Hardware IP Protection against Confidentiality Attacks and Evolving Role of CAD Tool (Invited Paper)

Swarup Bhunia
University of Florida

Amitabh Das
AMD

Saverio Fazzari
Booz Allen Hamilton

Vivian Kammler
Sandia National Laboratory

David Kehlet
Intel

Jeyavijayan Rajendran
Texas A&M University

Ankur Srivastava
University of Maryland

ABSTRACT

With growing use of hardware intellectual property (IP) based integrated circuits (IC) design and increasing reliance on a globalized supply chain, the threats to confidentiality of hardware IPs have emerged as major security concerns to the IP producers and owners. These threats are diverse, including reverse engineering (RE), piracy, cloning, and extraction of design secrets, and span different phases of electronics life cycle. The academic research community and the semiconductor industry have made significant efforts over the past decade on developing effective methodologies and CAD tools targeted to protect hardware IPs against these threats. These solutions include watermarking, logic locking, obfuscation, camouflaging, split manufacturing, and hardware redaction. This paper focuses on key topics on confidentiality of hardware IPs encompassing the major threats, protection approaches, security analysis, and metrics. It discusses the strengths and limitations of the major solutions in protecting hardware IPs against the confidentiality attacks, and future directions to address the limitations in the modern supply chain ecosystem.

KEYWORDS

IP Protection, Confidentiality, Semiconductor Supply Chain, CAD Tool, Security Analysis, Reverse Engineering, Piracy, Extraction of Design Secrets, Logic Locking, Obfuscation, Metrics

ACM Reference Format:

Swarup Bhunia, Amitabh Das, Saverio Fazzari, Vivian Kammler, David Kehlet, Jeyavijayan Rajendran, and Ankur Srivastava. 2022. Hardware IP Protection against Confidentiality Attacks and Evolving Role of CAD Tool (Invited Paper). In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD '22), October 30-November 3, 2022, San Diego, CA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3508352.3561103

ICCAD '22, October 30-November 3, 2022, San Diego, CA, USA

ACM ISBN 978-1-4503-9217-4/22/10.
https://doi.org/10.1145/3508352.3561103

1 INTRODUCTION

Modern system-on-chip (SoC) designs predominantly rely on pre-verified, reusable hardware intellectual property (IP) blocks, which are interfaced with interconnect fabrics to create SoC designs. These IPs are typically developed through significant efforts and domain expertise and hence, represent carefully-guarded high-value assets for the IP owners. SoC manufacturers often acquire these IPs from third-party IP vendors specialized in development of specific IP classes (e.g., processor, memory, crypto, neural processing engine, etc.). The life cycle of these hardware IPs in today's semiconductor industry is often long, complex, and globally distributed. This is due to the nature of modern supply chain ecosystem that increasingly involves untrusted facilities, people, and tools. Emergence of such an ecosystem for hardware IPs has made them increasingly vulnerable to diverse attacks that compromise their confidentiality and/or integrity. In this paper, we focus on the confidentiality attacks on hardware IPs and their countermeasures.

Figure 1 illustrates the typical hardware IP life cycle. Attacks on IP confidentiality and integrity can be mounted at various stages of this long life cycle, as shown in the figure. A breach in IP confidentiality in an untrusted design house, foundry or in any other stage relates to illegal access, use, or distribution of a design, while a breach in integrity relates to unauthorized alteration of a design for malicious intent. The integrity attacks include insertion of hardware Trojans in a design with the intent to cause malfunction or information leakage [5]. Confidentiality breaches can be very costly to the IP vendors and semiconductor design houses – they can lead to significant loss of revenue to IP vendors or chip designers. Further, for mission-critical applications, such as, defense and communication systems, confidentiality attacks on the electronic components may lead to serious security concerns. Note that confidentiality issues have also been investigated to a great extent in the context of on-chip assets (such as, crypto keys, programmable

ICCAD '22, October 30-November 3, 2022, San Diego, CA, Computer-Aided Design

ACM ISBN 978-1-4503-9217-4/22/10.
https://doi.org/10.1145/3508352.3561103

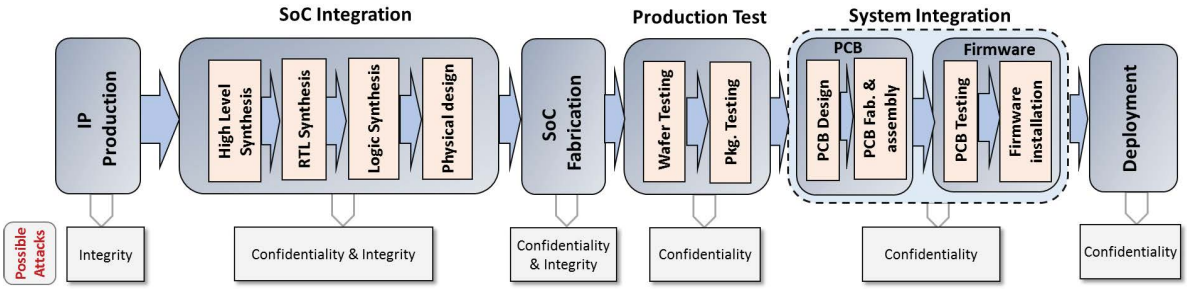

Figure 1: Distributed life cycle of hardware IP blocks in the modern supply chain ecosystem. Possible attack classes for each stage are also shown.

