

Hardware Security: Threat Models and Metrics

M. Rostami and F. Koushanfar
Rice University
Houston, TX, USA
{masoud, farinaz}@rice.edu

J. Rajendran and R. Karri
Polytechnic Institute of NYU
Brooklyn, NY, USA, 11201
jv.ece@nyu.edu, rkarri@poly.edu

ABSTRACT

The globalized semiconductor supply chain is vulnerable to hardware attacks including: Trojans, piracy of intellectual properties (IPs) and/or overbuilding of integrated circuits (ICs), reverse engineering, side-channels, and counterfeiting. In this paper, we explain the threat models, the state-of-the-art defenses, and the metrics used to evaluate the defenses. The threat models outlined in this paper enables one to understand the attacks. Defenses and metrics can help defenders to build stronger countermeasures and evaluate them against other protection techniques using the metrics.

Keywords

Hardware Trojans, Reverse Engineering, IP/IC Piracy, Side-Channel Attacks, Counterfeiting, Camouflaging

1. INTRODUCTION

The semiconductor supply chain shown in Figure 1 is distributed worldwide [1, 2]. The figure shows SoC design flow and system design. Designing an SoC involves procuring intellectual property designs (IPs) from outside design houses, designing in-house components, combining them, and generating the layout through several synthesis and verification steps. The foundry manufactures the integrated circuits (ICs), which are then tested. Fault-free ICs are then packaged and sold.

This semiconductor supply chain is vulnerable to the following attacks. Rogue elements may insert malicious circuits (*a.k.a.*, *hardware Trojans*) into the design [1]. An attacker may steal and claim ownership of the IP, resulting in *IP piracy*. An untrusted foundry may *overbuild ICs* and sell them illegally [3]. An attacker can *reverse engineer* the functionality of an IC/IP [4]. Furthermore, *side-channels* such as power and timing information can be used to compromise hardware implementation (for example, leaking secret keys of cryptographic algorithm implementations [5]). During system design, as shown in Figure 1 (shaded region on the right hand side), faulty, low-grade ICs can pollute the supply chain. In addition, ICs from outdated systems may be recycled and used into the target system. This is called *counterfeiting*.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD 32, November 18 - November 21 2013, San Jose, CA, USA.
Copyright 2013 ACM 978-1-4799-1071-7/13/ \$31.00.

This paper surveys these hardware attacks: Trojans (Section 2), IP piracy/IC overbuilding (Section 3), reverse engineering (Section 4), side-channels (Section 5), and counterfeiting (Section 6). For each attack, we explain the threat model, the state-of-the-art defenses, and the metrics used to evaluate the defenses. Section 7 concludes the paper.

2. HARDWARE TROJANS





	3PIP Vendor	SoC Integrator	Foundry	User
1	?			?
2			?	?

Figure 2: Two hardware Trojan attack scenarios: (i) foundry and (ii) 3PIP vendor. The devil depicts an attacker, the shield represents a defender, and "?" indicates an untrustworthy entity.

A Hardware Trojan is a malicious modification to a circuit by an attacker. A Trojan can control, modify, disable, or monitor the contents and communications of the circuit [6, 7, 8].

2.1 Threat models

Figure 2 illustrates the two hardware Trojan attack scenarios. In the first scenario, an attacker in the foundry inserts a Trojan into the design. These Trojans may be in the form of addition, deletion or modification of gates.

In the second scenario, a malicious IP is designed either by a rogue in the third party IP (3PIP) design house [8, 7, 9] or by a rogue in the in-house design team [10, 11, 12]. Since the malicious insider may not provide Trojan-related information, the validation team may not be able to detect them.

2.2 State-of-the-art defenses

Most detection techniques target Trojans inserted in the foundry. They are based on functional or structural tests such as IDDT [13, 14], path-delay measurements [15], gate-level characterization [16, 17], thermal profiling [18], or a combination of them [6, 18].

Defenses against malicious 3PIP and insider attacks include runtime self monitoring [12] and static verification [11]. Trojans are also prevented from activation by breaking the sequence/timing of events and by scrambling inputs supplied to the units [10]. SoC integrator and the 3PIP vendor can agree on a set of security properties which the integrator can verify [9].

