulation [37]. Changes in doping concentration, manipulating inter-layer dielectric, inserting dummy logic and interconnects are conventional techniques to achieve a device-level obfuscated hardware.

3.5.2. Circuit-level hardware obfuscation

The circuit-level hardware obfuscation focuses on hiding the gate functionality by modifying cell libraries [37]. Camouflage cells, filler cell, dummy vias, and dummy interconnects are examples of circuit-level obfuscation.

3.5.3. System-level or gate-level hardware technique

Logic locking techniques, i.e., combinational logic locking and FSM locking are considered as system-level or gate-level obfuscation techniques. The algorithms used for structural and physical obfuscation methods are also considered as system-level techniques for obfuscating the chip design.

4. Defense-in-depth

4.1. Motivation and definition of defense-in-depth

The vulnerabilities of core components leave a wide attack surface available for different adversaries to extract the assets, i.e., the locking key, layout, and design implementation, from the IC. Naturally, a single defensive mechanism against a specific vulnerability cannot protect the functionality and design of the chip against all potential threats. Once an attacker bypasses the only defensive mechanism implemented in the chip, the security of the entire locking mechanism is broken. For instance, developing mitigation against oracle-guided attacks, namely SAT attacks, cannot defend against the threat of physical attacks, like optical and electrical probing. As a result, multiple layers of countermeasures should be implemented to provide protection for the IP/chip against a wide range of attack vectors. Such a multi-layer defense approach is identified as defense-in-depth. In this paper, we present the defense-in-depth model where different layers of security system address different vulnerabilities of core components.

4.2. Developing the model for defense-in-depth

Developing a model for in-depth defense mechanism for logic obfuscation requires a complex set of analysis on interconnections and dependencies between the different aspects of the supply chain, threat model, system design, protection mechanism, and assets. Besides, providing effective monitoring and protection is required for mitigating the attacks on the IC. Developing a defense-in-depth model for hardware obfuscation can be compiled in four stages as shown in Fig. 5;

1. *Security Analysis of Core Components:* The first step for modeling the defense-in-depth is identifying the vulnerabilities that are present in the core components of logic locking. The assets and methodologies of extracting key and design implementation from an obscured chip, i.e., the attack surface of the IC is identified at this stage.
2. *Threat Model Analysis:* In developing countermeasures and standards for protecting IPs from piracy, overbuilding, or hardware Trojan insertion, the capability of the adversary has been critically underestimated. An attacker can exploit any existing vulnerability in the design which may remain undetected for a long period of time. Therefore, assessing the roles of the stakeholders in the supply chain facilitates in identifying the presence of potential adversaries in the supply chain. The attack surface can also be defined using the vulnerability analysis of supply chain. Analyzing the capabilities, goals of an adversary, and availability of assets is another dimension for selecting the attack methodology and significantly influence the defense-in-depth modeling.
3. *Developing the Defense-in-Depth Architecture:* At this stage, the designer defines the defense layers that protect the chip assets (for example, the defense-in-depth layers depicted in Fig. 1) based on the vulnerabilities of core components, the threat model, desired level of security and design budget allocated for the security of the design secrets. In addition, a designer should consider that, a malicious entity can gain unauthorized access to design assets through the simple shortcomings in the design architecture perimeter, or embedded capabilities in the design that are forgotten, unnoticed, or simply disregarded. Therefore, a multi-layer defense approach must address the protection for the aforementioned 'backdoors' in the device.
4. *Security Standards and Selection of Countermeasures:* The next step for developing defense-in-depth is to identify the effective countermeasures and protection schemes for protecting core components from the adversary. Design budget, i.e., area, power, and energy, defined at the architecture stage plays definitive role in the selection of countermeasures.

5. Security Analysis of Core Components

Although most research efforts have been confined to protect the obfuscated SoC by improving the security of obfuscated hardware and DFT, a comprehensive study about the possible vulnerabilities of other core elements in hardware obfuscation is still absent in the literature. In this section, we will discuss the vulnerabilities of the core elements in an obfuscated device.

5.1. Vulnerabilities of the key-storage element

Protecting the key-storage element is vital for logic locking schemes since the exposure of unlocking key breaks the security of the entire scheme. NVM and OTP memories are considered as possible key-storage candidates in logic locking schemes. NVMs, like ROM, EEPROM, and Flash, are the prominent candidates for key-storage. The NVM can be realized as off-chip or on-chip memory. As off-chip memory is vulnerable to data interception attack at chip boundary, on-chip NVM is the only suitable choice as secure key storage. Although aforementioned memory technologies are widely deployed by the industry as secure and tamper-proof memories, the main vulnerability of NVM is the availability of the data stored in the memory during the power-off state. In this state, the memory remains defenseless against any tampering attack. Therefore, an adversary can deploy advanced FA tools to reverse engineer the memory and readout its contents.

Another option for securing key-storage is OTP memory, such as ROM, electric fuse (eFuse) and antifuse. OTP memory facilitates to configure the device before shipping to the end user once the chip is fabricated. eFuse is a continuous metal or polysilicon shape etched on the silicon surface. An eFuse structure is shown in Fig. 6a. When a voltage is applied to the eFuse, electromigration causes the open circuit in the cell (the broken fuse in Fig. 6a) and program the eFuse [38]. An attacker with access to FA tools can deprocess the entire die and locate the location of eFuse. Later, using the SEM, she can differentiate between the programmed and unprogrammed eFuse link by observing the metal or silicide link of the eFuse. Similar information can be extracted using electrical probing [13,41]. On the other hand, due to scalability into 7 nm node technology, relatively smaller antifuse cells appear as rising solutions to key-storage element. Antifuse is a standard CMOS transistor which acts as a high resistance in its unprogrammed state. Once electrical stress is applied to the gate oxide of the transistor (see Fig. 6b), the transistor acts as a low resistance conductive path. Antifuse can also be placed as via between two metal lines in the chip. In such a case, detecting the location of antifuse is difficult with SEM imaging. SEM provides information about the die surface, i.e., the XY plane of the die. However, the lateral information of the metal layers in the die is required to distinguish the antifuse fabricated as via. The lateral information of the metal layers can only be observed by transmission electron microscopy

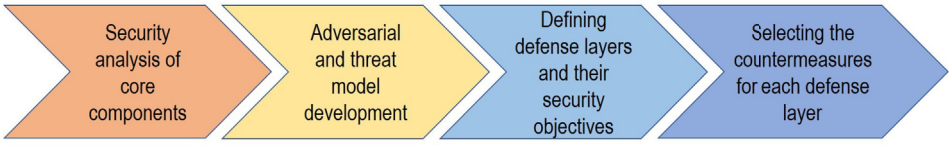
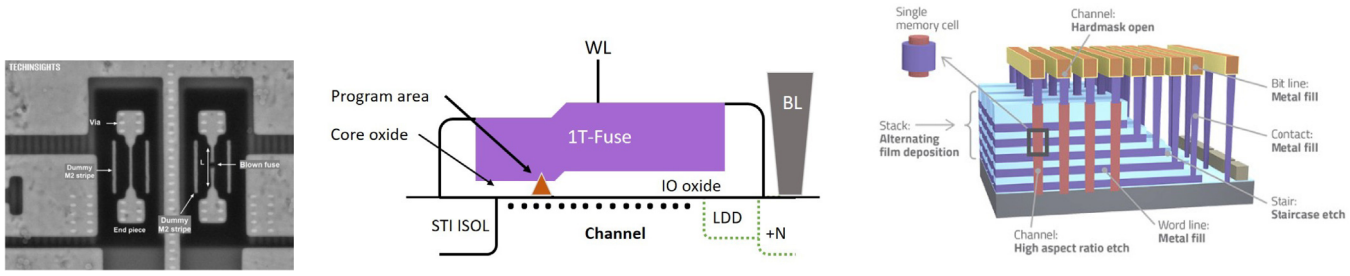


Fig. 5. Steps for developing a defense-in-depth model for logic locking.



(a) (b) (c)

Fig. 6. (a) Difference between before and after program of a TSMC eFuse structure in Qualcomm Gobi MDM9235 Modem 20 nm HKMG [38]; (b) 1T-Fuse Bit Cell in DesignWare OTP NVM IP. The cell is programmed by applying a controlled, irreversible breakdown voltage from the gate through the core (gate) oxide to the channel [39]; (c) Key process steps for 3D Nand fabrication process [40].

(TEM). As sample preparation and imaging for TEM are more challenging than SEM, differentiating between the programmed and unprogrammed bits is difficult but not impossible for antifuse. However, once the location of anti-fuse is extracted the stored bit can be probed. Moreover, all the OTPs require higher breakdown voltage and a large peripheral circuit, which introduces area overhead and higher power consumption [32].

Other conventional examples of NVMs are EEPROM and Flash memories. Each EEPROM cell has two transistors - a floating gate or storage transistor and a select transistor. The storage transistor has a floating gate which traps the electrons. A Flash cell only has the floating gate transistor and uses the same logic storage mechanism as EEPROM. Since both memory technologies use stored charges in the floating gate for storing the bit values, any attempt to image the memory cell with SEM or TEM can disturb the charges distribution and possibly erase the memory content. Therefore, reverse engineering of such NVMs has always been considered as a challenging task; even after the recent advancements in FA tools. Nardi et al. [42] solved the challenge of maintaining the value of stored charge by accessing the memory from the back-side of IC. Once an attacker gets access to the floating gates of EEPROM/Flash, she can use scanning Kelvin probe microscopy (SKPM), scanning probe microscopy (SPM), passive voltage contrast (PVC) or scanning capacitance microscopy (SCM) for extracting the stored value in the EEPROM/Flash [42,43]. However, the security of the 3D Flash chips (see 3D NAND flash cells in Fig. 6c) have yet to be investigated. In the 3D flash technology, the memory cells, previously organized horizontally, are now stacked vertically and connected with pillar and channels. Although such orientation requires further precaution during polishing the back-side of the chip and PVC analysis, the reverse engineering of 3D NAND memory is, in principle, still possible.

Physical unclonable functions (PUFs), as other possible candidates for secure key-storage, was developed to generate keys from intrinsic properties of the device [44]. Although PUF has been assumed to be tamper-evident against physical attacks, they have demonstrated vulnerabilities against several non- and semi-invasive attacks, like photonic emission analysis and laser fault injection [44] Furthermore, the response of PUF differs for each chip due to process variation which makes it incompatible for ASIC design, where the same mask would be used for fabricating all the chip in the same batch. On the other hand, storing the key value in the battery-backed RAM also does not add any significant security feature to the key-storage as they can be read out through optical attacks, such as thermal laser stimulation (TLS) [45].

Data remanence in key-storage like NVM and RAM is another class of vulnerability for all key-storage elements. Data remanence is the residual physical representation (e.g., the trapped charge or voltage) of the data that has been erased from the memory during a tampering attack or regular operation of the chip. A tamper-sensor enclosure can initiate the erasure procedure for memory if the tampering event is detected. The sensor connects the memory to the ground to zeroized the stored data. However, due to data remanence effect, an attacker can exploit the residual property of the memory to extract the content of the memory. The data remanence vulnerability occurs when data retention time exceeds the time required by a malicious entity to read out or dump the stored value in another memory location. Consequently, the protection mechanism can be defeated [46].

5.2. *Vulnerabilities of the interconnects*

Sensitive information transmitted on wires in ICs can be physically extracted using contact-based electrical probing attack [13]. In this type of attack, the chip's wires are contacted by a probe, and as a result, the signal carried by the wires can be read out when the chip is functioning. Therefore, electrical probing is considered as a *contact-based* method for extracting the assets in the chip. Electrical probing attacks can be classified into frontside probing, which is carried out through the passivation layer and upper metal layers, and back-side probing, which is mounted through the silicon substrate.

Due to the large size of probes in comparison to the size of metals' width and available space between wires, the frontside electrical probing is a challenging task. To overcome these limitations, attackers usually deploy focused ion beam (FIB), which is a powerful tool commonly used in the testing, development, and editing of ICs with nanoscale precision, to mill a narrow cavity, get access to the target wire on lower metal layers, and build a conducting path without damaging upper metal layers as shown in Fig. 7. Modern FIB systems, such as ZEISS.

ORION NanoFab, can edit out obstructing circuitry with a 5 nm precision. FIB aspect ratio is a key feature of FIB's capability, which is defined as the ratio between the depth and diameter of the milling cavity.

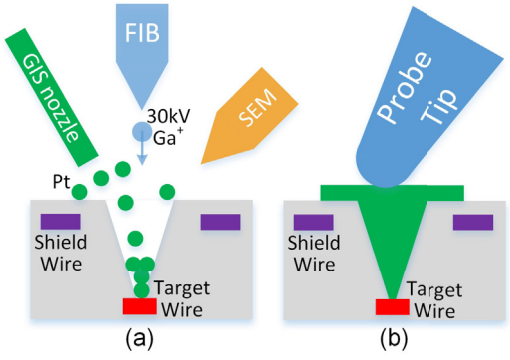


Fig. 7. (a) FIB deposits Platinum in the milling cavity to build a conducting path (green) from the target wire; (b) The deposited conducting path serves as a electrical pad for the probe contact [47]. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

Thus, the higher of the FIB aspect ratio, the thinner of the milling cavity, the less probability to damage signal wires on the chip, and the higher success rate to extract wire values.

