Invited: Independent Verification and Validation of Security-Aware EDA Tools and IP

Benjamin Tan*, Siddharth Garg*, Ramesh Karri*, Yuntao Liu†, Michael Zuzak†, Abhishek Chakraborty†, Ankur Srivastava†, Omid Aramoon†, Qian Xu†, Gang Qu†, Adam Porter‡, Jeno Szep‡, Warren Savage§
*Center for Cybersecurity, New York University, Brooklyn, USA
†Electrical and Computer Engineering, Institute for Systems Research, University of Maryland, College Park, USA
‡Center Mid-Atlantic, Fraunhofer USA, Riverdale, USA
§Applied Research Laboratory for Intelligence and Security, University of Maryland, College Park, USA

***Abstract*—Secure silicon requires seamless integration of new tools, new IP, and design flows to help designers protect integrated circuits from increasingly sophisticated attacks. Independent Validation and Verification (IV&V) of this integrated technology is important to ensure that the tools actually deliver on their security claims when used by independent parties (i.e., people who were not involved in designing the tools). This work discusses the principles and approaches for IV&V of such a complex design environment, including validation of the security strength of the various hardware security techniques, such as combinational and sequential logic locking, Trojan Detection, side-channel mitigation, and blockchain-based asset management. The main challenge in running an IV&V effort is to ensure that the process provides rigorous, methodical, and provable evaluation of the claims of not only the component tools and IP but whether such an integrated environment can produce security-hardened designs by a non-security expert.**

***Index Terms*—Security-aware electronic design automation, hardware security, validation, verification, blockchain, cryptography**

I. Introduction

Security-aware electronic design automation (EDA) flows require modularity and flexibility to integrate myriad security enhancements and intellectual property (IP) to address security requirements. This is necessary for a "defense-in-depth" approach that can better cover the attack surface throughout an integrated circuit's (IC) life-cycle. Hardware security threats such as reverse-engineering, malicious hardware insertion, and side-channel leakage [1] all require mitigation, such as logic locking/obfuscation [2], [3], watermarking [4], and blockchain-based management of supply chain integrity [5]. Furthermore, an aspiration for security-aware EDA tools is the ability for meaningful quantification of—and possibly trade-off between—metrics of security alongside metrics such as power, performance, and area. Non-experts in security should be able to use the flows and relevant security IP easily, so the appraisal of security-aware tool usability is in the scope of independent verification and validation (IV&V).

This work was supported by the Defense Advanced Research Projects Agency (DARPA) Automatic Implementation of Secure Silicon program (AISS) under agreement number HR0011-20-F-0045

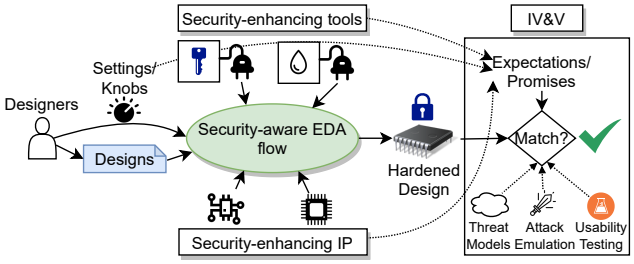

Fig. 1. IV&V examines if implementations match expectations in terms of security guarantees, usability, and correctness, even with designers who are not security experts.

As security techniques mature and development work progresses to transition tools out of the lab into industry, the techniques and their implementations benefit from IV&V. IV&V of security-aware EDA flows is important for increasing confidence that tools and IPs deliver on their security claims when used by independent parties (i.e., by those that were not involved in the implementation of the tools), as illustrated in Figure 1. Moreover, the IV&V process seeks to evaluate the conformance of the tools to overall requirements, the usability of the tools and supporting documentation, and red-team/blue-team characterization of security under various threat models. IV&V requires continual back-and-forth between the developers and IV&V team to establish and refine the scope of evaluations, align schedules, and provide timely, actionable feedback. In this paper, we present a snapshot of the principles and evolving approaches for performing IV&V of security-aware EDA tools, flows, and IP, focusing specifically on examining the implementation of techniques for preventing reverse-engineering, Hardware Trojan (HT) insertion, side-channel attacks, and attacks on the asset management infrastructure (AMI).