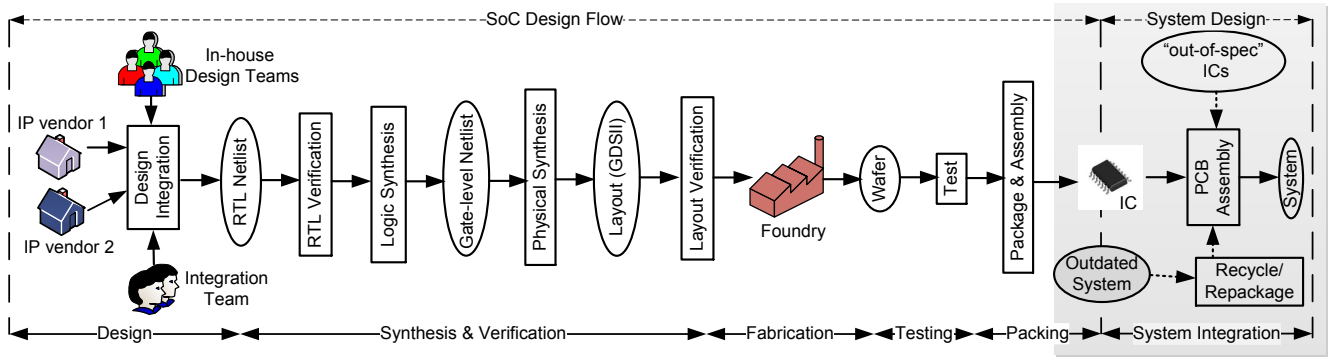


Figure 1: Semiconductor supply chain has two parts: (1) System-on-Chip design flow (only the steps and entities that relevant to this paper are shown) and (2) System design (the dotted lines represent how the fake and low-quality components enter the supply chain).

	3PIP Vendor	SoC Integrator	Foundry	User
1	O+W +F+M	👹	?	?
2	O+W +F+M	?	👹	?
3	?	O+W +F+M	👹	?

Figure 3: Threat scenarios for IP piracy and IC overbuilding. Obfuscation (O), Watermarking (W), Fingerprinting (F) and metering (M) are the defenses.

2.3 Metrics

(i) **Probability of detection**: the ratio of the number of Trojans detected by the technique to the total number of Trojans in the design. (ii) **Probability of false alarm**: the ratio of the number of Trojan-free designs that are incorrectly classified as Trojan to the number of Trojan-free designs. (iii) **Number of required clock cycles** to detect a Trojan in a 3PIPs scenario.

3. IP PIRACY AND IC OVERBUILDING

An attacker with access to IP or an IC can steal and claim ownership of it and/or can overbuild and sell them illegally [3, 19].

3.1 Threat models

Figure 3 illustrates the threat model for piracy and overbuilding. In scenario 1, the attacker in the SoC integration house can pirate the 3PIP or use more than the licensed number of 3PIP instances. In scenario 2, the attacker in the foundry can pirate the 3PIP after extracting it from the layout of the design. In scenario 3, the attacker in the foundry can pirate the IC design and/or overbuild.

3.2 State-of-the-art defenses

Piracy and overbuilding can be prevented by obfuscation, watermarking, fingerprinting, and metering. In scenarios 1 and 2, the 3PIP vendor can protect his IP by obfuscating it, or by embedding his watermark or fingerprint it. In scenario 3, the SoC integrator can obfuscate or embed his watermark or fingerprinting the design before delivering it to the foundry.

Obfuscation hides the functionality and implementation of a design by inserting additional gates into it. When a wrong value is applied to these gates, they modify the functionality of the design. In one type of obfuscation, additional (black) states are introduced in the finite state machine (FSM) [19, 20]. The FSM is modified in such a way that the design reaches a valid state only on applying

the correct key. If the key is withdrawn, the design ends up in a black state. In another type of obfuscation, XOR/XNOR gates [3, 21] and memory elements [22] are added. The obfuscated design will function correctly only on applying the correct value to these gates and memory elements.

Watermarking is done by embedding a designer's signature in the design [23]. The designer can reveal the watermark and claim ownership of an IC/IP. Watermarks are constructed by adding black states to the FSM, and secret constraints during high-level [24], logical, and physical synthesis [25] and during FPGA design [26].