Some high-security level chips, such as smart cards, may have shield-like mechanisms to protect the chip against frontside probing attacks. However, this type of countermeasure may still be compromised by bypass and reroute attacks [13] using advanced FIBs. In the case of bypass attacks, the attacker can utilize the limited space between shield wires to approach lower target wires without hurting the adjacent shield wires using high aspect ratio FIB. For reroute attacks, on the other hand, the attacker can build a copy path between two equipotential points on shield wires using FIB's deposition capability, so the original path between these two equipotential points can be cut at will. As a result, even shielding cannot provide adequate security protection and it can still be vulnerable to sophisticated attackers equipped with advanced FIB systems. The electrical probing attack can be mounted from the backside of the IC as well [14]. In this case, the silicon substrate on the backside of the chip is penetrated to create access to the lower metal layers. Therefore, while reaching sensitive wires on the lower metal layers is challenging through frontside attacks, they can be accessed through the backside where there are little to no protection mechanisms.

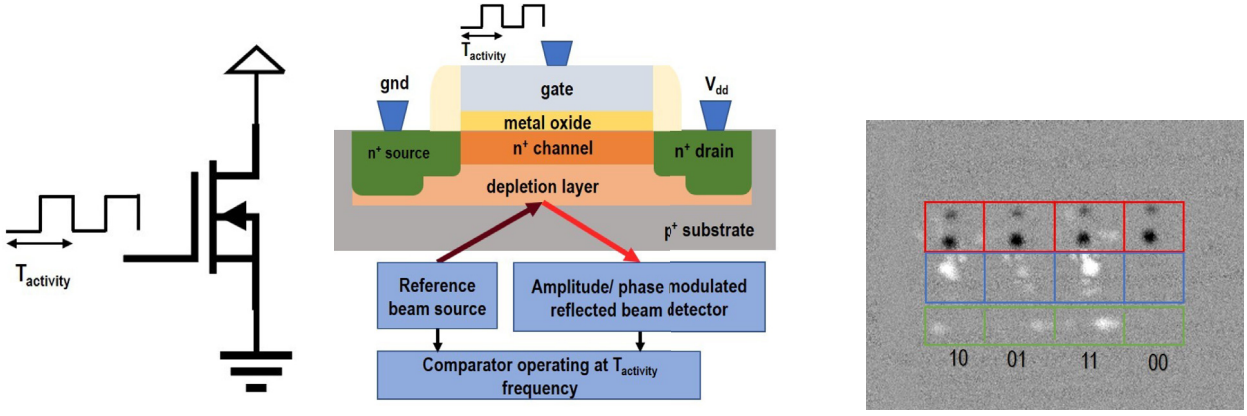
5.3. Vulnerabilities of the key-delivery unit

Similar to contact-based methods, the contactless optical probing [15] techniques can impose the threat of exposing security-sensitive information to an adversary, e.g., the key value in logic locking schemes. Optical probing is a semi/non-invasive chip debugging method, which enables the probing of the volatile and on-die-only values of key-registers and key-gates at run-time. In modern ICs, multiple interconnect layers at the frontside of the chip obstruct the optical path from the transistor. On the contrary, no such protection is available on the backside of the device. Hence, attacking the logic locking and FSM using optical probing is more convenient if conducted from the backside.

In optical probing the chip must be operational. Therefore, the selection of sample preparation method for the DUA depends on the packaging, i.e., non-flip or flip chip packaging technique. In non-flip chips, the die backside can be accessed by decapsulating the packaging. Such challenges can be avoided if the DUA is in a flip-chip package. The silicon substrate in a flip-chip package is usually covered with a heat-sink which can be removed easily using a lab knife and hotplate [16]. Once the chip is decapsulated, the device receives a global polishing to increase the resolution for back-side FIBing and electrical probing. In flip-chip, such polishing is not necessary for optical probing, and therefore, optical probing can be considered as a non-invasive physical attack which makes such attack more attractive to an adversary [16]. Besides, in the case of optical probing, the spatial resolution can be increased if the adversary has access to solid immersion lens (SIL).

To attack the key-delivery unit using optical probing, an adversary requires access to a laser scanning microscope, which is available in advanced FA labs. Since silicon is transparent to near-infrared (NIR) light source, the activity in the die can be measured using electro-optical frequency (EOFM) and electro-optical probing (EOP) [15]. These two methods are major optical techniques used for debugging nanoscale transistors. In both EOP and EOFM, the incident photons with NIR wavelength pass through the back-side of silicon substrate which leads to partial absorption and reflection at interfaces like back-side silicon and active region or first metal layer interconnect. In the case of EOP, the electrical signal at a node modulates the amplitude and phase of reflected light. The modulated light is fed to an optical detector and compared with the reference NIR wavelength laser beam (see Fig. 8b). As the modulation of the reflected beam signal is small, a sufficient signal-to-noise ratio is acquired through running the signal in a certain trigger frequency ($T_{activity}$ in Fig. 8a and 8b) and measuring the signal. In EOFM, a laser scans the region of interest (ROI) on the device under attack and feeds the detected signal from laser reflected signal into a spectrum analyzer acting as a narrow band frequency filter, for example in Fig. 8 the frequency of narrow bandpass filter of the spectrum analyzer is $T_{activity}$. The output from spectrum analyzer is mapped in a 2D image using grayscale or false color representation [15]. Analyzing the output from EOP or EOFM, the data stored in a node is extracted. The EOFM activity of an 8-bit register measured at two different frequency – clock frequency and $T_{activity}$, and stored value in the 8-bit register is shown in Fig. 8c. Hence, an adversary can probe the data stored in the registers from the backside of the chip die without using the invasive methods like FIB.

A malicious entity can always use advanced reverse engineering tools to extract the gate-level netlist of the chip. Access to gate-level netlist enables the intruder to dig deeper in the chip design and localize the key-gates and key-delivery unit or the interconnects carrying the locking key to the chip. Therefore, by learning the operating frequency for the key-delivery unit and using EOFM, an attacker can probe different key-carrying elements like key-gates, key-registers or key-management logic and learn the locking key [16]. Hence, optical probing is a direr



(a) (b) (c)

Fig. 8. (a) The input signal connected to the gate terminal of an n-MOSfet operating at $T_{activity}$ frequency; (b) Reference beam got modulated due to the activity of the transistor. The modulated reflected beam is compared and filtered at the same frequency at the gate is operating; (c) EOFM activity measurement of a 8-bit register. The black dots in red rectangles represent the clock activity, white dots in blue rectangles represent flip-flop activity and white dots in green rectangles represent the output buffer activity. The stored value in each register is mentioned at the bottom of the output buffer.

threat for logic obfuscation as this method can extract the locking key in a contactless manner; without using invasive methods, like FIBing or circuit edit, and contact-based method, i.e., electrical probing.

5.4. Vulnerabilities of the DFT

Jeopardized by the worldwide IC supply chain, scan infrastructure can be used to assist non-invasive attacks, thereby compromising security. The exposed scan chains may leak critical information such as intellectual property (IP) or secret keys to the attackers, which can be carried out by any entity within the IC supply chain. Hence practical solutions are needed to protect ICs against scan-based side-channel attacks [48]. In the last decade, there have been a number of scan-based attacks on various cryptosystems. In Ref. [49], the risk of scan-based attack is presented as a general threat to a stream cipher. To obtain critical information, the attackers can ascertain the internal structure of the scan chain by running encryption in normal mode and then switching to test mode [50]. have successfully uncovered scan-based attacks on the dedicated hardware implementation of the Data Encryption Standard (DES), Elliptic Curve Crypto-systems (ECC), Advanced Encryption Standard (AES), and RSA. Since scan chains directly reveal the internal state of the logic blocks, attackers can use them to perform IP piracy With the knowledge of the design, attackers can also control the chip without authorization by scanning illegal values into the system status registers to disrupt the chip. In light of these threats, ensuring scan security has become a great concern to the industry, and various countermeasures have been proposed which are summarized in Table 1. A detail discussion of these threats and existing countermeasures are discussed below.

- Differential Attack and Defense: The differential attack [51] is based on applying challenge pairs, running the crypto algorithm, and comparing the outputs to extract the key. This attack has been facilitated using scan chain due to added controllability and observability. Through switching from functional mode to test mode, the attacker can identify key flip-flops from the scan chain. Then, the key can be recovered through the already constructed correlation among input pairs, key flip-flops, and key [51]. The most direct solution to refrain from differential attack is to defuse the poly-silicon fuses connecting the scan-in or scan-enable ports [52]; however, this prohibits in-field testing which is a must in advanced ICs. Some test mode protection techniques have been proposed [53,54] which attempt to reset the data registers when the chip is switched to test mode and wrap the non-volatile memories. However, test mode only differential attacks [55] successfully extracted the key.
- Advanced Industrial DFT Techniques: On-chip compression, X-tolerance, and X-masking are considered natural barriers to scan-based attacks [56]. However, the compression bypassing mode is always kept for the sake of debugging and diagnosis. Recently some attacks have been made even in the presence of on-chip compression [55], X-masking [57], and X-toleranc [58].
- Scan Interface Encryption: In addition to the on-chip compression used in advanced DFT structures, scan chain encryption has been developed as countermeasures. In Ref. [59], the scan patterns/responses are decrypted/encrypted at each scan input/output, respectively, which is conducted by highly efficient and secure block cipher at each scan port. But these countermeasures are defeated by resetting attack [60] and flushing attack [61]. By resetting the scan cells or flushing the scan chain with the known patterns, the fixed inverted bits [60] and modified bits [61] in the obfuscated scan chain can be identified so that the plaintext can be deciphered.
- Partial Scan: The secure scan architectures presented in Ref. [50] exclude flip-flops containing sensitive information from the scan chain. However, only part of the scan chain cells can be protected. It becomes very difficult for automatic test pattern generation (ATPG) tools to detect defects in the excluded registers. Furthermore, the extensive use of partial scan can significantly reduce test coverage, which in turn reduces yield.
- Obfuscated Scan: In Refs. [60–64], dummy flip-flops or other obfuscation logic (i.e., inverters, XOR gates, etc.) have been inserted into the scan chain to randomize scan outputs. A scan chain access authorization process usually controls obfuscation. The scan out responses are determined by the test authentication status. However, some obfuscation logic inserted into the scan chain are not robust against reset or flushing attacks [60,61]. More importantly, the scan authorization key bits hidden in the test patterns are usually easy to locate [62–64]. Furthermore, the authentication key bit flipping would make scan out vectors differ, while a non-key bit would not. This would significantly reduce the difficulty of identifying the key bits and becomes vulnerable to bit-role identification attack [65].
- Scan Chain Reordering: In Ref. [66], the order of scan cells is dynamically reconfigured by an unpredictable scrambler, which increases the routing overhead significantly. In Ref. [67], each scan chain is divided into several segments, and then the test controller determines the segments' scanning out sequence. In Refs. [49,60], scan tree architecture is applied to reorder the scan chains. However, these methods still could not defend against a differential attack [55], and require significant change to the DFT flow.
- Combinational Function Recovery Attack: Since the scan chains unfold the sequential logic as combinational and directly reveal the internal states of the circuit, extracting design information from them has become easier. Thus, the device's functionality can be reverse engineered [68].
- Oracle-guided Attacks: While logic locking can be an effective technique to establish trust among different entities of the IC supply chain, it has not seen application due to its lack of attack resiliency. The logic locking is proved to be vulnerable against Oracle-guided attacks which will be discussed in detal in Sect. 5.5.

Table 1
Scan-based attack and countermeasures.

Attacks	Exploits	Existing Countermeasures
Differential [51]/Test mode only Attack [69]	Internal States	Scan encryption [59], DOS [65]
Resetting Attack [60]	Internal	LCSS [62], DOS [65], Lock & Key
Flushing Attack [61]	Secrets	[70], Scan encryption [59]
Bit-role Identification		
Combinational Function Recovery [68]	Functionality	DOS [65]
SAT Attack [5]	Obfuscation Key	SARLock [12], Anti-SAT [11], SFLL [71]

5.5. Vulnerabilities of the obfuscated hardware

The source of the vulnerabilities for obfuscated hardware lies in the techniques used for obscuring the functionality and layout of the chip. Any shortcoming in the security of obfuscation techniques weakens the security of obfuscating key as well as all the assets in the chip. Therefore, we analyze the vulnerability of logic locking and physical obfuscation techniques in detail.