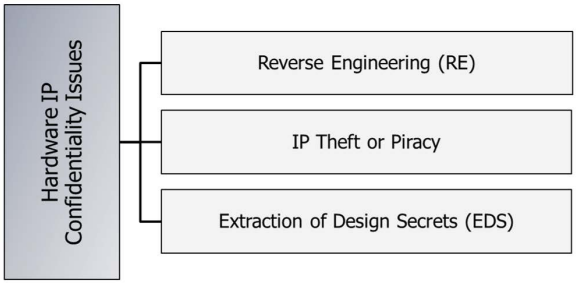

Figure 2: A taxonomy of hardware IP confidentiality issues.

fuses, sensitive data, firmware, etc.) of an SoC, primarily with respect to a software adversary. Generally, confidentiality of these on-chip assets is maintained at various stages of the SoC life cycle through well-designed security policies (e.g., access-control and information flow policies) that are implemented by IP writers or SoC integrators [4]. These confidentiality issues do not fall within the scope of the current article and we will limit our discussion to hardware IP confidentiality in modern supply chain.

The confidentiality attacks on hardware IPs can be put into three broad classes, as shown in Fig. 2: (1) reverse engineering (RE), (2) IP theft or piracy, and (3) extraction of design secrets (EDS). These attacks can be targeted to different abstraction levels (e.g., RTL, netlist, and GDS) as well as on silicon instances of an IP. More sophisticated attacks may cross-cut these classes - e.g., RE of a design can be combined with piracy of specific parts of an IP and similarly, RE can be used to facilitate extraction of secret design information from an IP. Protection of hardware IPs against these attacks has become a critical area of research in the field of hardware security. While watermarking [10] has long been considered as an effective low-cost protection against these attacks, new defense mechanisms, such as logic locking [13], obfuscation [8], and hardware redaction [14] have been studied as promising techniques to provide robust protection against them. These new defenses are poised to defend IP confidentiality under the evolving "zero trust model" that considers most life-cycle stages of an IP as untrusted and assumes "no implicit trust in any one component of a system" [11]. In parallel, a vast body of recent works has presented successful attacks [28] [24] to break many locking and obfuscation methods. These attacks have often been able to retrieve the secret key used in the locking or obfuscation process and restore the original functionality.

Diminishing trust at various stages of IP life cycle has accentuated the need for IP confidentiality in the past decade. Consequently, it has fueled vast amount of research activities towards developing low-cost attack-resistant countermeasures that protect hardware IPs against the confidentiality attacks and integrating them into conventional electronic design automation (EDA) tool flow. These countermeasures are expected to provide mathematically sound, yet practical solutions to the confidentiality issues. Figure 3 provides a taxonomy of IP protection solutions targeted to the confidentiality attacks. Watermarking has been used in practice as a prevalent protection approach to provide "passive" defense - i.e., it cannot prevent an attack - but helps to verify ownership or provenance of an IP if an attack occurs. On the contrary, the "active" defense approaches can potentially prevent these attacks from occurring by making RE and piracy difficult. These active protections are becoming increasingly attractive to the IP producers to complement the passive ones.

Active defense mechanisms against confidentiality attacks fall into four broad classes: logic locking (LL) [13] [2], state space obfuscation [7] [17] [8], hardware redaction [14], and IP encryption [15]. Logic locking (LL) and state space obfuscation have emerged as promising solutions for protecting hardware IP through its life cycle against major attacks. Many variants of obfuscation and LL solutions that aim at protecting an IP against both black-box usage and RE attacks have been explored over the past decade. At the high level, logic locking solutions introduce locking functions at strategic places of a design controlled by bits of a secret key and then perform constrained logic synthesis. While LL solutions have shown promise to provide effective protection against IP piracy and RE in untrusted design and fabrication facilities, researchers have also come up with powerful attacks against LL that compromise the protection. Over the years, these attacks have grown in sophistication and the field of LL has seen a healthy competition in terms of increasingly capable attacks and commensurate defenses.