Fingerprinting helps the defender to track down the source of piracy by embedding the buyer's signature (for instance, his public key) along with the designer's watermark [27]. When challenged, the designer can reveal the watermark to claim the ownership and the buyer's signature to reveal the source of piracy. For example, the power, timing, or thermal fingerprint of an IC is revealed on applying a set of input vectors. Fingerprinting can be also be applied during high-level, logical, and physical synthesis [27]. Another possibility is to use fingerprints from an IC's SRAM [28]. Some fingerprinting techniques use physical unclonable functions (PUFs). PUFs are circuit structures that extract a unique set of fingerprints from each IC [29].

Metering is a set of tools, methodologies, and protocols used to track the manufactured IC. In passive metering, part of the functionality is used for metering, even for the ICs manufactured from the same mask [30]. The identified ICs may be matched against their record in a database. This will reveal unregistered ICs or overbuilt ICs. In active metering, parts of the IC's functionality can be only accessed, locked, or unlocked by the designer and/or IP rights owners [20].

3.3 Metrics

Obfuscation: Metrics for obfuscation include: (i) Number of brute force attempts required to unlock the FSM or to determine the secret key [20, 21]. (ii) Hamming distance between the outputs of an obfuscated netlist on applying an incorrect key (or configuration) and the original netlist [22, 31]. (iii) Number of input patterns that produce an incorrect output on applying an incorrect key to the design [19].

Watermarking [24]: Metrics for watermarking include: (i) Probability of a watermarking algorithm generating the same solution for different buyers' signatures. (ii) Probability of an attacker changing one or more watermarking bits by modifying the design.

Fingerprinting: Metrics for fingerprinting include: (i) Average Hamming distance between the responses to the same challenge obtained from two different ICs. (ii) Average Hamming dis-







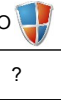











	3PIP Vendor	SoC Integrator	Foundry	User
1	O 		?	?
2	O 	?		?
3	?	O 		?
4	O 	?	?	
5	?	O 	?	
6	?	?	C 	
7	O 	?	C 	
8	?	O 	C 	

Figure 4: Threat scenarios for reverse engineering. Obfuscation (O) and camouflaging (C) are the defenses.

tance between the responses to the same challenge (or a repeatedly measured fingerprint) applied at different times and environmental conditions to the same IC. (iii) Non-digital measures of distances (i)/(ii). and (iv) Number of independent IDs that can be generated.

4. REVERSE ENGINEERING

Reverse engineering (RE) of an IC involves (i) identifying the device technology used [32], (ii) extracting its gate-level netlist [4], and/or (iii) inferring the implemented functionality [33]. Techniques and tools have been developed to reverse engineer¹ ICs [34, 35]. RE can be misused to steal and/or pirate a design, to identify the device technology, and to illegally fabricate the target IC.

4.1 Threat models

Figure 4 illustrates the threat models for RE. In scenario 1, the attacker in the SoC integration house can reverse engineer the 3PIP. The functionality of 3PIP modules can be extracted by behavioral matching against a library of known components [36], or by performing Boolean satisfiability analysis against a library of components [37]. In scenario 2, the attacker in the foundry can extract the 3PIP from the layout of the IC. In scenario 3, the attacker in the foundry can reverse engineer the IC. He can extract the transistor-level netlist from the layout [35], and then the gate-level netlist from it [38]. In scenarios 4–8, the user performs reverse engineering. He may depackage the IC, delayer it, image the layers, stitch those images, and extract the netlist.

4.2 State-of-the-art defenses

Obfuscation (see Section 3.2) and camouflaging are two defenses that have been proposed to thwart RE. In scenarios 1, 2, 4 and 7, a 3PIP vendor can obfuscate his IP. In scenarios 3, 5 and 6, an SoC integrator can obfuscate his design. A trusted foundry can camouflage the layout (scenarios 6–8), providing an additional layer of defense beyond obfuscation.

Camouflaging is a layout-level technique that hampers image processing-based extraction of a gate-level netlist from an IC. In one embodiment of camouflaging, the layouts of standard cells are designed to look alike, resulting in incorrect extraction of the

¹These tools enable reverse engineering to collect competitive intelligence, to verify a design, to check for commercial piracy, to determine patent infringements, and to detect hardware Trojans [34, 35, 33].

netlist. IC camouflaging can leverage unused spaces in an IC by filling them with filler cells [39], can use programmable standard cells [40], or can use dummy contacts [41].

4.3 Metrics

Metrics for obfuscation are given in Section 3.3.