5.5.1. Vulnerabilities of logic locking techniques

In the past decade, there has been a number of attacks proposed to retrieve the key from the logic locked circuit. The attacks available in the literature can be classified into two classes – Oracle-guided attacks and Oracle-less attacks. In Oracle-guided attacks, the attacker has access to an unlocked or functional chip. A functional chip carries the key value in the key-storage element. Therefore, such an IC can generate the correct output for any input pattern and the attacker can make use of the correct input/output pairs to rule out incorrect keys and extract the correct obfuscation key. For example, most logic obfuscation techniques are vulnerable to Boolean satisfiability (SAT)-based oracle-guided attack, key-sensitizing attack [8] and EPIC attack [72]. The key sensitizing

attack utilizes ATPG tool to propagate the effect of a key gate to a primary output. SAT attack [5] breaks most combinational logic obfuscation techniques in a short matter of time by finding distinguishing input patterns (DIPs). DIPs rule out incorrect keys utilizing the output corruptibility of the miter circuit constructed using locked design and activated design. For sequential designs, it is assumed that an IC's internal states can be accessed and controlled via scan chains to read/write the value of the flip-flops. To resist SAT attack, several SAT-resistant logic obfuscation techniques have been proposed- SARLock [12], Anti-SAT [11] and SFLL [71]. SARLock and Anti-SAT resists SAT attack by increasing the number of required distinguishing input patterns (DIPs), thus exploiting a point function to corrupt the output of the design for all the incorrect keys. While these two SAT resistant techniques are strong enough to withstand the power of oracle-guided attacks, they are vulnerable to Bypass attack [73], SPS attack [7], and AppSAT [6] attack. SFLL [71] technique strips some of the functionality of the original design and hides it in the form of a secret key. Once correct secret key is applied, original functionality of the design is restored. SFLL was briefly considered the state-of-the-art SAT resistant logic obfuscation technique. Then a recent functional analysis attack (FALL) [10] was proposed that uses structural and functional analyses on the locked design to identify the locking key, without even having access to an oracle. EPIC attack [72] uses a hill-climbing search based algorithm that monitors test response to guess the secret key. The attack tries to reach zero hamming distance between the test response of the activated IC and the encrypted circuit by flipping the individual bits of the initial key guess if the flip reduces the hamming distance. Along the aforementioned Oracle-guided attacks, side-channel information like differential power analysis and test data can be used to learn the key value in a locked chip. Over the past several years, the security community has focused on assessing the vulnerabilities due to Oracle-guided attacks. While protecting the structural obfuscation from the above-mentioned attacks received so much attention, unfortunately, no evaluation has been performed to find the information that can be extracted from the netlist alone. The change due to logic locking in the netlist is local, i.e., the key-gates combine with the logic elements in the netlist to transform a new structure. Such structure can also be identified if the adversary has prior knowledge about the synthesis tools. Therefore, in desysnthesis attack [74], authors have proposed, re-synthesizing the locked netlist with a random key and then using hill climbing search to find the key value yields the maximum similarity between the locked netlist and re-synthesized netlist. Using machine learning techniques, it is also possible to revert the locked circuit into the pre-synthesis version of the design and retrieve the original design and functionality of the chip [9,75].

5.5.2. Physical vulnerabilities to reverse engineering the obfuscated hardware

Physical obfuscation mainly focuses on preventing the reverse engineer from stripping the ICs layer by layer and extracting gate-level for duplicating a netlist without authorization of the IP holder. Shrinking the device dimension was never an issue for reverse engineering. Continuous improvement and automation in FA tools along with the netlist extraction software, such as Pix2Net, Degate, etc. always proved to be successful against smaller node technologies like 14 mm. The reverse engineering software use image processing techniques to identify the functionality of the gates. In order to thwart automated image processing based reverse engineering, several subtle obfuscation techniques like gate camouflaging, dummy contacts, dummy interconnects, filler cells, variation in doping concentration have been proposed [3,22]. However, layout obfuscation methods can be detected using advanced imaging tools like PVC, SEM or dynamic optical beam induced current circuit analysis (DOCA) [76]. Using PVC or varying the beam voltage of an SEM, a reverse engineer can distinguish between the active cell and filler cells due to variation in doping concentration [37].

The aforementioned camouflaging techniques are not only vulnerable to failure analysis tools, but they are also vulnerable to several attack methods, as for example SAT attack, brute force attack, and behavior analysis. An adversary can isolate the camouflage gates and sensitize the output of the gate using input pattern to resolve the functionality of the gate using the brute force attack [3]. Again, the adversary can perform behavior matching against a library of components with known functionalists to expose the functionality. SAT-based de-camouflaging and removal attacks can also debunk the gate level camouflaging [77].

5.6. Security breach through Hardware Trojan insertion

Device assets such as the locking key should be protected by hardware. The hardware contains physical countermeasures against several physical attacks, tampering, side-channel analysis and probing in particular. The aforementioned protection imposes a significant barrier to attackers thus implicitly providing a basic level of protection against key extraction. However, an untrusted foundry can intentionally introduce side-channel leakage by inserting hardware Trojan in the design, in a similar fashion described in Refs. [78,79] for the key to cryptomodule. Identifying the location of the key-storage elements and the key-delivery unit and implementing a Trojan to facilitate the side-channel analysis can empirically serve the purpose. Hence, the possibility of a security breach due to the presence of hardware Trojan into the design cannot be ignored.

5.7. Summary of the vulnerabilities of the core elements

Each of the core components described in Sect. 3 acts as a link in the web of logic locking to defend the chip design from IP piracy and violation of root-of-trust. On the basis of the above discussion, the attack methods for breaking into the core components of an obfuscated chip and tamper its security can be categorized in five classes;

1. attacks that involve either structural or information reverse engineering methods,
2. attacks that involve contactless probing methods like optical probing. In such methods, no direct contact with the transistors is required for extracting the secret data like locking key,
3. attacks that involve contact-based probing methods like electrical probing,
4. attacks that involve access to design-for-test structure such as scan chain, and
5 attacks on logic obfuscation techniques, for example, SAT and SAIL attack.

Fig. 9 summarize the vulnerabilities of the core components based on the above-mentioned five attack categories.

6. Threat Model Analysis: security threats in IC supply chain

In this section, the security and trust issues in the supply chain, the stake holders, and the threat analysis for potential adversaries are discussed.

6.1. Vulnerability analysis in supply chain of SoC

In the last decade, the SoC supply chain has shifted to a horizontal business model. In the horizontal model, several stakeholders are involved in the manufacturing steps and supply chain of the SoC (Fig. 10). Usually, OCM starts the design process by acquiring the IP which is developed in-house or purchased from *third-party IP vendors* (3PIP Vendors). Later, the SoC designer incorporates the in-house developed and procured 3PIPs to generate the RTL specification of the whole SoC. The SoC integrator synthesizes the RTL description into a gate-level netlist using a computer-aided design (CAD) tool, for example, Design Compiler from Synopsys. The gate-level netlist then goes through formal equivalence checking to verify that the netlist is functionally

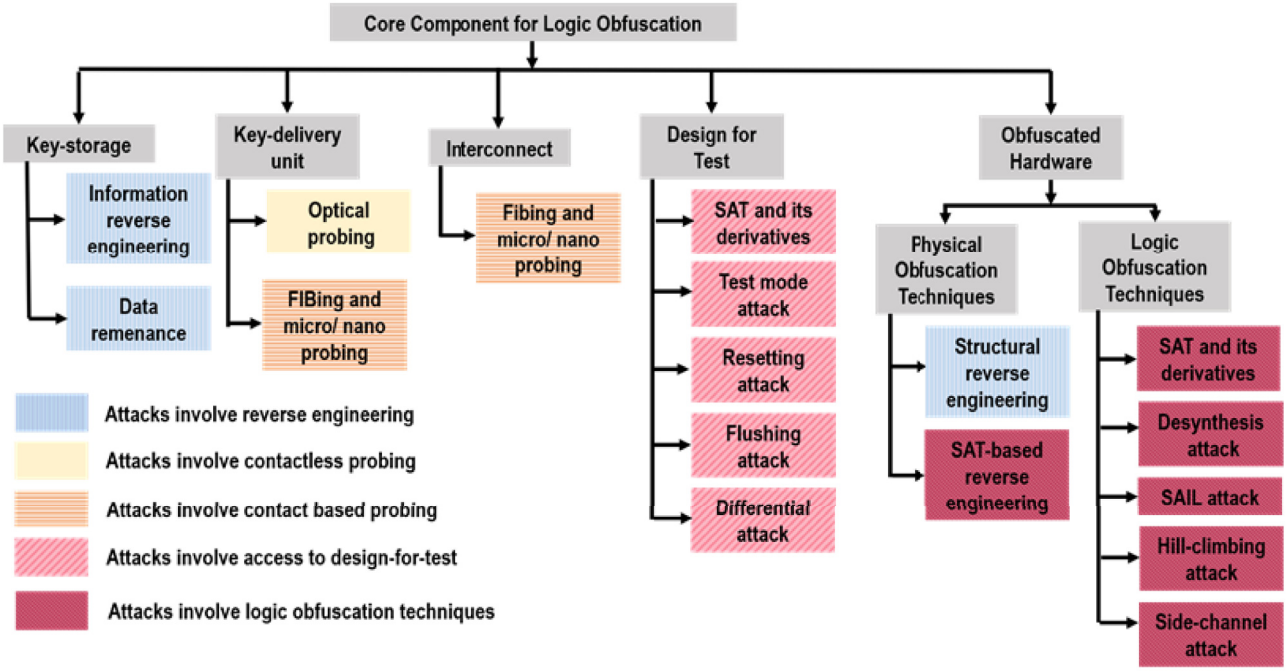


Fig. 9. Attack methods for the core elements in a logic locked chip.

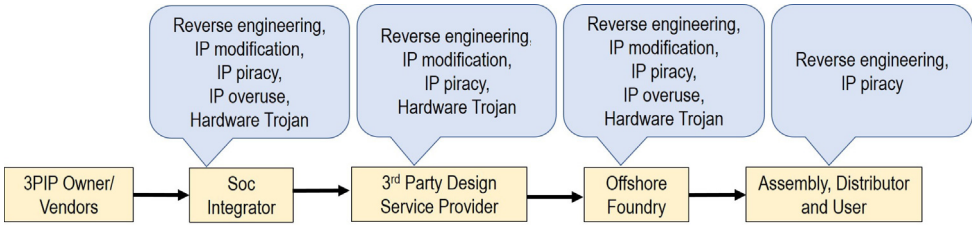


Fig. 10. Stake holders and corresponding IP threats in the horizontal supply chain.

equivalent to the RTL representation. Moreover, the gate-level netlist is also verified to check if the design meets timing, power, and area requirements. Thereafter, the SoC integrator integrates the DFT structure to enable the IC to be thoroughly tested during fabrication, package assembly, and in the field operation to ensure its correct functionality. Due to aggressive time-to-market, design houses may outsource some portion of the design, e.g., DFT insertion, physical layout design, to *third-party design service providers* and receive final GDS from them. In the past two decades, most design houses have become fabless. Therefore, they fabricate their products in *third-party offshore foundries*. In this process, the SoC design house can enjoy state-of-the-art fabrication technologies, however, at the cost of reduced trust in the manufacturing process (product integrity will be in doubt). After fabrication, the offshore foundry sends tested wafers to the assembly line to cut the wafers into die, and package the good ones to produce chips. After these processes are done, assembly performs structural tests to find defects in the chip that could be introduced during the assembly process. After performing these tests, the chips without defects are shipped to the distributors or the system integrator. The distributors sell these ICs in the market. With all these discussions we can summarize that IC design flow encompasses entities that design their own chips (fabless design houses), entities that offer design services to other firms (third-party design service providers or IP vendors), entities that offer fabrication facilities (offshore foundries), and entities that design and manufactures their chips in-house [18]. Different stakeholders in the supply chain have different motivations for IP infringements, therefore, introduce different vulnerabilities in the supply chain, as shown in Fig. 10.

6.2. Potential adversaries

The objective, assets, and capabilities available to an attacker influence the vulnerabilities that she might be interested to exploit. As shown in Fig. 10, the untrusted foundry, SoC integrator, third-party design service provider, and end-users can be identified as the potential antagonist against logic obfuscation.

6.2.1. Foundry

The combinational logic locking and FSM locking consider an offshore foundry as the primary source of threat in the supply chain [2,5,73]. Since the foundry has access to the GDS II file which they use to develop the costly mask for chip fabrication, an untrusted foundry is a major suspect for IP infringement. Besides, the attacker also can obtain an activated chip from the open market, a malicious insider is a trusted entity in the supply chain, or from a fielded system. The capability of each foundry also includes access to the state-of-the-art FA tools and reverse engineering capabilities. Access to DFT structure for detecting and analyzing the failure in the die is another asset available to the foundry. Access to aforementioned capabilities enables the foundry to reverse engineer the chip and localize the key-storage element, interconnect, key-delivery unit, key-gates, and DFT distribution to bypass the security of the obfuscated design. Consequently, the implementation of the circuit is crystal-clear to the foundry.