II. IV&V of Logic Locking/Obfuscation

Logic locking/obfuscation (henceforth, *locking*) is a set of techniques for mitigating the threat of reverse-engineering. Essentially, given a design $\mathcal{C}(I, O)$, a logic locking technique modifies the design to produce $\mathcal{C}_{obf}(I', O)$ where the design's inputs or state space are expanded such that the design only

works completely as intended after a legitimate user provides the correct key input ($k \subset I'$) [2] or after applying the correct sequence of unlocking inputs [3]. The effect of such techniques is typically measured by metrics for corruptibility (e.g., output corruption for an incorrect key or number of failing compare points in formal equivalence checking) as this captures the (in)ability of an adversary to recover the design's intended functionality. There are a variety of attacks on locking, typically classified as oracle-guided (which assumes access to a functional, unlocked design + scan-chain) or oracle-less (which focus on structural traces/effects of applying the locking technique) [6].

To perform IV&V of security-aware EDA tools incorporating logic locking, we form blue and red teams, as illustrated in Figure 2. The blue team is responsible for independently preparing reference designs (which are free of any inadvertent idiosyncrasies that might exist in in-house testing during development) and pushing them through the security-aware EDA flow, tuning tool parameters to explore the capabilities offered by the EDA tool development team. The red team takes the locked and synthesized design, evaluating their security based on: (i) the run-time of attack tools, (ii) the success rate of retrieving keys/sequences, (iii) the output corruption given partial/guessed keys. The IV&V team reports findings and feedback to the EDA tool team for subsequent iterations and refinement of the security techniques and implementation.

The first step is a preparatory phase; this includes an initial survey of leading attacks on locking approaches (as revealed in prior work [6]) as well as continual scanning for new advances in the domain. After identifying attack techniques and implementations, we select relevant candidates (judged based on the threat models and assumptions made by the EDA tool team) and prepare them for execution in a common environment with known computational capabilities. The common environment provides context on empirical measures such as whether an attack times out. The IV&V red/blue teams also need to discuss and agree on the interface between them, i.e., the format of artifacts shared between teams.

As the security-aware EDA tools become more mature, the IV&V process begins in earnest: first, through a process of *white-box* evaluation. The focus of *white-box* evaluation is to explore a "worst-case" scenario, where an adversary has a lot of detail about the locked design (i.e., collateral such as datasheet information or functional testbenches)—the main goal here is to check that the locking implementations behave as documented by the EDA tool team and to validate/quantify the security of individual modules (e.g., in terms of SAT-attack resilience and corruption in response to guessed/partial keys). IV&V then proceeds to grey- and black-box scenarios, involving artifacts with information increasingly withheld from the red team and, in principle, increasingly representative modeling of an outside adversary (such as untrusted foundry) with limited insider knowledge. In all evaluations, continual feedback is complemented by summative reports with results, an assessment on the "attacker-effort", and any other relevant insights.

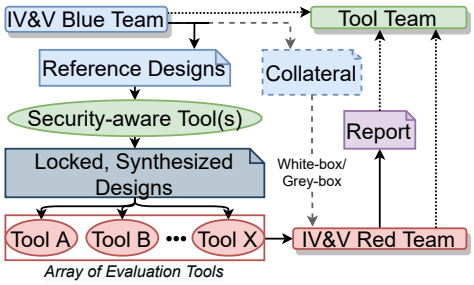

Fig. 2. IV&V of Logic Locking/Obfuscation

III. IV&V of Trojan Protection

HTs can be inserted at various stages in a design's life-cycle by a variety of potential adversaries, including rogue insiders and contractors or an untrusted foundry in the design's fabrication process. HTs are usually capable of malicious activities, such as modifying functionality, leak sensitive information, or cause denial of service. Each HT consists of two parts: a trigger and a payload. The trigger is a node in the circuit that represents the condition to activate the HT's functionality, whereas the payload implements the malicious functionality of the HT.

A security-aware EDA tool should ensure that any HTs that are inserted into a design produced by the EDA tool can be easily detected. The attacker's objectives hence include evading any HT detection technique, which means the HT should not be triggered in the testing process. In addition to regular ATPG-based testing, researchers have proposed many testing techniques for HT detection, including test pattern generation based on simulation [7] and formal methods [8]. It is also desirable for the attacker to know *exactly* which input patterns can trigger the Trojan.