Reverse engineering: Metrics for reverse engineering include:

(i) Percentage of gates correctly extracted from a layout [4]. (ii) Percentage of gates whose functionality is correctly inferred [37]. (iii) Number of signals correctly matched between the signals in the component with known functionality and the signals in the target design [36].

Camouflaging: Metrics for camouflaging include: (i) Number of brute force attempts required to identify the functionality of camouflaged gates [42, 43]. (ii) Hamming distance between the outputs of the original netlist and the netlist in which the functionality of camouflaged gates are assigned by the attacker [42].

5. SIDE-CHANNEL ATTACKS

Side-channel attacks exploit the leakage of physical information when an application is being executed on a system [5]. Side-channel attacks are powerful and have broken all major cryptographic algorithms [44].

5.1 Threat models

Information about secret keys can leak from an IC through its power consumption traces [5], timing traces [45], electromagnetic emanations [46], photonic emissions [47], scan chains [48], and faults injected in the designs [49]. Information leaked from side-channels do not completely overlap with each other. Hence, an adversary can combine the information leaked from several side-channels to increase the effectiveness of the attack [44].

5.2 State-of-the-art defenses

Leakage of information can be reduced by either decreasing the dependency between the side-channel trace and secret key, or by injecting noise into the side-channel [50]. The dependency between power consumption and secret key can be reduced by using dynamic and differential logic [51], asynchronous logic [52], and dual-rail with pre-charge logic [53].

Noise injected into the side-channels by adding dummy circuits that consume random amount of power for each transaction, or by performing random operation independent of the secret keys can reduce information leakage [54].

Though leakage reduction and noise injection do not provide theoretical security, they increase the effort of an attacker to extract the secret keys. For instance, decreasing the SNR of the side-channel information by a factor of K linearly or quadratically increases the number of required input patterns for side-channel analysis [55].

Key update prevents the accumulation of side-channel information by regularly updating the secret key after a pre-determined number of input patterns [56].

Leakage-resilient cryptography entails designing cryptographic primitives that are intrinsically resilient to information leakage through side-channels [57, 58, 59].

5.3 Metrics

Metrics for defenses against side-channel attacks include: (i) Number of input patterns required by an attacker to retrieve the secret key. (ii) Maximum amount of information leaked per input pattern [55]. (iii) Correlation between the secret key and the side-channel trace per input pattern [60].

6. COUNTERFEITING

A counterfeit semiconductor component is an illegal forgery or imitation of the original component². Counterfeiting is often performed by one of the many entities in the semiconductor supply chain, including new product vendors or secondary (recycled) IC vendors. IC counterfeiters profit by selling a cheaper and low quality IC. Although the primary incentive for selling fake ICs is financial, the ease of inserting hardware Trojans or spyware in fake ICs makes them a real security threat for the system that contains them [61].

6.1 Threat models







	Design	Test	Re-packaging/ Recycling	PCB Assembly
1	?		?	
2		?		?
3	?	?		

Figure 5: Threat scenarios for counterfeiting.

Figure 5 illustrates the counterfeit IC threat models. In scenario 1, defective ICs, i.e., those which failed the manufacture-time testing and have been discarded, are used in consumer products [61]. An untrustworthy entity at the test facility can be the source of leaking defective ICs. In scenarios 2 and 3, a dishonest entity in the IC supply chain mislabels a product and sells it as another IC potentially through a vendor [61]. The functionality of the mislabeled IC is likely not the same as the intended IC specification. In addition, used and recycled ICs are repackaged as new [62, 63]. The attackers are the second-hand vendors who buy or collect old electronic systems and remove ICs from them. The extracted ICs are repackaged and sold as new, in particular as spare parts for older electronic systems which are out of production line.

6.2 State-of-the-art defenses

In scenario 1, the faulty and low-grade chips can be detected by re-testing them before deploying into a system. In scenario 2, proactive techniques like hardware metering, fingerprinting, watermarking (Section 3.2), and sensors to determine IC aging are the used at the design phase to enable counterfeit detection [64, 61, 65]. Mislabeled chips can be detected by visual inspection, depackaging, or X-ray photography of the packages [65]. In scenario 3, using non-invasive measurements such as power and timing, one can determine an IC's age or reliability, thereby potentially detecting used/old/recycled ICs [64, 62, 63]. One can also implement aging sensors to report the amount of usage (or age) of the ICs which include them.