The objective of a rogue foundry is overbuilding and selling the chip in the open market. The adversary can also locate any specific IP from the design and learn about the implementation and functionality of that IP for hardware Trojan insertion or IP piracy. Depending on the objective of attack and obfuscation technique implemented in the design, a malevolent foundry can select its attack methodology. As the foundry can learn about the location of key-gates and key-delivery unit; applying FA methods like optical and electrical probing for extracting the key value of the key-gate is more convenient for the attacker [16,80]. However, foundry can perform black box analysis of structural obfuscated chip and exploit the Oracle-guided (for example, SAT, bypass, and SPS attacks) and Oracle-less (for instance SAIL, and desynthesize attacks) attacks. However, the success of Oracle-based and Oracle-less attacks is not always guaranteed.

Further, the foundry can deploy hardware Trojan for extracting the locking key. Fig. 11a summarized the assets and capabilities of a foundry and corresponding attack methodologies of an untrusted foundry.

6.2.2. SoC designer

An SoC designer has access to the soft/hard IP core, knowledge about the functionality of each IP, and unlocked functional obfuscated chip. Besides, the design undergoes extensive functional analysis for bug detection. Furthermore, a rogue designer may have access to DFT structures like the scan chain. The integrator also has the knowledge of synthesis tools. The capability of the SoC integrator can also include state-of-the-art FA tools and netlist reverse engineering software.

The primary intention of a malevolent SoC designer for attacking an obfuscated IP is IP piracy/theft. A rogue design house may report a less number of chips to the IP owner or clone the IP for selling it to other OCM. Hence, 3PIP vendors always have trust issues with the SoC integrator.

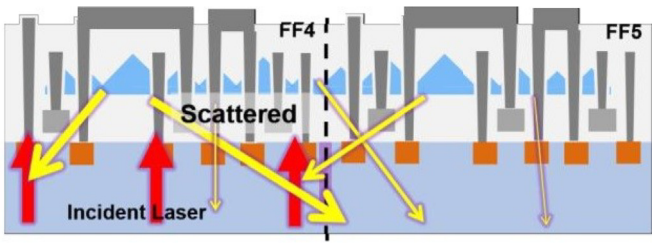
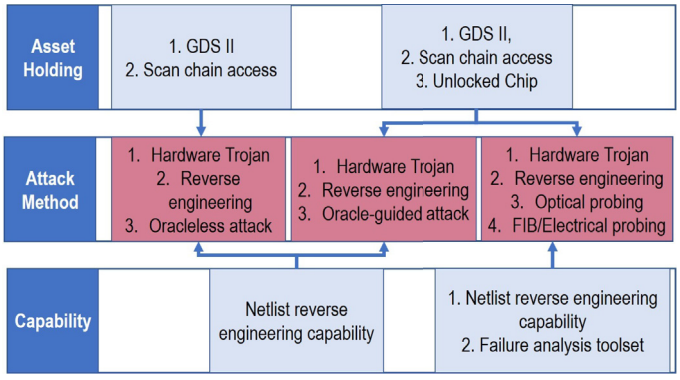


Fig. 12. Scattered reflection of incident laser beam in a nanopyramid implemented device [92].

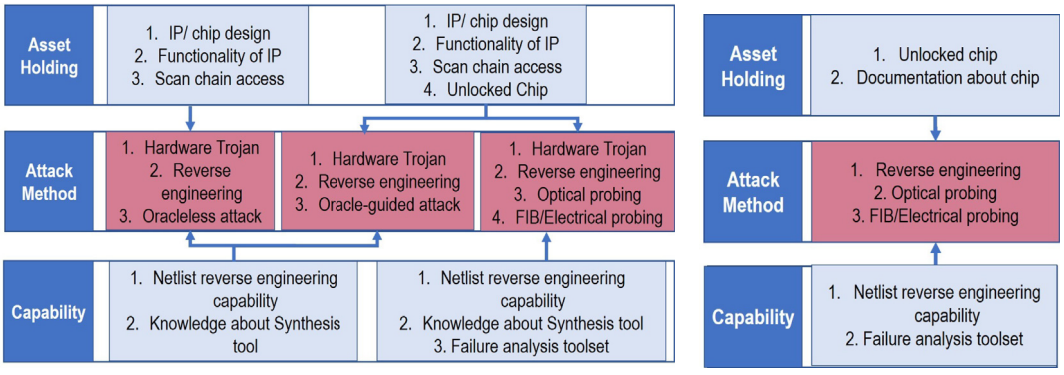
With the aforementioned assets, performing an hardware Trojan insertion, Oracle-guided and Oracle-less attacks on the chip is more convenient for an SoC designer. Aside from black-box analysis, a rough SoC integrator with access to reverse engineering and FA tools can also deploy physical attacks like optical probing.

6.2.3. 3rd party design service provider

Though in the horizontal supply chain, OCMs [17,81] outsource different design steps from 3rd party design service providers, the security threat imposed by the 3rd design service provider is still absent in most reported research in the literature. In a supply chain, a trusted SoC designer does not imply that the 3rd party design service provider is also trusted. As described in Sect. 6.1, in the current SoC design flow, the 3rd party design service provider has complete access to the gate-level netlist as well as the scan chain implemented in the device. Besides, the 3rd party design service provider can also gain access to an activated chip. Their capability may also include netlist reverse engineering and access



(a)



(b) (c)

Fig. 11. The threat model depending on asset and capability available to different untrusted entity in the supply chain – a) threat model for the untrusted foundry, b) threat model for the untrusted 3rd party design service provider and the SoC designer, c) threat model for the end user.

to FA lab. The goal for attacking the hardware obfuscation for a 3rd party service provider is hardware Trojan insertion, IP piracy, and IP overuse. Due to access to similar assets like SoC designer, exploiting Trojan, Oracle-guide and Oracle-less attacks is more convenient for 3rd party service provider. Furthermore, they can apply tools used for FA to extract the key value for logic obfuscation or FSM locking. The selection of attack methodologies depending on assets and capability for SoC designer and 3rd party design service provider is depicted in Fig. 11b.

6.2.4. End user

The threat of end user is the most overlooked concern in hardware obfuscation. The reason behind such an assumption is a common perception that full-blown reverse engineering is an expensive and expertise oriented process. In recent years, advancements in the reverse engineering process should compel the research community to revisit the threat of IP piracy by end users. An end user only has access to the unlocked chip and documentation related to that design. However, she can delayer each layer, image those layers with.

SEM and extract the gate-level netlist using reverse engineering software like Pix2Net or Chipwork. Even without having access to FA tools and reverse engineering capabilities, an end user without reverse engineering capability can still exploit the design vulnerabilities for extracting key value of FSM or logic locked circuitry using side-channel analysis and probing methods.

The potential adversaries for hardware obfuscation, their access to assets, their capabilities, and possible attack methods are summarized in Fig. 11. The possible access to capabilities and possible attack methods in Fig. 11 are ranked from the easiest to hardest.

7. Architecture for defense-in-depth

The objective of logic obfuscation is to protect the functionality and design implementation of the chip. The unlocking key in structural obfuscation is considered as the center of attacker interest and protecting the key is the objective of defense-in-depth. Therefore, the defense-in-depth layers are organized based on the threat model and vulnerabilities of the core components considered during the design stage. The capability of an attacker and asset availability to an attack also influence the organization of the defense layers. For example, probing attack is a possible approach for an attacker with access to FA tools. On the other hand, oracle-less attack is a possible approach for an adversary with access to GDSII. In addition, the dependency between the attacks needed to be considered during the layer organization, such as, during SAT attack, localizing the key-gates in the logic cones requires netlist reverse engineering of the design. Therefore, protection against reverse engineering must be placed before protection against SAT attack. Furthermore, a security designer must consider the fact that, an attacker can bypass the security implementation once the defense mechanism implemented in the device is exposed. Hence, failure in one defense layer may impact and even sacrifice the integrity of other defense layers. For example, success in structural reverse engineering allows a hacker to identify suitable point of interest (PoI) for probing and even expose the defense mechanism implemented against the electrical or optical probing attacks. Based on the above considerations, we have considered six layers of security for securing the key in hardware obfuscation as shown in Fig. 1.

Layer –1: Hardware Assurance. The security of the logic locking is established on the assumption that the hardware is secured. Any malicious modification detected in the design violates the assumption for root of trust, as well as impose dire threat towards the assets protected in the device. Hardware Trojan can also weaken the security mechanism implemented in the chip. Therefore, the objective of hardware assurance is to establish the root of trust of the device by evaluating the presence of hardware Trojan in the manufactured chip. The first step towards developing the defense-in-depth for logic locking ensures the root of trust before deploying the device in the filed.

Layer – 2: Defense against Reverse Engineering. Defense against reverse engineering, both structural and information, is considered as the second layer of defense for the obfuscated chip which is available to the end users. Attacking an obfuscated chip starts with breaking into the layout obfuscation techniques, learning the implementation of the design and detecting the point of interest for extracting the assets from the device. Although the cost, time, and expertise are always considered as a challenge for reverse engineering; once the completed reverse engineering attempt exposes valuable information to the adversary. An attacker can use that information for completing other attack methods like optical and electrical probing. Hence, protection against structural reverse engineering, increases the complexity of probing, and Oracle-guided attacks. Again, from the vulnerability analysis of key storage element shown in Fig. 9, it is also evident, extracting the key value through the information reverse engineering can be a straight forward task for breaking into the logic locking.

Layer – 3: Defense against Contactless Probing. Once, the adversary knows the location of the key-delivery unit and key-gates from layout reverse engineering, they can raid the key-delivery unit and interconnect layers using contactless method like optical probing from the backside of the chip (see Fig. 9). In Ref. [16], authors showed the location of key-delivery unit can also be extracted through partial reverse engineering. Due to non/semi-invasive nature of the optical probing, cost and time required for key extraction is much lower than contact-based electrical probing attack. The FA tools required for such analysis (laser microscope) can be rented for a few hundred dollars per hours. Nonetheless, a modern chip does not have any protection mechanism for the backside of the substrate. Therefore, protection against contactless probing has been placed in the third layer in defense-in-depth model.

Layer – 4: Defense against Contact-based Probing. Extracting key value from interconnects and key-delivery unit using FIB and electrical probing analysis involves invasive analysis. Similar to FA tools used for contactless probing; the tools required for contact-based probing can be rented almost at the same rate. However, due to the invasive nature of the attack, the time, cost and expertise required for electrical probing is considered higher than optical probing. Although several defence mechanisms have already been proposed, with access to right equipment an adversary can still bypass that defense mechanism. Hence, fourth layer in defense-in-depth should protect the chip assets from FIB/electrical probing (see Fig. 1).

Layer – 5: Defense for Design-for-Test. Literature showed that access to scan chain makes logic obfuscation vulnerable to several scan-based, Oracle-guided and Oracle-less attacks (See Fig. 9). However, the access is constrained to certain stakeholders which have been discussed in Sect. 6, hence, the protection of the scan chain is placed as the fifth layer in defense-in-depth.

Layer – 6: Defense for Logic Obfuscation Techniques. Lastly, logic obfuscation protects the functionality of the design. Attacking logic locking techniques requires reverse engineered gate-level netlist, i.e., success in breaking the second layer of defense in the obfuscated chip. Similar to scan chain attacks, logic locking can also be exploited using Oracle-guided or Oracle-less attack methods to learn the key value (see Fig. 9). As the presence of sequential logic poses difficulty against Oracle-guided attack, the defense for logic obfuscation is placed in the sixth layer of the defense-in-depth.

8. Security measures for defense-in-depth

In this section, we will discuss the security measures and future directions for developing defense-in-depth countermeasures for hardware obfuscation for major elements in chip design, i. e, the key-storage interconnect, key-delivery unit, DFT, and obfuscation techniques.

8.1. Hardware assurance

Detecting malicious modification in the design is the main objective

of hardware assurance layer in multi-layer defense approach. Several hardware Trojan detection techniques, e.g., run-time monitoring, test based approach, side-channel fingerprinting, have already been proposed to ensure the root of trust for the device [82]. However, none have proved to equally effective or limited due to golden chip requirement, time and memory consumption, process variation, subject matter expert involvement, etc.

Reverse engineering can be an effective means for verifying the trust and assurance of a chip fabricated in an untrusted foundry. However, the application of revers engineering is limited by the lack of automation and invasive nature of the method. The time and resources required for Trojan detection can be further reduced by applying computer vision and machine learning approach. In Ref. [83] authors suggested that, A fast SEM image collected from the backside thinned IC can be compared with the golden layout available to the designer for detecting potential malicious circuitry. In this case, Supervised machine learning and image processing is used to compare the DUA and golden layout. A security designer can also insert golden gate circuits (GCC) in the unused space of the design and use the GCC to improve the accuracy of machine learning classifier for detecting the any suspicious modification in the SoC [84]. The aforementioned techniques for hardware assurance can prevent the asset leakage like locking key. However, meeting the aggressive time-to-market requirement can still be a challenge for the OCM.

8.2. Defense against reverse engineering

The defense against the reverse engineering evolves around two core components in the obfuscated IC – key-storage and obfuscated hardware. Here, the protection mechanisms of those core components are reviewed.