The Trojan IV&V Red Team will emulate an untrusted foundry that tries to insert HTs into the designs produced by the security-aware EDA tool. The objectives of the HT insertion are as follows:

- Able to evade known test-based HT detection techniques.
- HT trigger pattern is known to the attacker.
- HT causes about 50% output pins to flip when triggered.

Based on these objectives, the Red Team will strategically select the location to insert the HT, the trigger pattern of the HT, and the HT's payload. Defenders have full access to the design, including scan chain access, *i.e.,* the ability to write to and read from the internal flip-flops of the chip at test-time. In other words, this allows the defender to reduce the entire design to a collection of combinational sub-circuits which can be tested separately. In light of this, the Red Team will insert the HT into the largest combinational sub-circuit in order not to expose the HT. Notice that the knowledge of each combinational sub-circuit is not available until it is extracted from the full gate-level netlist of the entire design.

Existing Trojan insertion techniques usually use rarely sensitized values of internal nets of the circuit as Trojan trigger [9], [10]. Therefore, Trojan detection techniques that are based on sensitizing these rare values have been developed. To evade

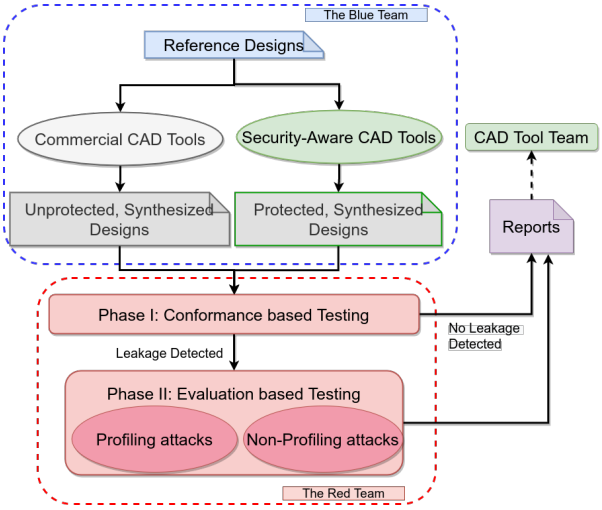

Fig. 3. IV&V flow on side-channel attack mitigations.

such detection techniques, the HT must avoid sensitizing such values.

To achieve desired output corruption (*e.g.,* 50% Hamming Distance), the HT's payload needs to be selected carefully. High output corruption can be achieved by flipping an internal node that impacts a large number of output pins. A simulation-based fault impact analysis can be conducted to determine each node's impact on the output when a stuck-at fault occurs at the node. The result of such fault analysis can be used to guide the selection of Trojan payload.

It is essential for the security-aware EDA tool to produce a design to consider such HT insertion technique and defeat the HT insertion attack by increasing the difficulty of doing so and/or making detection of HTs easier.

IV. IV&V of Side-Channel Mitigation

Side-Channel Attacks (SCA) pose a serious threat to the security of cryptographic systems. They leverage unintended information leakage from the physical implementations to break cryptographic modules. Since the seminal work of Paul Kocher et al. in the late 90s [11], SCAs have shown to be effective in breaking both symmetric and asymmetric algorithms running on a wide range of target devices by leveraging the variations in power consumption, electromagnetic (EM) emissions, time delay, characterizing cache access patterns, and intermediate encryption data in scan chains [12], [13].

A security-aware EDA tool should be capable of equipping hardware and software implementations of in-demand cryptographic cores with arrays of mitigation features against a pre-defined set of SCAs. Moreover, this security-aware EDA tool should be able to characterize and evaluate the security of cryptographic cores against known or potential vulnerabilities.

The IV&V team uses the Blue Team-Red Team approach for the assessment of the security-aware EDA tool's SCA mitigation capability (Figure 3). The Blue Team will prepare a set of reference designs and synthesize them separately with current commercial EDA tools and the security-aware EDA tools. The Red Team takes the synthesized unprotected and protected designs and evaluates the security with respect to the effectiveness of the underlying countermeasures. The Red Team essentially emulates an adversary who has access to the physical implementation of cryptographic cores and is capable of performing side-channel acquisition and analysis. Reports will be sent to the EDA tool team for further refinement of the security features and their implementation.