6.3 Metrics

The metrics for hardware metering, fingerprinting, watermarking are discussed in Section 3.3. Metrics for counterfeit detection using **circuit-aging** include: (i) Probability of detection is the ratio of the number of counterfeit ICs detected by the technique to the total number of counterfeit ICs [62, 63]. (ii) Probability of false alarm is

²Note that we make a distinction between the pirated/overbuilt ICs and fake ICs (although a clear distinction may be blurry in certain scenarios). IC piracy and overbuilding entail making ICs by illegally copying or stealing an authentic blueprint/IC during one of the design, synthesis, or production phases.

the ratio of the number of genuine ICs that are incorrectly classified as counterfeit ICs to the number of genuine ICs [64, 62].

7. CONCLUSIONS

Threat models, the state-of-the-art countermeasures, and the metrics used to evaluate the defenses against Hardware Trojans, IC and IP piracy, reverse engineering, side-channels, and counterfeiting were introduced. Until now, most evaluations of defenses have been informal and anecdotal. The authors believe that the metrics are an important first step in formalizing the evaluation of the strengths of defenses. Similarly, a consistent classification of threat models was not available. By organizing the threat/defense scenarios, we hope the countermeasures can be compared against one another based on the target threat model and the corresponding metrics.

8. ACKNOWLEDGMENTS

This research was supported in parts by an Office of Naval Research grant (ONR R17460) and a NSF grants to Rice University (CNS-1059416) and NYU-Poly (CNS-1059328).

9. REFERENCES

- [1] "Defense Science Board (DSB) study on High Performance Microchip Supply," www.acq.osd.mil/dsb/reports/ADA435563.pdf, 2005.
- [2] SEMI, "Innovation is at risk as semiconductor equipment and materials industry loses up to \$4 billion annually due to IP infringement," www.semi.org/en/Press/P043775, 2008.
- [3] J. Roy, F. Koushanfar, and I. Markov, "EPIC: Ending Piracy of Integrated Circuits," *IEEE Computer*, vol. 43, no. 10, pp. 30–38, 2010.
- [4] R. Torrance and D. James, "The state-of-the-art in semiconductor reverse engineering," *IEEE/ACM Design Automation Conference*, pp. 333–338, 2011.
- [5] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," *Advances in Cryptology*, pp. 388–397, 1999.
- [6] F. Koushanfar and A. Mirhoseini, "A unified framework for multimodal submodular integrated circuits trojan detection," *IEEE Trans. on Information Forensics and Security*, vol. 6, no. 1, pp. 162–174, 2011.
- [7] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *IEEE Computer*, vol. 43, no. 10, pp. 39–46, 2010.
- [8] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.
- [9] E. Love, Y. Jin, and Y. Makris, "Proof-carrying hardware intellectual property: A pathway to trusted module acquisition," *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 1, pp. 25–40, 2012.
- [10] A. Waksman and S. Sethumadhavan, "Silencing hardware backdoors," *IEEE Symposium on Security and Privacy*, pp. 49–63, 2011.
- [11] M. Hicks, M. Finnicum, S. T. King, M. M. Martin, and J. M. Smith, "Overcoming an untrusted computing base: Detecting and removing malicious hardware automatically," *IEEE Symposium on Security and Privacy*, pp. 159–172, 2010.
- [12] C. Sturton, M. Hicks, D. Wagner, and S. T. King, "Defeating uci: Building stealthy and malicious hardware," *IEEE Symposium on Security and Privacy*, pp. 64–77, 2011.
- [13] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," *IEEE Symposium on Security and Privacy*, pp. 296–310, 2007.
- [14] R. M. Rad, X. Wang, M. Tehranipoor, and J. Plusquellic, "Power supply signal calibration techniques for improving detection resolution to hardware trojans," *IEEE/ACM Intl. Conference on Computer-Aided Design*, pp. 632–639, 2008.
- [15] Y. Jin and Y. Makris, "Hardware trojan detection using path delay fingerprint," *IEEE Intl. Workshop on Hardware-Oriented Security and Trust*, pp. 51–57, 2008.
- [16] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey, "Hardware Trojan horse detection using gate-level characterization," *IEEE/ACM Design Automation Conference*, pp. 688–693, 2009.