In this article, we provide in-depth analysis of hardware IP confidentiality issues and the protection mechanisms; discuss how the protections can be integrated into existing design flow; present possible attacks on these protections; point to methods for security evaluation; and discuss future directions. The remainder of the paper is organized as follows. In Section 2, we present an overview

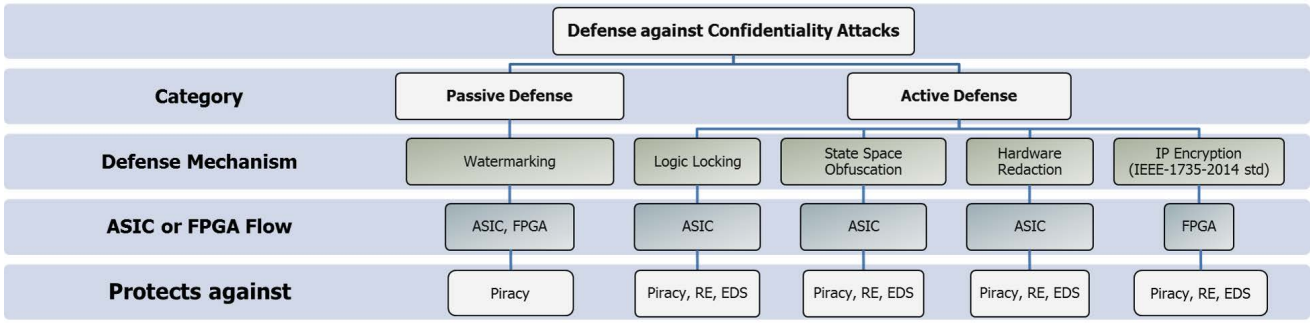

Figure 3: Taxonomy of hardware IP protection approaches against diverse confidentiality attacks.

of the hardware IP confidentiality issues through its life cycle. In Section 3, we discuss the protection approaches to address the confidentiality issues and their integration into commercial tool flow. In Section 4, we present the security analysis and metrics, followed by conclusion and future work on this topic in Section 5.

2 HARDWARE IP CONFIDENTIALITY ISSUES

Hardware IPs are vulnerable to various forms of confidentiality attacks as they flow through the supply chain. Figure 2 illustrates the major classes of confidentiality attacks on hardware IPs, namely piracy, reverse engineering, and extraction of design secrets. For these attacks, physical implementation of a VLSI design produced by an IP owner at register-transfer, netlist or layout (e.g., GDS-II) level is considered as the asset, which is vulnerable to RE or theft. An adversary may want to use the IP description as black-box to create cloned products; perform RE with the intent to copy or alter; or extract critical design secrets from the functional, structural or parametric properties of the design.

Another type of confidentiality attack relates to access violations on the sensitive on-chip assets in an SoC during its in-field operation. In today's complex SoCs, the security implication of interactions between two hardware IPs has become an ever-growing and difficult challenge to solve. Access control at the boundary of the IPs for incoming and outgoing transactions needs to be properly defined and enforced. The trust boundary and all allowed interactions of the IPs need to be clearly enumerated in the architecture and design phases to develop mechanisms aimed at detecting unauthorized access behaviors. If the IPs are of cryptographic nature, secure key storage and retrieval also assume utmost importance to maintain confidentiality of the hardware IPs executing the crypto algorithms. Secure memory access and memory isolation are also important to ensure that IPs are restricted to their own authorized memory ranges and do not access restricted memory regions of other IPs. A centralized security engine in an SoC can set up this memory isolation and monitor memory accesses to ensure that unauthorized memory accesses are prevented and reported.

3 PROTECTION FOR IP CONFIDENTIALITY

There are broadly two classes of techniques to protect hardware IPs against various forms of confidentiality attacks in the modern supply chain, as shown in Fig. 3. As discussed earlier, watermarking techniques form a class of passive protection against confidentiality

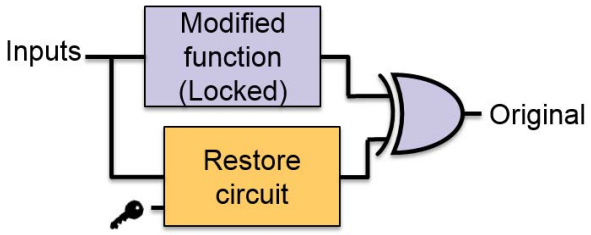

Figure 4: Stripped-functionality logic locking that provides a quantifiable trade-off between security and overhead. Source: [13].

attacks. It relates to insertion of unique identifiers (either functional or parametric) into a design that can be used to prove ownership or perform provenance analysis on hardware IPs. Logic locking, obfuscation, hardware redaction, and encryption of IPs form the second class of active protection techniques. Encryption of IPs relies on vendor-specific toolset and can generally be applied to FPGA devices. Logic locking, obfuscation, and redaction techniques, on the other hand, can be applied to IPs in both ASIC and FPGA design flows. These solutions distinctly differ in the transformation approaches and target components, as described below:

(1) **Logic Locking:** It relates to key-based transformation of the combinational logic that locks a design, making reverse engineering or black-box usage significantly harder.
(2) **Obfuscation (or sequential locking):** It relates to key-based transformation of the state space of a design such that reverse engineering of the embedded finite state machine (FSM) becomes significantly harder.
(3) **Hardware Redaction:** It removes security-critical parts of a design and replaces them with programmable fabrics that can be configured in a trusted facility. Removal of design information helps in preventing confidentiality attacks.
(4) **IP Encryption:** It employs traditional cryptography techniques (e.g., private-key encryption/decryption), to transform and protect soft and firm IPs (e.g., register transfer level or gate-level descriptions) against piracy and RE.