8.2.1. Protecting key-storage from reverse engineering

Developing a secured key-storage device is still a topic for extensive research.

Over the past decades, researchers have proposed several methods to protect the NVM memory from reverse engineering. Memory encryption can be a solution against key-storage reverse engineering. In fact, memory encryption techniques may be the topic of most research activity aimed for protecting the data stored in main memory. Encryption algorithms allow strong diffusion characteristics that ensure a single bit change in the plaintext results in several bit changes in the cipertext. Therefore, an attacker can retain the key persists in the NVM, but in an unintelligible form. Although such encryption prevents reverse engineering, the designer should also consider the twofold of vulnerabilities introduced by the memory encryption. The decryption method would increase the decryption latency for key-storage which will adversely affect the performance of the chip through affecting the activation time required for the chip [85,86]. Again, the decryption key is also available in the chip which introduces the vulnerability with side-channel attack and introduces vulnerability for the key-delivery unit.

Anti-fuse technology is a promising solution as secure key-storage due to difficulty in localizing and reading the stored values in anti-fuse. This is a mature technology used in FPGAs and PLAs. Memory cell with different threshold voltage is also proposed as a possible key-storage cell. Using controlled process variations like dopant value, the threshold voltage of manufactured transistors can be varied from nominal values. Later the variation in threshold voltage is used to define the output from a logic cell [87]. Nonetheless, before using this method potential vulnerabilities against SEM, PVC, and other charge probing techniques should be addressed to block the reverse engineering of NVM.

Emerging NVM memory technologies can be considered as possible alternatives of the existing key-storage like Flash, EEPROM. Emerging memories – resistive random access memory (RRAM), spin-transfer torque magnetic random-access memory (STT-RAM), phase change memory (PCM) do not use the charge as storage media. For example, RRAM typically operates by electrical switching between different resistance states by applying high voltage, observed in several metal oxides [32]. Applying high voltage across the metal plates switches resistance states of the device. The high resistance state is considered as bit '1′ and the low resistance state is considered as bit '0'. As there is no visual difference between the bit '1′ and '0', it is difficult to extract the stored value from memory. Therefore, the aforementioned memory technologies are protected against the conventional charge probing techniques like SKPM, SCM, PVC. However, the susceptibility of the aforementioned memories against the side-channel analysis, or other types of probing (for example, EBIC/EBAC), or microscopy (for example, spin-SEM) should be evaluated.

8.2.2. Protecting obfuscated hardware from structural reverse engineering

Several countermeasures have been proposed to protect IC camouflaging against SAT, brute-force, and sensitization attacks. In Ref. [25], authors have proposed to perturb the functionality of the given design minimally by adding or removing one minterm. A camouflaged block, CamoFix, built up using camouflaged inverter/buffer cells, is used to restore the perturbed minterm in the functionality of the design. However, these techniques are vulnerable to removal attacks [7]. Researchers have also proposed to use layout-inclusive interconnect locking scheme based on cross-bars of metal-to-metal programmable -via devices. Logic locking scheme using antifuse to connect two adjacent metal layer proposed in Ref. [88], incorporated dummy vias and filler cells to eliminate the requirement for secure key storage. In addition, timing camouflaging can also increases the resiliency against reverse engineering in terms of functionality extraction [28,89].

Another solution for camouflaging the gate is to use different threshold voltage defined (TVD) logic gates [90]. The TVD logic gates are implemented with different threshold voltage transistors by varying the doping implant in the transistor. The gates have an identical layout, however, the threshold voltage defines the functionality of each gate. Covert gate is another variant for camouflaging cells [24]. Variation in doping concentration and dummy contacts are used to develop covert gates that are indistinguishable from regular logic gates in a design. Further, the gates shows higher resistance to SAT attack unless the location of the covert gate is identified.

The challenge of developing physical layout obfuscation technique is area, power, and delay overhead incorporated with the camouflage cells. Besides, developing threshold dependent camouflage cells involves dopant variation which can be identified from SEM imaging of the die at different beam voltages. Programming the TVD logic gates at the post-manufacturing stage has also been proposed as a possible camouflage technique [91]. The scalability of the TVD logic gates is always a concern for the semiconductor industry.

8.3. Defense against contactless probing

Security against optical analysis mostly concerns protecting the backside of the chip. Backside protection of the chip has received more attention recently from the security research community. The possible countermeasure for the backside of a chip can be divided into two levels – device and circuit level.

A security designer can add a backside polishing detector to monitor the thickness of the bulk silicon existing below the transistor. It has already been proposed as a countermeasure against mechanical polishing [93]. Adding an active opaque layer can be another countermeasure against optical probing. Implementing an active monitoring scheme is required to detect the removal of such opaque layer by an adversary [94]. Since the optical beam stimulates the silicon active regions thermally, conventional photosensors fail to trigger during optical probing. On the other hand, the thermal simulation introduces temperature and current variations in the circuit, which can influence circuits, such as ring-oscillators (ROs) [95]. In this case, the implementation of ROs as a probing protection scheme can be used to generate an antitamper reaction in the chip to protect the locking keys. In Ref. [92], nanopyramid structures (see Fig. 12) are implemented in selective areas inside the chip

to mitigate optical probing attacks by scattering the reflected laser beam, and consequently, scrambling the measurements of the register contents. Another proposed countermeasure is implementing a sandwiched metal shield between two polymers, opaque to NIR, at the back of the chip. As the layer can be removed using acid etching or polishing the chip, associating the stability of the bulk silicon to that sandwiched layer is required to prevent the adversary from taking off [96].

A circuit-level solution can be widely accepted for the semiconductor industry. As the logic locking key is static and embedded in the device memory, it can be probed by the aforementioned attacks. As a solution, the IP owner can use dummy active registers connected to functional gates to disguise the key-registers and eventually hide the key-gates. Although, the circuit level countermeasures might be known to a malicious foundry, and they can be easily deactivated, these countermeasures can be considered more secure against end users.

8.4. Defense against contact-based probing attacks

Active shield, which is also called digital shield, is the most common countermeasure against front-side probing attack [97,98]. In active shield technology, a signal carrying shield is placed on the top layer(s) of the chip to detect whether one of the shield wires is cut or not as shown in Fig. 13. A pattern generator is required for an active shield to generate flipping patterns to be transmitted on the shield. Then, a comparator at the end of the shield compares the received pattern from upper shield wires and another shield pattern copy from lower layers. If there is a mismatch detected at the comparator, which means at least one of the shield wires are cut during the attack, an alarm will be triggered, e.g. erasing all sensitive data stored in memory. The generated pattern should not be predicted or controlled by the attacker since if the shield patterns are compromised, then the attacker can synchronize the pattern at the end of the shield using fault injection techniques. Therefore, the shield wires before the fault injection sites are free to cut, which results in that the integrity checking function of the active shield is totally disabled. Although the active shield is very popular, its large design overhead and vulnerability to advanced FIB system limit the wide application of it [99]. First, a naive active shield on the top layer is very vulnerable to reroute and bypass attack [47] with advanced FIB system as illustrated in the previous subsection. Then, the active shield typically occupies one entire routing layer [100] which is prohibitively expensive to designs with tight cost margin and technologies with few routing layers. Further, the requirements for a non-predictable and non-controllable pattern generator determines that it is not a simple and small component, e.g. a cipher-based pattern generator with finite state machine (FSM) as its input [100], which introduces large area and power overhead to the design, especially when the design itself is relatively small, such as an AES or DES encryption core. In addition, the attacker can also utilize FIB's circuit editing capability [13] to manipulate the control circuit and payload of active shield to disable it.

Analog shield and sensors are alternative approaches to active shield [101,102]. Unlike active shield which detects the attack by comparing digital patterns, analog shield and sensors utilize analog features, e.g. capacitance or RC delay, at specific chip locations to detect the attack. One example is the Probe Attempt Detector (PAD) [101] as shown in Fig. 14. It detects the attack by measuring the additional capacitance introduced by the probe on selected sensitive wires. Compared to active shield which is covering a large chip area, the PAD approach is wire-oriented which is difficult to be applied to a large group of sensitive wires [101]. Therefore, if only a few wires are identified as security-critical wires and need to be protected, PAD is a good option with small overhead. Another example is charge sensor [103] which detects the attack by sensing charges during the FIB navigation process before the exact milling. An extremely sensitive local charge sensor is placed close to the chip surface, which could capture the charge changes and store the state for later read-out. However, the charge sensor accuracy is limited by the environmental noise and other power, voltage, and temperature (PVT) variations. Further, the charge sensor is not working in real-time, which leaves opportunity for attackers to neutralize the charge before the read-out of the stored state in charge sensor. In addition, one common and main limitation for all analog-based countermeasures, which typically requires a threshold value to trigger an alert, is that they are less reliable against process variation [13]. It is very difficult to distinguish between an attack and a reasonable process variation when the attacker's footprint is getting smaller and smaller with advanced equipment.

Different from active shield and analog sensor which are designed to detect the probing attack, *t*-private circuit [104] is proposed to deter the attack by exhausting the number of simultaneous probes in a probe station system which typically has 4–8 concurrent probes, so that the attacker doesn't have enough concurrent probes to extract one bit of

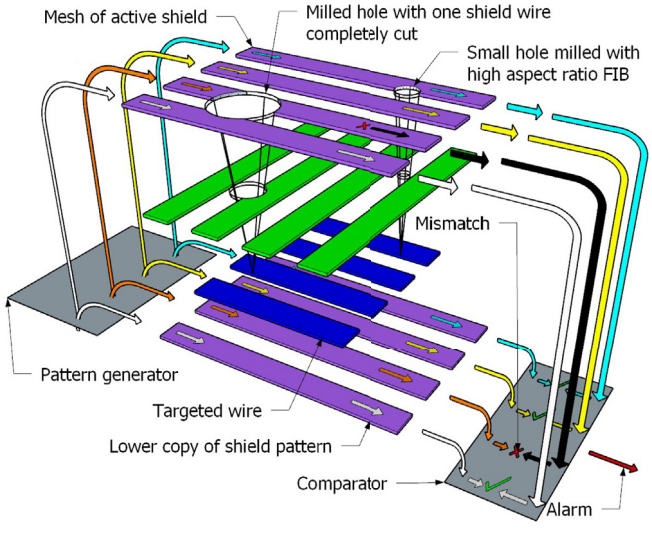


Fig. 13. Working principle of active shield and bypass attack on active shield.

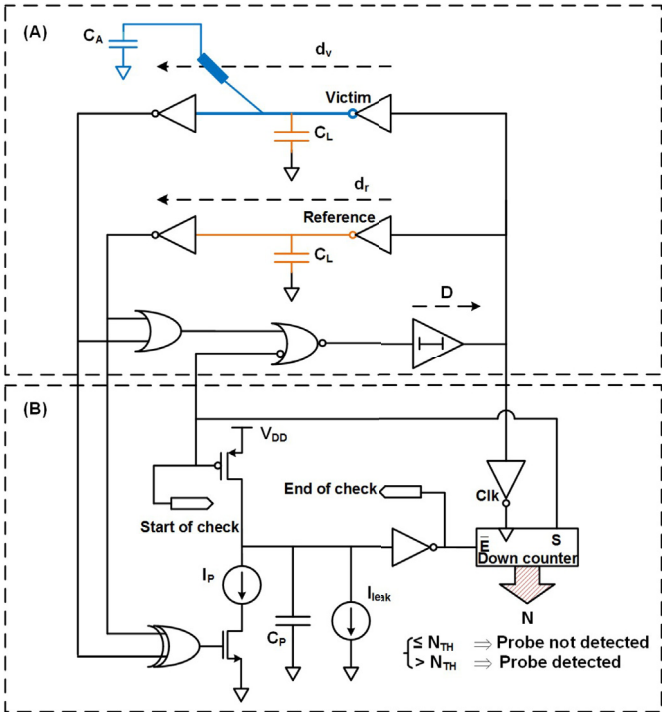


Fig. 14. Probe attempt detector (PAD).

information. Fig. 15 shows the diagram of *t*-private circuit which transforms the one bit signal *X* to m+1 bit signals (r_1,r_2, … r_{m+1}), so that at least m+1 probes are required within one clock cycle to extract one bit signal *X*. When m+1 exceeds the number of probes that the system provides, it would be very difficult for attackers to extract sensitive information through the probing attack. The main issue with *t*-private circuit is that the area overhead involved for transforming all signals in a chip is prohibitively expensive ($O(t^2)$) [99]. The scheme also requires a random bitstream generated at every clock cycle for the signal transformation.

To sum up existing countermeasures, we can find that every single solution is not efficient enough to resist probing attack and has its limitations [13]. So, we need a holistic and efficient solution against probing attack urgently because attacker's capability is always improving with advanced techniques. We believe that the following directions and suggestions are worth putting more effort to improve current countermeasures against the probing attack that can extract sensitive information from chip interconnects.