- [17] Y. Alkabani and F. Koushanfar, "Consistency-based characterization for IC Trojan detection," *IEEE/ACM Intl. Conference on Computer-Aided Design*, pp. 123–127, 2009.
- [18] K. Hu, A. N. Nowroz, S. Reda, and F. Koushanfar, "High-sensitivity hardware trojan detection using multimodal characterization," in *IEEE Design, Automation & Test in Europe*, pp. 1271–1276, 2013.
- [19] R. Chakraborty and S. Bhunia, "HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493–1502, 2009.
- [20] Y. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," *USENIX Security*, pp. 291–306, 2007.
- [21] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Security analysis of logic obfuscation," *IEEE/ACM Design Automation Conference*, pp. 83–89, 2012.
- [22] A. Baumgarten, A. Tyagi, and J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 66–75, 2010.
- [23] A. Kahng, J. Lach, W. Mangione-Smith, S. Mantik, I. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Watermarking techniques for intellectual property protection," *IEEE/ACM Design Automation Conference*, pp. 776–781, 1998.
- [24] F. Koushanfar, I. Hong, and M. Potkonjak, "Behavioral synthesis techniques for intellectual property protection," *ACM Trans. on Design Automation of Electronic Systems*, vol. 10, no. 3, pp. 523–545, 2005.
- [25] A. Kahng, S. Mantik, I. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Robust IP watermarking methodologies for physical design," *IEEE/ACM Design Automation Conference*, pp. 782–787, 1998.
- [26] J. Lach, W. Mangione-Smith, and M. Potkonjak, "FPGA fingerprinting techniques for protecting intellectual property," *IEEE Custom Integrated Circuits Conference*, pp. 299–302, 1998.
- [27] A. Caldwell, H.-J. Choi, A. Kahng, S. Mantik, M. Potkonjak, G. Qu, and J. Wong, "Effective iterative techniques for fingerprinting design IP," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 2, pp. 208–215, 2004.
- [28] D. Holcomb, W. Burleson, and K. Fu, "Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers," *IEEE Trans. on Computers*, vol. 58, no. 9, pp. 1198–1210, 2009.
- [29] U. Ruhrmair, S. Devadas, and F. Koushanfar, "Security based on physical unclonability and disorder," *Book Chapter in Introduction to Hardware Security and Trust*, 2011.
- [30] F. Koushanfar, G. Qu, and M. Potkonjak, "Intellectual property metering," *Information Hiding Workshop*, pp. 81–95, 2001 2001.
- [31] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, "Logic encryption: A fault analysis perspective," *IEEE Design, Automation & Test in Europe*, pp. 953–958, 2012.
- [32] Chipworks, "Intel's 22-nm Tri-gate Transistors Exposed," <http://www.chipworks.com/blog/technologyblog/2012/04/23/intels-22-nm-tri-gate-transistors-exposed/>, 2012.
- [33] DARPA, "Integrity and Reliability of Integrated Circuits (IRIS)," http://www.darpa.mil/Our_Work/MTO/Programs/Integrity_and_ReliabilityofIntegratedCircuits, 2012.
- [34] Chipworks, "Reverse engineering software." <http://www.chipworks.com/en/technical-competitive-analysis/resources/reerse-engineering-software>.
- [35] Degate. <http://www.degate.org/documentation/>.
- [36] W. Li, Z. Wasson, and S. Seshia, "Reverse engineering circuits using behavioral pattern mining," *IEEE Intl. Symp. on Hardware-Oriented Security and Trust*, pp. 83–88, 2012.
- [37] P. Subramanyan, N. Tsiskaridze, K. Pasricha, D. Reisman, A. Susnea, and S. Malik, "Reverse engineering digital circuits using functional analysis," *IEEE Design Automation & Test in Europe*, 2013.
- [38] W. M. V. Fleet and M. R. Dransfield, "Method of recovering a gate-level netlist from a transistor-level," *US Patent no. 6190433*, 1998.
- [39] J. P. Baukus, L. W. Chow, R. P. Cocchi, P. Ouyang, and B. J. Wang, "Camouflaging a standard cell based integrated circuit," *US Patent no. 8151235*, 2012.
- [40] J. P. Baukus, L. W. Chow, R. P. Cocchi, P. Ouyang, and B. J. Wang, "Building block for a secure CMOS logic cell library," *US Patent no. 8111089*, 2012.