3.1 Watermarking

Watermarking is a technique that embeds authentication "marks" or unique, immutable identifiers into a hardware IP for provenance analysis or ownership verification. It can help protect the rights for IP producers and owners against piracy in both ASIC and FPGA design flow. Generally, watermarks are inserted as hidden functional behavior or as a set of additional constraints during implementation of an IP [10], thereby creating unique functional or parametric signature for a design. These signatures can be verified for authentication purpose at different stages, both pre- and post-silicon, of an IP's life cycle. Such signatures are expected to have the following properties: (1) hard to identify through visual and machine-based inspection, (2) permanently embedded into an IP's functionality or structure, (3) hard to remove and tamper, (4) easy to verify, (5) incurs low cost, and (6) remains invariant to design transformation (e.g., the watermark is not altered when a design is transformed from RTL to logic or logic to layout).

3.2 Logic Locking and State Space Obfuscation

Logic locking inserts additional circuitry along with additional key inputs to perform Boolean algebraic transformation of a combinational logic circuit. The correct functionality is retrieved if the valid key [22, 26] is applied. Thus, logic locking can defend against hardware threats, such as reverse engineering and IP piracy. However, input-output query-based (I/O) attacks [21, 25] and structural attacks [16, 19, 27] can discover the correct key or retrieve the circuit with the unlocked functionality. Over the past few years, the stripped-functionality style of logic locking (SFLL) has been studied as a family of logic locking techniques that provides provable security against I/O attacks and output corruptibility [13]. Figure 4 illustrates the basic concept. Additionally, by carefully selecting the input patterns (i.e., part of the circuit to protect), SFLL can also prevent structural attacks [16]. Furthermore, this technique provides a variety of implementation-friendly styles ranging from XOR/XNOR gates to lookup tables to multiplexors circuits, thereby providing quantifiable trade-offs among security, power, performance, and area overheads. To improve the scalability of a LL approach, researchers have considered partitioning a large design into a set of non-overlapping logic cones using hypergraph partitioning approaches and then applying transformation to each logic cone [2]. Such an approach, illustrated in Fig. 5, can be combined with attack-resistant transformation of logic functions (e.g., SFLL approach [13]), which leads to an efficient CAD solution for protecting against the confidentiality attacks.

State Space Obfuscation techniques aim at transforming the state transition functions of a design within its sequential boundary [7] [17], as illustrated in Fig. 6. Such a transformation is driven by a key, applied as a sequence of primary inputs, which forms the secret a designer will keep to safeguard the confidentiality of an IP. The goal of this transformation is to exponentially increase the reachable state space in a FSM, such that, without access to the secret transformation key, retrieving the original state machine can be practically infeasible. Figure 7 shows an example of state space explosion for an open-source AES-128 design. Such an approach complements the logic locking solutions in its application scope and enhances the overall protection of an IP.

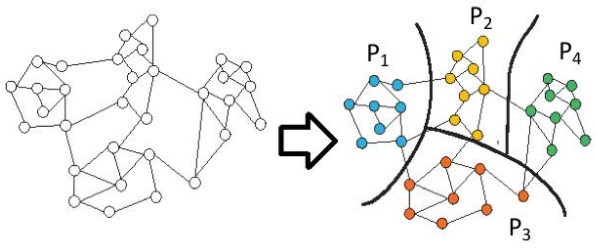

Figure 5: A combinational circuit can be divided into a set of partitions and each partition then can be subjected to Boolean algebraic transformation based on a key to provide scalability and improve security against both functional and structural attacks [2].

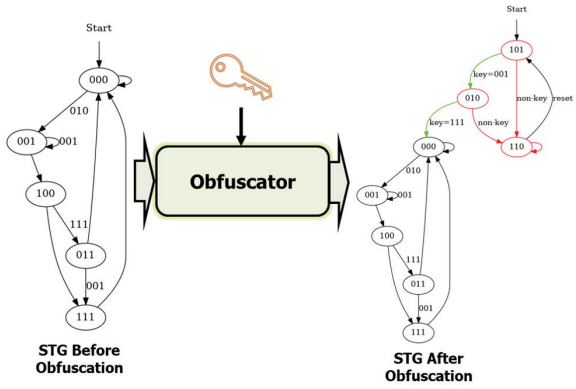

Figure 6: Example of state space obfuscation for a simple finite state machine that performs key-based transformation of its state transition function.

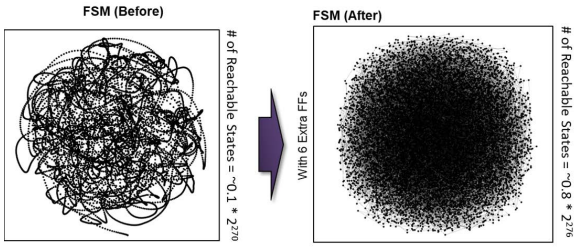

Figure 7: The reachable state space blows up with the state space obfuscation approach, as shown here for an AES-128 benchmark design.