The IV&V SCA team focuses on power, timing, EM, cache, and scan chain side-channels in this effort. While the Blue Team will act as a user of the security-aware EDA tool, the Red Team will do most of the IV&V job. We outline below a two-phase assessment approach for the Red Team.

In phase I, the Red Team performs conformance-based testing to detect the presence of information leakage and identify leakage points using statistical and information-theoretic tests such as the popular Test Vector Leakage Assessment (TVLA), which uses Welch's t-test to detect the presence of side-channel leakage. Should the design be perceived as leaky, the Red Team will proceed to phase II, in which the feasibility of the state-of-the-art attacks in recovering the cryptographic key will be evaluated by both profiling and non-profiling SCAs.

Profiling attacks correspond to the worst-case security analysis in which a powerful adversary has access to an identical copy or clone of the targeted design. Such SCAs have two stages: a profiling stage during which the attacker utilizes the side-channel information from the clone device to characterize its leakage model; and an attack stage in which the learned leakage model is used to recover the cryptographic key of the targeted device from a limited set of collected side-channel traces. Non-profiling SCA represents the modeling of a resource-constrained adversary who can only obtain side-channel information from the targeted device, which is the more common scenario.

V. IV&V of the AMI

Given numerous threats to the integrity of the hardware supply chain (including threats of cloning, recycling, counterfeiting, etc.), a security solution is needed to carefully manage assets throughout an IC's life-cycle. An Asset Management Infrastructure (AMI) provides a secure and immutable record about a device from creation to salvage. The AMI is implemented as a blockchain-based ledger where assets are registered, cryptographic keys, watermarking data on internal IP cores, manufacturing data, and chain of custody information. The registered items have a complex internal structure, including multi-level encapsulations. The AMI has two major components, the blockchain-based distributed ledger and the REST API. The architecture of the IV&V process is shown in Figure 4.

The efforts of the IV&V team can be grouped into three categories: Design and software validation, Operational and performance testing, and Security testing. The validation process is based on the requirements, and the selection of validation activities, tasks, and work items are reflecting the complexity of the AMI design and the risk associated with the use of the system for the intended use. The operational and

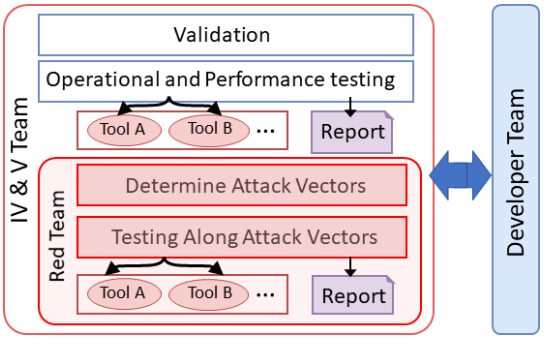

Fig. 4. The architecture of the AMI IV&V process

performance testing requires automation due to the complexity and the expected changes during the further development and deployment phases [14]. The baseline of the Security testing is established with a Risk Analysis that includes vulnerability assessment, a digital attack surface model [15], threat assessment, building a security threat model, and a risk model [16]. Once the possible attack vectors are established, the required toolset will be developed [14].

The effective IV&V process requires continuous communication between the developers and the IV&V team to align understanding of requirements, tune the scope of evaluations, and provide feedback to the developers in a timely manner. Frequent communication between the two teams is beneficial for the developers because it allows corrections in the design and development process at the earliest phase, and it is also beneficial to the IV&V team because it helps to direct the focus of the process at the most critical areas. The IV&V process has a two-phase approach. The first phase covers the interval while the AMI system is designed and developed. During this time, the IV&V team develops an internal prototype, which is a simplified version of the AMI system, and also develops the testing and evaluation toolset. The prototype has both the blockchain and the REST API components designed and developed on the same principles as the AMI. The purpose of the internal prototype is to test and evaluate the testing and evaluation tools. This preparation phase is an enabler for the effective IV&V process once the real AMI system will become available for testing by the IV&V team. The second phase refers to the actual testing and evaluation of the AMI system.