- [41] SypherMedia, "SypherMedia library circuit camouflage technology." <http://www.smi.tv/solutions.htm>.
- [42] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security Analysis of Integrated Circuit Camouflaging," *ACM Conference on Computer Communications and Security*, 2013.
- [43] J. Rajendran, O. Sinanoglu, and R. Karri, "VLSI Testing based Security Metric for IC Camouflaging," *IEEE Intl. Test Conference*, 2013.
- [44] P. Rohatgi, "Improved techniques for side-channel analysis," *Cryptographic Engineering*, pp. 381–406, 2009.
- [45] F. Koene and F.-X. Standaert, "A tutorial on physical security and side-channel attacks," *Foundations of Security Analysis and Design III*, pp. 78–108, 2005.
- [46] P. Rohatgi, "Electromagnetic attacks and countermeasures," *Cryptographic Engineering*, pp. 407–430, 2009.
- [47] A. Schlösser, D. Nedospasov, J. Krämer, S. Orlic, and J.-P. Seifert, "Simple photonic emission analysis of AES," *Journal of Cryptographic Engineering*, pp. 1–13, 2013.
- [48] M. Agrawal, S. Karmakar, D. Saha, and D. Mukhopadhyay, "Scan based side channel attacks on stream ciphers and their counter-measures," *INDOCRYPT 2008*, pp. 226–238, 2008.
- [49] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerer's apprentice guide to fault attacks," *Proc. of the IEEE*, vol. 94, no. 2, pp. 370–382, 2006.
- [50] P. Rakers, L. Connell, T. Collins, and D. Russell, "Secure contactless smartcard ASIC with DPA protection," *Journal of Solid-State Circuits*, vol. 36, no. 3, pp. 559–565, 2001.
- [51] K. Tiri, M. Akmal, and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," *European Solid-State Circuits Conference*, pp. 403–406, 2002.
- [52] S. Moore, R. Anderson, R. Mullins, G. Taylor, and J. J. Fournier, "Balanced self-checking asynchronous logic for smart card applications," *Microprocessors and Microsystems*, vol. 27, no. 9, pp. 421–430, 2003.
- [53] M. Stanojlovic and P. Petkovic, "Strategies Against Side-Channel-Attack," *Small Systems Simulation Symp.*, pp. 86–89, 2010.
- [54] M. Joye, "Basics of side-channel analysis," *Cryptographic Engineering*, pp. 365–380, 2009.
- [55] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011.
- [56] P. C. Kocher, "Leak-resistant cryptographic indexed key update," 2003. US Patent 6,539,092.
- [57] J. Katz and V. Vaikuntanathan, "Signature schemes with bounded leakage resilience," *Advances in Cryptology*, pp. 703–720, 2009.
- [58] Y. Yu, F.-X. Standaert, O. Pereira, and M. Yung, "Practical leakage-resilient pseudorandom generators," *ACM Conference on Computer and communications security*, pp. 141–151, 2010.
- [59] F.-X. Standaert, O. Pereira, Y. Yu, J.-J. Quisquater, M. Yung, and E. Oswald, "Leakage resilient cryptography in practice," *Towards Hardware-Intrinsic Security*, pp. 99–134, 2010.
- [60] J. Demme, R. Martin, A. Waksman, and S. Sethumadhavan, "Side-channel vulnerability factor: a metric for measuring information leakage," *IEEE Intl. Symposium on Computer Architecture*, pp. 106–117, 2012.
- [61] F. Koushanfar, S. Fazzari, C. McCants, W. Bryson, M. Sale, P. Song, and M. Potkonjak, "Can EDA Combat the Rise of Electronic Counterfeiting?," *IEEE/ACM Design Automation Conference*, 2012.
- [62] K. Huang, J. Carulli, and Y. Makris, "Parametric counterfeit IC detection via Support Vector Machines," *IEEE Intl. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, pp. 7–12, 2012.
- [63] X. Zhang, K. Xiao, and M. Tehranipoor, "Path-delay fingerprinting for identification of recovered ICs," *IEEE Intl. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, pp. 13–18, 2012.
- [64] Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak, "Trusted integrated circuits: A nondestructive hidden characteristics extraction approach," in *Information Hiding*, pp. 102 – 117, 2008.
- [65] K. Chatterjee and D. Das, "Semiconductor Manufacturers' Efforts to Improve Trust in the Electronic Part Supply Chain," *IEEE Trans. on Components and Packaging Technologies*, vol. 30, no. 3, pp. 547–549, 2007.