3.3 Hardware Redaction

High-profile data breaches, such as the August 2021 T-Mobile event [3] have highlighted the need for data confidentiality, and hardware technology to support data confidentiality is now available in commercial FPGA devices. In these devices, hardware blocks implementing the Advanced Encryption Standard (AES) can be used to help protect data as it leaves and enters the device. In contrast to

data confidentiality, hardware intellectual property confidentiality is an emerging consideration. The hardware IP at risk belongs to users and developers of semiconductors, a much smaller group than the general public. Still, the impact of hardware IP theft may range from a competitive disadvantage in business to a military disadvantage in armed conflict.

The FPGA is a good solution for protecting hardware IP, especially during device fabrication and testing as the FPGA's program containing the hardware IP to be protected is not present during these manufacturing phases. Yet the size, weight and power (SWAP) advantages of ASICs over FPGAs [18] is driving a desire to achieve protection against hardware IP confidentiality attacks on ASICs. During ASIC manufacturing and test, an IP is vulnerable to an adversary, who we assume has access to: the ASIC physical design (GDSII), netlist, test vectors, simulation testbench, test hardware and state-of-the-art physical analysis equipment and techniques, such as, laser voltage probing. If hardware IP confidentiality of an ASIC design could be achieved, an ASIC developer requiring protection would be able to use any semiconductor fab.

Recent advances in this field have led to the development of fine-grain redaction approaches that remove logic from a design and replace them with lookup tables (LUTs) and a bitstream [1] [14]. Majority of research in this domain has investigated the use of embedded FPGA (eFPGA) fabric in place of the removed logic blocks [14]. An alternative redaction approach focuses on including custom LUTs of various sizes in the standard cell library and using the traditional ASIC synthesis process [1]. These schemes have the security benefit of an FPGA during manufacturing and test in that the bitstream program of the design is not present. When the transformed design is powered up and programmed with its bitstream is the ASIC functional. Such hardware redaction has potential for area and performance advantages over alternative redaction technologies, such as embedded FPGA blocks.

3.4 IP Encryption

In this approach, representation of an IP in hardware description language (e.g., SystemVerilog, Verilog, VHDL) is encrypted by an IP developer to protect it against unintended use during its distribution. The encryption/decryption process generally complies with an IEEE standard, namely IEEE-1735-2014 [15]. IP encryption requires support for vendor-specific toolset and hence is typically used in a FPGA design flow. It protects an IP from the stage of design entry to the bitstream generation. IP developers can express how a FPGA design tool should interact with an IP and thereby manage its access rights. Visibility and capability to modify an encrypted IP is restricted for the downstream users during the synthesis, simulation and bitstream generation steps. While it presents a strong cryptography-based solution to protect IP confidentiality, the requirement for vendor-specific toolset prevents its practical application in ASIC flow.

3.5 Integration into Commercial EDA Flow

The IP protection solutions described above rely on well-defined algorithms for modifying a design. They are all amenable to integration into existing EDA tool flow for ASIC or FPGA. IP encryption and watermarking can be applied to various levels of design abstraction, e.g., register-transfer and netlist level). Logic locking, state space obfuscation and hardware redaction techniques, on the other hand, are typically applied at netlist level, and hence, should be incorporated as a design transformation step following RTL synthesis. The fundamental principles of design transformation in all these methods, however, can be applied to RTL and higher level designs [8]. Any protection on RTL designs needs to be incorporated before the RTL synthesis stage. Since these techniques incorporate modification of a design's functional behavior, there is a need to run functional verification step (based on either simulation or formal equivalence checking) after the transformation.

4 ATTACKS, SECURITY ANALYSIS, METRICS

In this section, we describe possible attacks on protected IPs and how knowledge about them can lead to the security evaluation and quantification process. Figure 8 shows a simple taxonomy of the security evaluation methods. For the above-mentioned defense approaches, in general, we need to consider both functional as well as structural analysis. The research community has developed wide array of functional query based attacks, as well as graph or machine learning based structural analysis attacks. These attacks can lead to systematic evaluation of vulnerability of a protected design against RE and piracy. They can also be used to quantify the level of difficulty to retrieve the secret key (for a key-based transformation) or the original design, leading to the development of effective security metrics. Next, we provide additional insights into major attacks on the protection methods and metrics that quantify security.

4.1 Attack Models

In order to evaluate the severity of the threats and protect the confidentiality of hardware IPs, we need to develop quantifiable measures of the information leaked about the hardware IP. To this end, we need to identify possible sources of information leakage and examine which sources are available to the adversary under each specific attack type. This systematic approach can also be generalized to account for new attack types identified in the future.