Security designers should keep in mind that a successful probing attack consists of many steps, such as navigation, milling, depositing, data extraction, etc [99]. Do not only focus on the milling step, like most shield-based countermeasures. If we can efficiently detect or deter two or more necessary steps in an attack, we could improve our protection performance and confidence to a great extent.

With the rapid improvement of the attacker's capability, especially for those attackers equipped with advanced FIB. FIB's capabilities, features, and limitations should be well modeled and considered in the countermeasure development. For example, FIB's aspect ratio, which is the ratio between depth and diameter, should be considered in a shield-based countermeasure [47]. It is because of the fact that the width and space of the shield wires and the depth difference between shield layer and probing target layer could determine if the shield is useless for a FIB system whose aspect ratio is larger than a specific value.

Almost all existing countermeasures have scalability issue with large overheads in chip area and layers and performance degradation [13], which is not acceptable for most high-end chips, e.g., CPU, that have a very limited budget for security. Therefore, a highly efficient solution is needed to protect those most sensitive nets in the design with minimal overhead.

In addition, there is no effective countermeasure against back-side probing occurring from the substrate of the chip which might be more threatening than front-side probing because it is much easier to get access to the transistors and sensitive nets on lower layers from the backside.

8.5. Defense for Design-for-testability

Several countermeasures have been proposed in the literature so far to thwart scan-based oracle-less and oracle-guided attacks. A brief discussion of existing techniques is given below.

Dynamically Obfuscated Scan (DOS): The authors in Ref. [105] proposed a design and test methodology against scan-based attacks throughout the supply chain, which includes a dynamically obfuscated scan for protecting IP/ICs. By perturbing test patterns/responses, and protecting the obfuscation key, the proposed architecture is proven to be robust against existing noninvasive scan-based attacks and can protect all scan data from attackers in the foundry, assembly, and system development, without compromising the testability. The key difference of this technique from other countermeasures is, rather than using a static obfuscation key, authors have proposed a dynamic obfuscation approach where the obfuscation key changes periodically based on the given permutation rate and initial seed of the LFSR [105], making the overall design resistant to scan-based side-channel attacks.

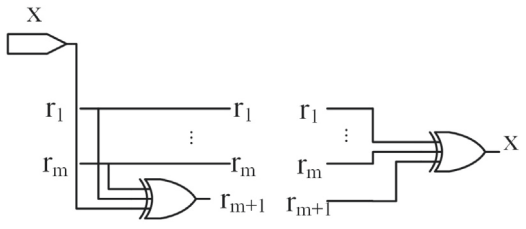


Fig. 15. Input encoder (left) and output decoder (right) for masking in t-private circuit.

Low-Cost Secure Scan (LCSS): In Ref. [106], authors have presented the low-cost secure scan (LCSS) solution. LCSS is implemented by inserting dummy flip-flops into the scan chains; it inserts the key into the test patterns concerning the position of the dummy flip-flops in the chains. By doing so, it verifies that all vectors scanned-in comes from an authorized user, and the correct response can be safely scanned-out after functional mode operation. If the correct key is not integrated into the vector, an unpredictable response is scanned-out, making analysis very difficult for an attacker. By using an unpredictable response, attackers would not be able to immediately realize that their intrusion has been detected, as could be discerned if the CUT were to immediately reset [106].

Lock and Key: Lock & Key solution was developed to neutralize the potential for scan-based side-channel attacks [70]. The Lock & Key technique provides a flexible security strategy to modern designs, without significant changes to the scan structure used in practice. Using this technique, the scan chains in an SoC are divided into smaller sub-chains. With the inclusion of a test security controller, the values of access to sub-chains are randomized when being accessed by an unauthorized user. Random access reduces repeatability and predictability, making reverse engineering more difficult. Without proper authorization, an attacker would need to unveil several layers of security before gaining proper access to the scan chain to exploit it.

Obfuscated Scan: Secure scan architecture using test key randomization (SSTKR) was developed to address security and testability issues [63]. Specifically, SSTKR is a key-based technique to prevent an attacker from illegally obtaining critical information while using scan infrastructure. The authentication keys are generated through a linear feedback shift register and inserted into test vectors. Furthermore, test keys are embedded into test vectors in two different ways: with dummy flip-flops and without dummy flip-flops. In the first case, dummy flip-flops holding the key are inserted into the scan chain to randomize scan outputs. It should be noted that all dummy flip-flops should not be connected to the combinational logic. In the second case, authentication keys are inserted into the positions of don't-care bits, generated by ATPG to reduce area overhead and test time.

Scan Encryption: A countermeasure against scan-based side-channel attacks could be done through the encryption of the scan chain content [59]. These attacks use an efficient and secure block cipher placed at each scan port to decrypt/encrypt scan patterns/responses at each scan input/output, respectively.

Scan-chain Reordering: A secure scan tree architecture is developed to protect cryptosystems against scan-based attacks [60]. This architecture offers low area overhead compared with the traditional scan tree architecture followed by a compactor, locking, and test access port (TAP) architecture. In contrast to the normal scan tree architecture, this architecture is based on the flipped scan tree (F-scan tree). To be exact, they adopt special flip-flops (that is, flipped FFs) in which inverter gates are added at the scan-in pin of scan flip-flop. The flipped scan tree architecture is built through normal SDFFs and flipped FFs. Since the attacker cannot identify the position of inverters, he/she is neither able to control the inputs, nor observe the outputs of the flip-flops.

8.6. Defense for Logic Obfuscation Techniques

Although logic obfuscation can be an effective mechanism for

establishing trust in the hardware design flow, it has not seen a widespread application in the semiconductor industry due to its lack of attack resiliency and formal notion of security. For example, most logic obfuscation techniques are vulnerable to SAT-based attacks [5].

To resist SAT attack, several SAT-resistant logic obfuscation techniques [11,12,71] can be implemented in the IP design. These SAT-resistant logic locking techniques that increase SAT attack complexity by increasing the number of required distinguishing input patterns (DIPs) [11,12] or by striping some of the functionality of the logic locked design [71] and hiding it in the form of a secret key, possess their several critical vulnerabilities. For example, researchers have proposed Bypass attack [73], SPS attack [7], App-SAT attack [6], and FALL attack [10] that can easily circumvent the effect of the SAT-resistant locking schemes. Further, these SAT-resistant schemes are known to possess low corruptibility, and thus do not provide the desired functional obfuscation. Hence, there remains a need for developing SAT-resistant logic obfuscation infrastructure. Since the SAT attack relies on access to the scan chain, effectively obfuscating/locking the scan chain to scramble scan-in and scan-out should help resist such attacks. A recently proposed scan architecture [105] resists bypass, reset, flushing and other scan-based attacks by dynamically obfuscating scan chain where scan chain obfuscation key changes periodically. This idea of dynamically changing the obfuscation key can also be utilized to resist the SAT attack. SAT attack requires access to unlocked IC (oracle), and locked netlist to rule out incorrect keys. If the obfuscation key can be changed each time before the SAT attack succeeds, then attack complexity would be drastically increased. Recently, a new programmable logic and routing blocks (PLRs) based logic obfuscation [89], similar but an upgraded approach to layout based interconnect locking [88], have been proposed as a possible solution against SAT and its derivative attacks.

Developing a key-gate insertion algorithm to improve the output corruptibility for wrong keys as well as thwart attacks similar to key-sensitization [8] can improve the defense mechanism of the logic locking. A fault analysis (FA) based key-gate insertion algorithm has already been proposed [29]. Increasing the dependency among the keys in the key-gate placement is also explored in a strong logic locking (SLL) algorithm [107]. However, all these algorithms are vulnerable to key-sensitization, or logic cone based [108] attacks. Moreover, the key-gate insertion algorithm can only be successful to protect the key value, if the chip is protected from reverse engineering and probing (both contactless and contact-based methods) attacks.

9. Research opportunities

There is no single silver bullet solution for addressing all the vulnerabilities in logic locking. Thus, multi-layer protection for the logic obfuscation is necessary to prevent an attacker from stealing the design secret. Although in this paper a possible framework for planning and selection of the defense layers has been laid, several other questions are yet to answer.

9.1. Selection of countermeasure

Critical challenges in developing a multi-layer defense mechanism are to select the appropriate countermeasure that can address the corresponding threat comprehensively. The security designer has to decide the countermeasures to implement for each defense layer.

Identifying the security metric and security rule check for defense layers can address the issue of countermeasure selection. For example, the designer can enumerate all known alternative safeguarding techniques for contact-based electrical probing technique and estimate the cost and time required for breaking into the defense layer using the metric developed. Furthermore, involving attacker capability is also necessary for developing a framework for the assessment of security metrics. Therefore, developing a framework to analyze the vulnerabilities and assessing the security of the design at all design stages can be a huge contribution in selecting the countermeasures for each defense layer.

9.2. Low overhead countermeasure

Another factor for defense-in-depth implementation is the allocation of security budget in terms of area, speed, power, and design cost for any specific embedded device. Such analysis enables the integration of the functional and countermeasure design in a holistic fashion. Moreover, the reliability of countermeasure is also dependent on the speed, power, and temperature variation such as sensor-based optical probing detector may not be able to detect low power laser beam if the security constraints are not selected properly. The sensor may ignore the local increase in temperature while optical probing is carried out at low laser power. Most of the time, the attack level and available security budget for a specific product are correlated. A high-end product with high IP value may be confronted with attackers with the most advanced equipment, and thus, may have more budget to adopt more countermeasures in the design. Therefore, when the IP value and the attacking threat of a product can be accurately estimated, the security designers can have more clues to determine which protection technique can be incorporated in the design.

9.3. Security metric for key-storage

Confidentiality of unlocking key-value significantly affects the security of structural obfuscated IP/chip. Therefore, a key-storage should have the following three properties:

(a) The key-storage must be read-proof, i.e., the malicious entity cannot reverse engineer or extract the key-value from the storage.
(b) The key-storage must be tamper-evident, i. e, it can detect tampering attempts and zeroized the content irrespective power status of the chip.
(c) The key-storage introduces lower area or power overhead to the chip to be an effective solution for chip.

Developing a framework for accessing the attack resiliency to different memory technology can be a contribution to the research community. In recent years, several emerging memory technologies have been proposed as a possible selection for secured key-storage. The performance of those memory technologies against known attack methods – invasive, non-invasive or semi-invasive methods is yet to be evaluated. Again, protecting the backside of NVM from unauthorized access can contribute to protecting the hardware obfuscation. Developing an active or tamper-evident shield to protect the memory can also be a significant advancement towards securing the key-storage. Developing a light key encryption algorithm can thwart exposing the key. Furthermore, several other questions needed to be answered like how to overcome the bottleneck due to the read latency for key-storage, erasing the residual data, and masking the location of OTP from advanced imaging tools.

9.4. Security of DFT structure

From the discussion in Sect. 5.4, it is apparent that none of the existing countermeasures can provide full protection against attacks that exploit scan infrastructure. For example, most countermeasures targeting scan-based side-channel attacks, do not consider protecting against IP Piracy, over-production, tampering and counterfeiting. The attacks in the first five-row of Table 1 target gaining access to DFT structures to leak security-critical information, and SAT attack targets logic locking circuit. In the case of a sequential circuit, the SAT attack requires access to DFT structures to divide the sequential design into smaller combinational designs that SAT solvers can handle. A combination of these attacks can be utilized to compromise the security of logic locked circuit. A dynamic scan chain obfuscation technique [65] has been suggested for protecting IPs against most of the scan-based attacks discussed above by

dynamically changing scan obfuscation key and scrambling scan-in patterns and scan-out responses. But this countermeasure does not consider the threat model of oracle-guided attacks e.g., SAT attack [5]. Hence, developing such countermeasure is necessary that can protect its secret against not only scan-based side-channel attacks abut also scan facilitated oracle-guided attacks.

9.5. Dynamic nature of security threat

The success of the defense-in-depth approach in protecting the locking key largely depends on the definition of the threat model. However, in a multi-layer approach, a designer can still overlook a backdoor and leave an attack vector accessible to the attacker. Therefore, developing a security architecture capable of addressing the dynamic nature of security threats can protect the IP after deploying it in the field. Developing SoC architecture, capable of hardware patching [34], can facilitate in implementing reconfigurable security policies. Consequently, it can address the issue of security vulnerability of on-field devices as well.

10. Conclusion

In this paper, we have presented a comprehensive study of different vulnerabilities of the core components, i.e., key-storage element, key-delivery unit, interconnect, DFT and structural obfuscation; in hardware obfuscation. Hardware obfuscation is emerging as a promising tool for protecting the IP/chip design and root-of-trust. Therefore, the dire threat imposed by the vulnerabilities of hardware obfuscation core components in an SoC can not thwart by a one-to-one protection scheme. Advancement in FA tools and algorithm-based attacks do not leave scope to consider any specific protection scheme as the ultimate preserver of the confidentiality and integrity of chip design. Through using multiple safeguard techniques to protect core components can defend the obscured chip from a variety of attacks. Therefore, this paper introduced the idea of a multi-layered defense mechanism that can ensure defense-in-depth for chip security. We have presented the contribution of each stakeholder in the supply chain of the semiconductor device. The outline for the comprehensive threat model is also presented considering the possible capabilities and assets available for all possible untrusted entities in the supply chain. Based on the above analysis we proposed a multilayer defense structure to establish the defense-in-depth in the IC. We also discussed the state-of-the-art defense mechanism for each layer and challenges for paving the path of the secured chip for developing a multi-layer protection scheme. Addressing the challenge of incorporating the multi-layer defense mechanism can be a significant advancement in the field of logic obfuscation.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.vlsi.2019.12.007.