VI. Conclusions

In this work, we provided insights into the principles and processes of an independent verification and validation (IV&V) effort for security-aware EDA tools and IP. We discussed various blue team/red team approaches for investigating different security techniques and described different ways to conceive measures of security and realize (emulate) attacks. Fundamentally, the IV&V process is designed to independently stress-test implementations so that one can reveal any potential shortcomings or make challenges to the assumptions made by security-aware tool and IP designers.

Acknowledgment

This work was supported by the Defense Advanced Research Projects Agency (DARPA) Automatic Implementation of Secure Silicon program (AISS) under agreement number HR0011-20-F-0045. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor. DISTRIBUTION A. Approved for public release: distribution unlimited.

References

[1] M. Rostami, F. Koushanfar, and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," *Proc. IEEE*, vol. 102, no. 8, pp. 1283–1295, Aug. 2014.

[2] N. Limaye, E. Kalligeros, N. Karousos, I. G. Karybali, and O. Sinanoglu, "Thwarting All Logic Locking Attacks: Dishonest Oracle with Truly Random Logic Locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, pp. 1–1, 2020.

[3] R. S. Chakraborty and S. Bhunia, "HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 10, pp. 1493–1502, Oct. 2009.

[4] J. Kuai, J. He, H. Ma, Y. Zhao, Y. Hou, and Y. Jin, "WaLo: Security Primitive Generator for RT-Level Logic Locking and Watermarking," in *2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*. Kolkata, India: IEEE, Dec. 2020, pp. 01–06.

[5] X. Xu, F. Rahman, B. Shakya, A. Vassilev, D. Forte, and M. Tehranipoor, "Electronics Supply Chain Integrity Enabled by Blockchain," *ACM Transactions on Design Automation of Electronic Systems*, vol. 24, no. 3, pp. 31:1–31:25, Jun. 2019.

[6] B. Tan, R. Karri, N. Limaye, A. Sengupta, O. Sinanoglu, M. M. Rahman, S. Bhunia *et al.*, "Benchmarking at the Frontier of Hardware Security: Lessons from Logic Locking," *arXiv:2006.06806 [cs]*, Jun. 2020. [Online]. Available: http://arxiv.org/abs/2006.06806

[7] I. Pomeranz and S. M. Reddy, "A measure of quality for n-detection test sets," *IEEE Trans. Comput.*, vol. 53, no. 11, pp. 1497–1503, 2004.

[8] Y. Lyu and P. Mishra, "Automated trigger activation by repeated maximal clique sampling," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, IEEE. IEEE, 2020, pp. 482–487.

[9] I. H. Abbassi, F. Khalid, S. Rehman, A. M. Kamboh, A. Jantsch, S. Garg, and M. Shafique, "Trojanzero: Switching activity-aware design of undetectable hardware trojans with zero power and area footprint," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 914–919.

[10] J. Cruz, Y. Huang, P. Mishra, and S. Bhunia, "An automated configurable trojan insertion framework for dynamic trust benchmarks," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE. IEEE, 2018, pp. 1598–1603.

[11] P. Kocher, J. Jaffe, B. Jun *et al.*, "Introduction to differential power analysis and related attacks," 1998.

[12] A. Cui, Y. Luo, H. Li, and G. Qu, "Why current secure scan designs fail and how to fix them?" *Integration*, vol. 56, pp. 105–114, 2017.

[13] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing." *IACR Cryptol. ePrint Arch.*, vol. 2005, p. 388, 2005.

[14] A. Parisi, *Securing Blockchain networks like Ethereum and Hyperledger Fabric: learn advanced security configurations and design principles to safeguard Blockchain networks*, 2020, oCLC: 1196206129.

[15] J. Moubarak, E. Filiol, and M. Chamoun, "On blockchain security and relevant attacks," in *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*. Jounieh: IEEE, Apr. 2018, pp. 1–6.

[16] F. Caron, "Blockchain: Identifying Risk on the Road to Distributed Ledgers," *ISACA Journal*, vol. 5, pp. 24–29, Sep. 2017. [Online]. Available: https://www.isaca.org/resources/isaca-journal/issues/2017/volume-5/blockchain-identifying-risk-on-the-road-to-distributed-ledgers