An adversary's core objective is to obtain the proprietary design details of an IP. To analyze the worst-case security guarantees, we should assume that the adversary has the gate-level netlist since this is often directly available to the SoC designer and can be reverse-engineered from the layout. Physical RE is required for testing facilities and end-users. Such reverse engineering of an IC is a process of identifying its structure, design, and functionality. RE is a multi-step process involving de-packaging an IC, delayering it, imaging individual layers, and analyzing the collected images to extract the netlist. Multiple companies provide RE service of ICs.

The adversary can also have access to a working IC, which can be, for example, purchased from the market. As this is not true in general, two scenarios where access to a working IC is or is not granted usually need to be discussed when we evaluate IP protection techniques. A working chip serves as an oracle to an adversary, who can apply inputs of his/her choice and observe the correct outputs. The presence of an oracle enables oracle-guided attacks. In some cases, the adversary may have access to the oracle

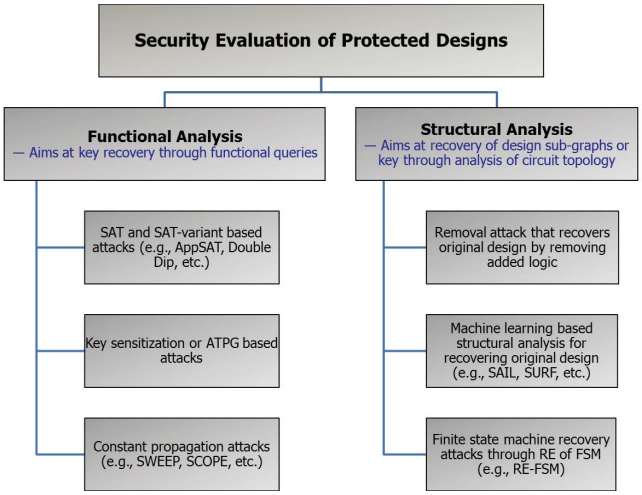

Figure 8: Security evaluation approaches of protected hardware IPs that can provide quantifiable assurance against possible attack vectors.

beyond the inputs and outputs. For example, on-chip test structures, e.g., scan chain, may be accessible. The internal node values may also be probed through various side-channels, subject to accuracy and resolution constraints. The adversary may even be able to inject faults into the chip. If any security guarantee is to be claimed, the details of the adversary's capabilities must be described.

4.2 Salient Attacks on Protected IPs

The design secrets in the protected portions in hardware IPs will not be exposed directly if the IP's gate-level netlist is obtained by the adversary. We roughly classify the attacks into two categories: one with a working chip, the other without a working chip. When a working chip is not available, the adversary must rely on the reverse-engineered of netlist to figure out the correct key/bitstream. In this case, the correlations between these two can indeed reveal lots of information. The Signal Probability Skew (SPS) [28] identifies the AND-tree structures used in protection techniques that force SAT attack to undergo exponentially large number of iterations. Recently, several oracle-less structural attacks targeting XOR/XNOR or MUX-based logic locking techniques have been proposed [12] [29]. The Desynthesis Attack [12] proposes a method to search for a key assignment that maximally retains the circuit structure of the locked design, when the key is synthesized in the netlist. SAIL attack [6] targets XOR/XNOR based locking. They observed a correlation between the key gate type and the structural changes in the surrounding logic, which in turn makes the key value predictable. SnapShot [9] uses machine learning approaches to directly predict key bit values based on locality vectors extracted around key gates. SWEEP [9] identifies functional and structural features associated with correct keys from synthesis reports and provides predictions of individual key bit values. TGA [29] exploits the observation that some circuits contain repetitive sub-circuit structures and uses such sub-circuits without key gates as self-references to reconstruct instances of the same unit function inserted by key gates.

The Boolean Satisfiability (SAT) based attack [25] is the most prominent oracle-guided attack. An adversary needs to create a Boolean SAT formula derived from the gate-level netlist. The formula is then iteratively solved and updated. In each iteration, the search space of the correct key/bitstream is narrowed down, until the correct one is found. SAT attack is a strong mathematical formulation by itself. It has also become the theoretical basis for several newer attacks. For example, AppSAT [23] and DoubleDIP [24] search for approximately correct keys that would inject negligible amounts of error. StatSAT [31] provides a way to obtain keys in approximate computation scenarios. More recently, side-channel analysis [30] and fault injection [20] have been combined with the SAT attack. By expressing the physical measurement results of these attacks in Boolean SAT clauses and integrating the clauses with those in the SAT attack formula, the combined attacks have proven stronger than any of the individual attacks. Discovery of these attacks has led to, on one hand, development of more robust locking/obfuscation methodologies, on the other, more comprehensive security evaluation approaches.