References

[1] J.A. Roy, F. Koushanfar, I.L. Markov, Epic: ending piracy of integrated circuits, in: Proceedings of the Conference on Design, Automation and Test in Europe, ACM, 2008, pp. 1069–1074 (2008).
[2] R.S. Chakraborty, S. Bhunia, Harpoon: an obfuscation-based soc design methodology for hardware protection, IEEE Trans. Comput. Aided Des. Integr Circuits Syst. 28 (10) (2009) 1493–1502 (2009).
[3] J. Rajendran, M. Sam, O. Sinanoglu, R. Karri, Security analysis of integrated circuit camouflaging, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, ACM, 2013, pp. 709–720 (2013).
[4] K. Zamiri Azar, H. Mardani Kamali, H. Homayoun, A. Sasan, Threats on logic locking: a decade later, in: Proceedings of the 2019 on Great Lakes Symposium on VLSI, ACM, 2019, pp. 471–476 (2019).
[5] P. Subramanyan, S. Ray, S. Malik, Evaluating the security of logic encryption algorithms, in: 2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2015, pp. 137–143 (2015).
[6] K. Shamsi, M. Li, T. Meade, Z. Zhao, D.Z. Pan, Y. Jin, Appsat: approximately deobfuscating integrated circuits, in: Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on, IEEE, 2017, pp. 95–100 (2017).
[7] M. Yasin, B. Mazumdar, O. Sinanoglu, J. Rajendran, Removal attacks on logic locking and camouflaging techniques, IEEE Trans. Emerg. Topic. Comput. (2017).
[8] J. Rajendran, Y. Pino, O. Sinanoglu, R. Karri, Security analysis of logic obfuscation, in: Proceedings of the 49th Annual Design Automation Conference, ACM, 2012, pp. 83–89 (2012).
[9] P. Chakraborty, J. Cruz, S. Bhunia, Sail: machine learning guided structural analysis attack on hardware obfuscation, in: 2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), IEEE, 2018, pp. 56–61, 2018.
[10] D. Sirone, P. Subramanyan, Functional analysis attacks on logic locking, in: 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2019, pp. 936–939, 2019.
[11] Y. Xie, A. Srivastava, Anti-sat: mitigating sat attack on logic locking, in: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018.
[12] M. Yasin, B. Mazumdar, J. J. Rajendran, O. Sinanoglu, Sarlock: sat attack resistant logic locking, in: 2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, vol. 2016, pp. 236–241 (2016).
[13] H. Wang, D. Forte, M.M. Tehranipoor, Q. Shi, Probing attacks on integrated circuits: challenges and research opportunities, IEEE Design Test 34 (5) (2017) 63–71, https://doi.org/10.1109/MDAT.2017.2729398 (Oct 2017).
[14] C. Helfmeier, D. Nedospasov, C. Tarnovsky, J.S. Krissler, C. Boit, J.-P. Seifert, Breaking and entering through the silicon, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, vol. 2013, ACM, 2013, pp. 733–744.
[15] S. Tajik, H. Lohrke, J.-P. Seifert, C. Boit, On the power of optical contactless probing: attacking bitstream encryption of fpgas, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, vol. 2017, ACM, 2017, pp. 1661–1674.
[16] M.T. Rahman, S. Tajik, M.S. Rahman, M. Tehranipoor, N. Asadizanjani, The Key Is Left under the Mat: on the Inappropriate Security Assumption of Logic Locking Schemes, Report 2019/719, Cryptology ePrint Archive, 2019, https://eprint.iacr.org/2019/719.
[17] S. Newsroom, Samsung and esilicon taped out 14nm network processor with rambus 28g serdes solution.URL https://news.samsung.com/global/samsung-and-esilicon-taped-out-14nm-network-processor-with-rambus-28g-serdes-solution.
[18] D.B. Fuller, Chip design in China and India: multinationals, industry structure and development outcomes in the integrated circuit industry, Technol. Forecast. Soc. Chang. 81 (2014) 1–10 (2014).
[19] R.P. Cocchi, J.P. Baukus, L.W. Chow, B.J. Wang, Circuit camouflage integration for hardware ip protection, in: Proceedings of the 51st Annual Design Automation Conference, ACM, 2014, pp. 1–5 (2014).
[20] S. Amir, B. Shakya, D. Forte, M. Tehranipoor, S. Bhunia, Comparative analysis of hardware obfuscation for ip protection, in: Proceedings of the on Great Lakes Symposium on VLSI 2017, ACM, 2017, pp. 363–368, 2017.
[21] I. C. Society, Ieee recommended practice for encryption and management of electronic design intellectual property, https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7274481, accessed: 2017-09-30.
[22] G.T. Becker, F. Regazzoni, C. Paar, W.P. Burleson, Stealthy dopant-level hardware trojans: extended version, J. Cryptogr. Eng. 4 (1) (2014) 19–31 (2014).
[23] K. Shamsi, M. Li, T. Meade, Z. Zhao, D.Z. Pan, Y. Jin, Cyclic obfuscation for creating sat-unresolvable circuits, in: Proceedings of the on Great Lakes Symposium on VLSI 2017, ACM, 2017, pp. 173–178 (2017).
[24] B. Shakya, H. Shen, M. Tehranipoor, D. Forte, Covert gates: protecting integrated circuits with undetectable camouflaging, IACR Trans. Cryptogr. Hardware Embedd. Syst. (2019) 86–118 (2019).
[25] M. Yasin, B. Mazumdar, O. Sinanoglu, J. Rajendran, Camoperturb: secure ic camouflaging for minterm protection, in: 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, 2016, pp. 1–8 (2016).
[26] S. Bhunia, H. Shen, M. M. Tehranipoor, D. J. Forte, N. Asadizanjani, Vanishing via for hardware ip protection from reverse engineering, US Patent App. Vol. 15/863,133 (Jul. 12 2018).
[27] G.L. Zhang, B. Li, B. Yu, D.Z. Pan, U. Schlichtmann, Timingcamouflage: improving circuit security against counterfeiting by unconventional timing, in: 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2018, pp. 91–96 (2018).
[28] M. Alam, S. Ghosh, S.S. Hosur, Toic: timing obfuscated integrated circuits, in: Proceedings of the 2019 on Great Lakes Symposium on VLSI, ACM, 2019, pp. 105–110 (2019).
[29] J. Rajendran, H. Zhang, C. Zhang, G.S. Rose, Y. Pino, O. Sinanoglu, R. Karri, Fault analysis-based logic encryption, IEEE Trans. Comput. 64 (2) (2015) 410–424 (2015).
[30] A. Baumgarten, A. Tyagi, J. Zambreno, Preventing ic piracy using reconfigurable logic barriers, IEEE Design Test Comput. 27 (1) (2010).
[31] K. Shamsi, D.Z. Pan, Y. Jin, On the impossibility of approximation-resilient circuit locking, in: 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2019, pp. 161–170, 2019.
[32] Y. Xie, X. Xue, J. Yang, Y. Lin, Q. Zou, R. Huang, J. Wu, A logic resistive memory chip for embedded key storage with physical security, IEEE Trans. Circ. Syst. II: Express Briefs 63 (4) (2016) 336–340, 2016.
[33] S. Engels, M. Hoffmann, C. Paar, The end of logic locking? a critical view on the security of logic locking, IACR Cryptol. ePrint Arch. 2019 (2019) 796, 2019.