4.3 Security Metrics

CAD tools allow designers to meet their design goals for a target technology. The metrics are essential for the evaluation of CAD techniques that impact any design parameter, including security. It enables a designer to measure his/her ability to achieve the desired goal for an application. Common metrics for design transformation are power, performance, and area (PPA), which allow us to understand the impact of a design choice relative to the specifications. It provides information, which gives confidence that the approach is effective, since PPA typically is used to demonstrate an improvement, which will provide value. Better performance provides opportunities to achieve new features which are needed. Reducing area will lower the cost for fabrication of the design. This allows the CAD developer to demonstrate the impact of their technology. The other area where these metrics are important is compliance with standards or requirements. In a number of cases, designers are required to demonstrate compliance with measured data. For post-fabrication testing, there is a requirement for stuck-at-fault test coverage, and for mission-critical applications, there are requirements to show traceability in a design for compliance. Metrics enable us to quantify our achievement of performance or guarantee compliance. Metrics, however, need to be accepted by the community – both CAD tool developers and the users.

Security poses an interesting challenge when it comes to metrics. The security of the hardware might be the actual design itself (as in the case of logic locking or obfuscation), or protection of the data flowing on the hardware. The hardware security community makes some assumptions about the usage scenario and then generates data to demonstrate results. It is often recognized that there is no "golden" metric for security. Typically, a threat model is defined and the lowest cost solution is created and presented. Acceptance of the threat model considered in a protection approach and quantifying its impact on protecting against the threat has now become the biggest challenge. The usability of the approach must be presented to show the impact on the assumptions and the results. If it produces large overhead for a particular design style, how should that be

handled? It is important to tie the two together so that it becomes clear what might drive acceptance.

To be successful, a metric must be acceptable to the community for usage. Much of the existing work on IP confidentiality has been based on defining a series of threats and identifying a way to prevent them. The issue is that hardware threats are typically harder to formally represent and share unlike their software counterparts. There are established processes in the software space, which allow companies to share threats and fixes. There is a push to leverage standard organizations or the firms, such as MITRE that track and publish threats, for representation of known hardware threats. The good news is that software bugs have a link to the hardware so things can be inferred. This allows people to understand the value of security metrics. There is not, however, a need for that goal in their product development, as it does not impact their systems in most cases. Therefore, the security metrics need a goal that is desired by the community in order to be successful. Since security represents protection of critical assets against known or unknown attacks by an adversary, therefore, the metrics in many cases may not be absolute but be relative.

The common metrics used for overhead analysis are impact on PPA values caused by the design modifications. The metrics for security are generally derived as the difficulty to retrieve the protected information (e.g., the secret key for LL/obfuscation, the bitstream for redaction) from a transformed design or the design itself. The security metrics fall into two broad categories: (1) brute-force attack complexity, and (2) practical attack complexity. The former captures the time-complexity for an adversary to perform worst-case search over the key space (e.g., for an effective key length of n in LL, it is 2^n). The latter captures the robustness of a protection method against more efficient practical attacks that consider access to golden functional behavior (referred to as "oracle"), or structural information of a protected design, and powerful data analysis tools/systems. For example, SAT attacks [25] consider access to oracle and SAT solver tools. The robustness against SAT attacks can be estimated by capturing the number of clauses and the number of Boolean variables in the clauses [2]. Similarly, the SAIL attack assumes access to internal node properties (e.g., fanout, activity, signal probability, etc.) for a protected design. It uses a machine learning tool to retrieve original sub-graphs from the transformed ones. The robustness against SAIL attack can be obtained by computing the average accuracy of sub-graph recovery from a protected design [6].

5 CONCLUSIONS

Protecting the confidentiality of hardware IP blocks against diverse attacks has emerged as a major concern for both IP vendors and SoC designers. The long, distributed, and globalized nature of modern hardware supply chain, which increasingly involves many untrusted parties, bring new challenges to hardware IP protection. In this article, we have presented the needs and challenges associated with hardware IP protection against confidentiality attacks and discussed several promising protection approaches.

IP confidentiality issues are expected to grow and remain as critical concerns in the semiconductor industry. In parallel, EDA companies are expected to integrate design and verification solutions to protect IPs against these attacks through combination of passive (e.g., watermarking) and active (e.g., logic locking, obfuscation, or hardware redaction) techniques. Reduction of the design overhead for the protection approaches and improving their robustness against possible attack vectors will remain as major research directions for the hardware security community. One promising research avenue in this field is the use of machine learning approaches to develop robust protection methodologies, which can resist both known and future attacks by systematically using the evolving knowledge on the attack vectors to guide the design transformation process.