[34] A.P.D. Nath, S. Ray, A. Basak, S. Bhunia, System-on-chip security architecture and cad framework for hardware patch, in: Proceedings of the 23rd Asia and South Pacific Design Automation Conference vol. 2018, IEEE Press, 2018, pp. 733–738.
[35] A. Arm, Security Technology-Building a Secure System Using Trustzone Technology, ARM Technical White Paper, 2009.
[36] N.S.N.V., Realizing Today's Security Requirements: Achieving End-To-End Security with a Crossover Processor, Tech. rep., NXP, 2018, 09-30 (8 2017).
[37] A. Vijayakumar, V.C. Patil, D.E. Holcomb, C. Paar, S. Kundu, Physical design obfuscation of hardware: a comprehensive investigation of device and logic-level techniques, IEEE Trans. Inf. Forensics Secur. 12 (1) (2017) 64–77, 2017.
[38] N. Chen, The benefits of antifuse otp. URL https://semiengineering.com/the-benefits-of-antifuse-otp/.
[39] Synopsys, Antifuse-based split-channel 1t-fuse bit cell for otp nvm ip. URL https://www.synopsys.com/dw/ipdir.php?ds=nvm_1t-bit-cell.
[40] M. LAPEDUS, Nand market hits speed bump. URL https://semiengineering.com/nand-market-hits-speed-bump/.
[41] R. Druyer, L. Torres, P. Benoit, P.-V. Bonzom, P. Le-Quere, A survey on security features in modern fpgas, in: 2015 10th International Symposium on Reconfigurable Communication-Centric Systems-On-Chip (ReCoSoC), IEEE, 2015, pp. 1–8, 2015.
[42] C. De Nardi, R. Desplats, P. Perdu, F. Beaudoin, J.-L. Gauffier, Oxide charge measurements in eeprom devices, Microelectron. Reliab. 45 (9–11) (2005) 1514–1519, 2005.
[43] F. Courbon, S. Skorobogatov, C. Woods, Reverse engineering flash eeprom memories using scanning electron microscopy, in: International Conference on Smart Card Research and Advanced Applications, Springer, 2016, pp. 57–72, 2016.
[44] S. Tajik, H. Lohrke, F. Ganji, J.-P. Seifert, C. Boit, Laser fault attack on physically unclonable functions, in: 2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), IEEE, 2015, pp. 85–96, 2015.
[45] H. Lohrke, S. Tajik, T. Krachenfels, C. Boit, J.-P. Seifert, Key extraction using thermal laser stimulation, IACR Trans. Cryptogr. Hardware Embedd. Syst. (2018) 573–595, 2018.
[46] S. Skorobogatov, Hardware security implications of reliability, remanence, and recovery in embedded memory, J. Hardware Syst. Secur. 2 (4) (2018) 314–321, 2018.
[47] H. Wang, Q. Shi, D. Forte, M.M. Tehranipoor, Probing assessment framework and evaluation of antiprobing solutions, IEEE Trans. Very Large Scale Integr. Syst. (2019) 1–14, https://doi.org/10.1109/TVLSI.2019.2901449, 2019.
[48] J. Dworak, A. Crouch, A call to action: securing ieee 1687 and the need for an ieee test security standard, in: 2015 IEEE 33rd VLSI Test Symposium (VTS), IEEE, 2015, pp. 1–4, 2015.
[49] D. Mukhopadhyay, S. Banerjee, D. RoyChowdhury, B.B. Bhattacharya, Cryptoscan: a secured scan chain architecture, in: 14th Asian Test Symposium (ATS'05), IEEE, 2005, pp. 348–353 (2005).
[50] B. Yang, K. Wu, R. Karri, Secure scan: a design-for-test architecture for crypto chips, IEEE Trans. Comput. Aided Des. Integr Circuits Syst. 25 (10) (2006) 2287–2293 (2006).
[51] J.D. Rolt, G.D. Natale, M.-L. Flottes, B. Rouzeyre, A novel differential scan attack on advanced dft structures, ACM Trans. Des. Autom. Electron. Syst. 18 (4) (2013) 58 (2013).
[52] O. Kömmerling, M.G. Kuhn, Design principles for tamper-resistant smartcard processors, Smartcard 99 (1999) 9–20 (1999).
[53] D. Hely, F. Bancel, M.-L. Flottes, B. Rouzeyre, Secure scan techniques: a comparison, in: 12th IEEE International On-Line Testing Symposium (IOLTS'06), IEEE, 2006, p. 6 (2006).
[54] G.-M. Chiu, J.C.-M. Li, Ieee 1500 compatible secure test wrapper for embedded ip cores, in: 2008 IEEE International Test Conference, vol. 2008, IEEE, 2008, 1–1.
[55] S.M. Saeed, S.S. Ali, O. Sinanoglu, R. Karri, Test-mode-only scan attack and countermeasure for contemporary scan architectures, in: 2014 International Test Conference, vol. 2014, IEEE, 2014, pp. 1–8.
[56] C. Liu, Y. Huang, Effects of embedded decompression and compaction architectures on side-channel attack resistance, in: VTS, vol. 2007, 2007, pp. 461–468.
[57] J. Da Rolt, G. Di Natale, M.-L. Flottes, B. Rouzeyre, Are advanced dft structures sufficient for preventing scan-attacks?, in: 2012 IEEE 30th VLSI Test Symposium (VTS), vol. 2012 IEEE, 2012, pp. 246–251.
[58] J. DaRolt, G. Di Natale, M.-L. Flottes, B. Rouzeyre, Scan attacks and countermeasures in presence of scan response compactors, in: 2011 Sixteenth IEEE European Test Symposium, vol. 2011, IEEE, 2011, pp. 19–24.
[59] M. Da Silva, M.-l. Flottes, G. Di Natale, B. Rouzeyre, P. Prinetto, M. Restifo, Scan chain encryption for the test, diagnosis and debug of secure circuits, in: 2017 22nd IEEE European Test Symposium (ETS), vol. 2017, IEEE, 2017, pp. 1–6.
[60] G. Sengar, D. Mukhopadhyay, D.R. Chowdhury, Secured flipped scan-chain model for crypto-architecture, IEEE Trans. Comput. Aided Des. Integr Circuits Syst. 26 (11) (2007) 2080–2084, 2007.
[61] Y. Atobe, Y. Shi, M. Yanagisawa, N. Togawa, Dynamically changeable secure scan architecture against scan-based side channel attack, in: 2012 International SoC Design Conference (ISOCC), vol. 2012, IEEE, 2012, pp. 155–158.
[62] J. Lee, M. Tebranipoor, J. Plusquellic, A low-cost solution for protecting ips against scan-based side-channel attacks, in: 24th IEEE VLSI Test Symposium, IEEE, 2006, p. 6 (2006).
[63] M.A. Razzaq, V. Singh, A. Singh, Sstkr: secure and testable scan design through test key randomization, in: 2011 Asian Test Symposium, vol. 2011, IEEE, 2011, pp. 60–65.
[64] S. Paul, R.S. Chakraborty, S. Bhunia, Vim-scan: a low overhead scan design approach for protection of secret key in scan-based secure chips, in: 25th IEEE VLSI Test Symposium (VTS'07), IEEE, 2007, pp. 455–460 (2007).
[65] X. Wang, D. Zhang, M. He, D. Su, M. Tehranipoor, Secure scan and test using obfuscation throughout supply chain, IEEE Trans. Comput. Aided Des. Integr Circuits Syst. 37 (9) (2017) 1867–1880 (2017).
[66] D. Hely, M.-L. Flottes, F. Bancel, B. Rouzeyre, N. Berard, M. Renovell, Scan design and secure chip, in: IOLTS, vol. 4, 2004, pp. 219–224 (2004).
[67] J. Lee, M. Tehranipoor, C. Patel, J. Plusquellic, Securing designs against scan-based side-channel attacks, IEEE Trans. Dependable Secure Comput. 4 (4) (2007) 325–336 (2007).
[68] L. Azriel, R. Ginosar, A. Mendelson, Exploiting the Scan Side Channel for Reverse Engineering of a Vlsi Device, Technion, Israel Institute of Technology, vol. 897, Tech. Rep. CCIT Report, 2016.
[69] S.S. Ali, O. Sinanoglu, S.M. Saeed, R. Karri, New scan-based attack using only the test mode, in: 2013 IFIP/IEEE 21st International Conference on Very Large Scale Integration (VLSI-SoC), vol. 2013, IEEE, 2013, pp. 234–239.
[70] J. Lee, M. Tehranipoor, C. Patel, J. Plusquellic, Securing scan design using lock and key technique, in: 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05), vol. 2005, IEEE, 2005, pp. 51–62.
[71] M. Yasin, A. Sengupta, M.T. Nabeel, M. Ashraf, J.J. Rajendran, O. Sinanoglu, Provably-secure logic locking: from theory to practice, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, vol. 2017, ACM, 2017, pp. 1601–1618.
[72] S.M. Plaza, I.L. Markov, Protecting integrated circuits from piracy with test-aware logic locking, in: Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design, vol. 2014, IEEE Press, 2014, pp. 262–269.
[73] X. Xu, B. Shakya, M.M. Tehranipoor, D. Forte, Novel bypass attack and bdd-based tradeoff analysis against all known logic locking attacks, in: International Conference on Cryptographic Hardware and Embedded Systems, vol. 2017, Springer, 2017, pp. 189–210.
[74] M.E. Massad, J. Zhang, S. Garg, M.V. Tripunitara, Logic Locking for Secure Outsourced Chip Fabrication: A New Attack and Provably Secure Defense Mechanism, 2017 arXiv preprint arXiv:1703.10187.
[75] P. Chakraborty, J. Cruz, S. Bhunia, Surf: joint structural functional attack on logic locking, in: 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), vol. 2019, 2019, pp. 181–190.
[76] MicroNet Solutions, Inc.URL http://micronetsol.net/pix2net-software/.
[77] C. Yu, X. Zhang, D. Liu, M. Ciesielski, D. Holcomb, Incremental sat-based reverse engineering of camouflaged logic circuits, IEEE Trans. Comput. Aided Des. Integr Circuits Syst. 36 (10) (2017) 1647–1659 (2017).
[78] A. Jain, Z. Zhou, U. Guin, Taal: Tampering Attack on Any Key-Based Logic Locked Circuits, 2019 arXiv preprint arXiv:1909.07426.
[79] L. Lin, M. Kasper, T. Güneysu, C. Paar, W. Burleson, Trojan side-channels: lightweight hardware trojans through side-channel engineering, in: International Workshop on Cryptographic Hardware and Embedded Systems, vol. 2009, Springer, 2009, pp. 382–395.
[80] M.T. Rahman, Q. Shi, S. Tajik, H. Shen, D.L. Woodard, M. Tehranipoor, N. Asadizanjani, Physical inspection & attacks: new frontier in hardware security, in: 2018 IEEE 3rd International Verification and Security Workshop (IVSW), vol. 2018, IEEE, 2018, pp. 93–102.
[81] Macworld, Where are apple products made? URL https://www.macworld.co.uk/feature/apple/"https://www.macworld.co.uk/feature/apple/where-are-apple-products-made-3633832.
[82] M. Tehranipoor, F. Koushanfar, A survey of hardware trojan taxonomy and detection, IEEE Design Test comput. 27 (1) (2010).
[83] N. Vashistha, H. Lu, Q. Shi, M.T. Rahman, H. Shen, D.L. Woodard, N. Asadizanjani, M. Tehranipoor, Trojan scanner: detecting hardware trojans with rapid sem imaging combined with image processing and machine learning, in: ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis, ASM International, 2018, p. 256, 2018.
[84] Q. Shi, N. Vashistha, H. Lu, H. Shen, B. Tehranipoor, D.L. Woodard, N. Asadizanjani, 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), IEEE, 2019, pp. 61–71 (2019).
[85] J. Yang, L. Gao, Y. Zhang, Improving memory encryption performance in secure processors, IEEE Trans. Comput. 54 (5) (2005) 630–640 (2005).
[86] S. Chhabra, Y. Solihin, i-nvmm: a secure non-volatile main memory system with incremental encryption, in: 2011 38th Annual International Symposium on Computer Architecture (ISCA), IEEE, 2011, pp. 177–188 (2011).
[87] S. Keshavarz, D. Holcomb, Threshold-based obfuscated keys with quantifiable security against invasive readout, in: Proceedings of the 36th International Conference on Computer-Aided Design, IEEE Press, 2017, pp. 57–64 (2017).
[88] K. Shamsi, M. Li, D.Z. Pan, Y. Jin, Cross-lock: dense layout-level interconnect locking using cross-bar architectures, in: Proceedings of the 2018 on Great Lakes Symposium on VLSI, ACM, 2018, pp. 147–152 (2018).
[89] H.M. Kamali, K.Z. Azar, H. Homayoun, A. Sasan, Full-lock:Hard distributions of sat instances for obfuscating circuits using fully configurable logic and routing blocks, in: Proceedings of the 56th Annual Design Automation Conference 2019, ACM, 2019, p. 89 (2019).
[90] B. Erbagci, C. Erbagci, N.E.C. Akkaya, K. Mai, A secure camouflaged threshold voltage defined logic family, in: 2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), IEEE, 2016, pp. 229–235 (2016).
[91] N.E.C. Akkaya, B. Erbagci, K. Mai, A secure camouflaged logic family using post-manufacturing programming with a 3.6 ghz adder prototype in 65nm cmos at 1v

nominal v dd, in: 2018 IEEE International Solid-State Circuits Conference-(ISSCC), IEEE, 2018, pp. 128–130, 2018.

[92] H. Shen, N. Asadizanjani, M. Tehranipoor, D. Forte, Nanopyramid: an optical scrambler against backside probing attacks, in: ISTFA 2018: Proceedings from the 44th International Symposium for Testing and Failure Analysis, ASM International, 2018, p. 280 (2018).

[93] S. Manich Bou, D. Arumi Delgado, R. Rodríguez Montañés, J. Mujal Colell, D. Hernández García, Backside polishing detector: a new protection against backside attacks, in: DCIS'15-XXX Conference on Design of Circuits and Integrated Systems, 2015, 2015.

[94] E. Amini, A. Beyreuther, N. Herfurth, A. Steigert, B. Szyszka, C. Boit, Assessment of a chip backside protection, J. Hardware Syst. Secur. 2 (4) (2018) 345–352 (2018).

[95] S. Tajik, J. Fietkau, H. Lohrke, J.-P. Seifert, C. Boit, Pufmon: security monitoring of fpgas using physically unclonable functions, in: 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS), vol. 2017, IEEE, 2017, pp. 186–191.

[96] S. Borel, L. Duperrex, E. Deschaseaux, J. Charbonnier, J. Cledi`ere, R. Wacquez, J. Fournier, J.-C. Souriau, G. Simon, A. Merle, A novel structure for backside protection against physical attacks on secure chips or sip, in: 2018 IEEE 68th Electronic Components and Technology Conference (ECTC), vol. 2018, IEEE, 2018, pp. 515–520.

[97] S. Briais, J. Cioranesco, J. Danger, S. Guilley, D. Naccache, T. Porteboeuf, Random active shield, in: 2012 Workshop on Fault Diagnosis and Tolerance in Cryptography, vol. 2012, Sep. 2012, pp. 103–113, https://doi.org/10.1109/FDTC.2012.11.

[98] X.T. Ngo, J. Danger, S. Guilley, T. Graba, Y. Mathieu, Z. Najm, S. Bhasin, Cryptographically secure shield for security ips protection, IEEE Trans. Comput. 66 (2) (2017) 354–360, https://doi.org/10.1109/TC.2016.2584041. Feb 2017.

[99] S. Bhunia, M. Tehranipoor, Hardware Security: A Hands-On Learning Approach, Ch. 10, Elsevier Science, 2018 (2018).

[100] J. Cioranesco, J. Danger, T. Graba, S. Guilley, Y. Mathieu, D. Naccache, X.T. Ngo, Cryptographically secure shields, in: 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), vol. 2014, May 2014, pp. 25–31, https://doi.org/10.1109/HST.2014.6855563.

[101] S. Manich, M.S. Wamser, G. Sigl, Detection of probing attempts in secure ics, in: 2012 IEEE International Symposium on Hardware-Oriented Security and Trust, vol. 2012, June 2012, pp. 134–139, https://doi.org/10.1109/HST.2012.6224333.

[102] M. Weiner, S. Manich, R. Rodŕiguez-Montañés, G. Sigl, The low area probing detector as a countermeasure against invasive attacks, IEEE Trans. Very Large Scale Integr. Syst. 26 (2) (2018) 392–403, https://doi.org/10.1109/TVLSI.2017.2762630. Feb 2018.

[103] C. Helfmeier, C. Boit, U. Kerst, On charge sensors for fib attack detection, in: IEEE International Symposium on Hardware-Oriented Security and Trust, 2012, 2012, pp. 128–133, https://doi.org/10.1109/HST.2012.6224332. June 2012.

[104] Y. Ishai, A. Sahai, D. Wagner, Private circuits: securing hardware against probing attacks, in: D. Boneh (Ed.), Advances in Cryptology - CRYPTO 2003, vol. 2003, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 463–481.

[105] X. Wang, D. Zhang, M. He, D. Su, M. Tehranipoor, Secure scan and test using obfuscation throughout supply chain, IEEE Trans. Comput. Aided Des. Integr Circuits Syst. 37 (9) (2018) 1867–1880, 2018.

[106] D. Hely, F. Bancel, M.-L. Flottes, B. Rouzeyre, Test control for secure scan designs, in: European Test Symposium (ETS'05), vol. 2005, IEEE, 2005, pp. 190–195.

[107] M. Yasin, J.J. Rajendran, O. Sinanoglu, R. Karri, On improving the security of logic locking, IEEE Trans. Comput. Aided Des. Integr Circuits Syst. 35 (9) (2016) 1411–1424 (2016).

[108] Y.-W. Lee, N.A. Touba, Improving logic obfuscation via logic cone analysis, in: 2015 16th Latin-American Test Symposium (LATS), vol. 2015, IEEE, 2015, pp. 1–6.