REFERENCES

[1] N. Dorairaj D. Kehlet A. Dasgupta, M.M. Rahman and S. Bhunia. 2022. RIPPER: Securing Hardware IP through Fine Grained Reduction of Boolean Functions. Annual Government Microcircuit Applications Critical Technology (GOMACTech).
[2] A. Alaql and S. Bhunia. 2021. SARO: Scalable Attack-Resistant Logic Locking. *IEEE TIFS* 16 (2021).
[3] B. Barrett. 2021. The T-Mobile Data Breach Is One You Can't Ignore. WIRED.
[4] A. Basak, S. Bhunia, T. E. Tkacik, and S. Ray. 2017. Security Assurance for System-on-Chip Designs With Untrusted IPs. IEEE TIFS.
[5] S. Bhunia, M.S. Hsiao, M. Banga, and S. Narasimhan. 2015. Hardware Trojan attacks: Threat analysis and countermeasures. *Proc. IEEE* 102, 8 (2015), 1229–1247.
[6] P. Chakraborty, J. Cruz, and S. Bhunia. 2018. SAIL: Machine Learning Guided Structural Analysis Attack on Hardware Obfuscation. Asian Hardware Oriented Security and Trust Symposium (AsianHOST).
[7] R. Chakraborty and S. Bhunia. 2009. HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection. *IEEE TCAD* 28, 10 (2009).
[8] R. S. Chakraborty and S. Bhunia. 2010. RTL Hardware IP Protection Using Key-Based Control and Data Flow Obfuscation. Intl. Conf. on VLSI Design.
[9] et. al D. Sisejkovic. 2021. Challenging the security of logic locking schemes in the era of deep learning: A neuroevolutionary approach. ACM JETC.
[10] A. B. Kahng et al. 1998. Watermarking techniques for intellectual property protection. Design Automation Conference (DAC).
[11] D. DiMase et al. 2021. Zero Trust for Hardware Supply Chains: Challenges in Application of Zero Trust Principles to Hardware. NDIA White Paper.
[12] M. E. Massad et al. 2017. Logic locking for secure outsourced chip fabrication: A new attack and provably secure defense mechanism. arXiv:1703.10187.
[13] M. Yasin et. al. 2017. Provably-Secure Logic Locking: From Theory To Practice. *ACM Conference on Computer & Communications Security* (2017), 1601–1618.
[14] P. Mohan et al. 2021. Hardware Redaction via Designer-Directed Fine-Grained eFPGA Insertion. Design Automation and Test in Europe Conference (DATE).
[15] IP Encryption Working Group. 2015. IEEE 1735-2014: IEEE Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP). https://standards.ieee.org/ieee/1735/4358/.
[16] Z. Han, M. Yasin, and J.V. Rajendran. 2021. Does logic locking work with EDA tools? *USENIX Security Symposium* (2021), 1055–1072.
[17] F. Koushanfar. 2021. Provably Secure Sequential Obfuscation for IC Metering and Piracy Avoidance. *IEEE Design Test* 38, 3 (2021), 51–57.
[18] I. Kuon and J. Rose. 2006. Measuring the Gap between FPGAs and ASICs. FPGA.
[19] M. Li, K. Shamsi, T. Meade, Z. Zhao, B. Yu, Y. Jin, and D.Z. Pan. 2016. Provably Secure Camouflaging Strategy for IC Protection. *IEEE/ACM International Conference on Computer-Aided Design* (2016), 28:1–28:8.
[20] N. Limaye, S. Patnaik, and O. Sinanoglu. 2021. Fa-SAT: Fault-aided SAT-based attack on compound logic locking techniques. DATE.
[21] J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri. 2012. Security Analysis of Logic Obfuscation. *IEEE/ACM Design Automation Conference* (2012), 83–89.
[22] J. A. Roy, F. Koushanfar, and I. L. Markov. 2008. EPIC: Ending Piracy of Integrated Circuits. *DATE* (2008), 1069–1074.
[23] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Pan, and Y. Jin. 2017. AppSAT: Approximately deobfuscating integrated circuits. IEEE HOST.
[24] Y. Shen and H. Zhou. 2017. Double dip: Re-evaluating security of logic encryption algorithms. Great Lakes Symposium on VLSI.
[25] P. Subramanyan, S. Ray, and S. Malik. 2015. Evaluating the security of logic encryption algorithms. Hardware Oriented Security and Trust (HOST).
[26] Y. Xie and A. Srivastava. 2019. Anti-SAT: Mitigating SAT Attack on Logic Locking. *IEEE/ACM TCAD* 38, 2 (2019), 199–207.
[27] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran. 2016. Security Analysis of Anti-SAT. *Asia and South Pacific Design Automation Conference* (2016).
[28] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran. 2017. Security analysis of Anti-SAT. Asia and South Pacific Design Automation Conference (ASP-DAC).

[29] Y. Zhang, P. Cui, Z. Zhou, and U. Guin. 2019. TGA: An oracle-less and topology-guided attack on logic locking. ACM ASHES Workshop.
[30] M. Zuzak, Y. Liu, I. McDaniel, and A. Srivastava. 2022. A Combined Logical and Physical Attack on Logic Obfuscation. ICCAD.
[31] M. Zuzak, A. Mondal, and A. Srivastava. 2021. Evaluating the Security of Logic-Locked Probabilistic Circuits. IEEE TCAD.

6 ACKNOWLEDGEMENT

This work was supported in part by DARPA AISS, Intel SHIP, and DARPA SAHARA programs and NSF grants 1822848, 1953285, 2114165, and 2142473.

7 DISCLAIMER

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.