INVITED: The Metric Matters: The Art of Measuring Trust in Electronics

Jonathan Cruz, Prabhat Mishra, and Swarup Bhunia
University of Florida
Gainesville, FL, USA

ABSTRACT

Electronic hardware trust is an emerging concern for all stakeholders in the semiconductor industry. Trust issues in electronic hardware span all stages of its life cycle - from creation of intellectual property (IP) blocks to manufacturing, test and deployment of hardware components and all abstraction levels - from chips to printed circuit boards (PCBs) to systems. The trust issues originate from a horizontal business model that promotes reliance of third-party untrusted facilities, tools, and IPs in the hardware life cycle. Today, designers are tasked with verifying the integrity of third-party IPs before incorporating them into system-on-chip (SoC) designs. Existing trust metric frameworks have limited applicability since they are not comprehensive. They capture only a subset of vulnerabilities such as potential vulnerabilities introduced through design mistakes and CAD tools, or quantify features in a design that target a particular Trojan model. Therefore, current practice uses ad-hoc security analysis of IP cores. In this paper, we propose a vector-based comprehensive coverage metric that quantifies the overall trust of an IP considering both vulnerabilities and direct malicious modifications. We use a variable weighted sum of a design's functional coverage, structural coverage, and asset coverage to assess an IP's integrity. Designers can also effectively use our trust metric to compare the relative trustworthiness of functionally equivalent third-party IPs. To demonstrate the applicability and usefulness of the proposed metric, we utilize our trust metric on Trojan-free and Trojan-inserted variants of an IP. Our results demonstrate that we are able to successfully distinguish between trusted and untrusted IPs.

KEYWORDS

hardware security; metric; information flow tracking;

ACM Reference Format:
Jonathan Cruz, Prabhat Mishra, and Swarup Bhunia. 2019. INVITED: The Metric Matters: The Art of Measuring Trust in Electronics. In *DAC '19: ACM Design Automation Confernece,June 2–6, 2019, Las Vegas, NV*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3316781.3323488

1 INTRODUCTION

To offset the increasing manufacturing costs, System-on-Chip (SoC) designers have adopted a global supply chain. An attacker anywhere along this supply chain can hide hard-to-detect malicious circuitry in third party IP (3PIP) known as hardware Trojans. These malicious insertions can lead to devastating effects such as information leakage or complete circuit malfunction. With the growing reliance on a globalized market, the issue of integrating and evaluating trustworthiness in 3PIP has become a significant concern.

Quantitatively modeling design vulnerabilities has garnered significant research effort to address the lack of comprehensive analysis for trust and integrity issues in hardware IP. With increasingly complex components integrated in an SoC, the task of verifying security properties is typically performed in an ad-hoc manner or by treating the IP as a black box, which can result in unanticipated design vulnerabilities later during the development cycle or after deployment. These integrity issues range from design vulnerabilities introduced by CAD tools to direct malicious modification [3].

In this paper, we propose Trust Coverage, a comprehensive metric for hardware IP trust evaluation, which quantifies a gate-level IP's trustworthiness. The main objective of this metric is to provide a quantitative measure for the trust level in 3PIP by considering multiple *vector-based coverage metrics* and design analyses. These metrics quantify both design vulnerabilities and potential malicious modifications. We then apply a variable weighted scheme according to design specification to provide a final trust value. Our trust metric will enable SoC designers to answer two important design security questions before including 3PIP: (1) *What is the level of functional and structural integrity of a 3PIP? (2) Is one 3PIP more trustworthy than a functionally equivalent one acquired from an alternate vendor?*

Figure 1 elaborates the components of our trust metric. We consider functional, structural, and asset coverage in our analysis. For functional coverage, we define equations for nodal and finite state machine (FSM) coverage analysis. Nodal analysis employs rare node coverage to estimate signal controllability, and FSM analysis identifies potentially malicious or vulnerable don't care states in an FSM. Structural coverage analysis evaluates the coverage for different q-triggered Trojan templates to quantify both controllability and observability, and identifies the existence of potentially suspicious subcomponents in a netlist. Finally, the asset coverage evaluates activated leakage paths over the total number of observable outputs in a design. Analysis of these three parameters provides insight into the overall trust in the IP and the potential presence of functional or leakage Trojans which can be either triggered or always on. The overall trust coverage produces a value $0 \leq T \leq 1$, with the value nearing 1 indicating a high degree of confidence against the possibility of malicious modifications. By including complementary metrics, the inclusion of any Trojan at the IP level will adversely affect at least one metric value reducing the overall trust which is illustrated in the results section.

We make the following major contributions in this paper:

- We propose three vector-based coverage metrics to evaluate several integrity issues present at the IP level. Our framework

DAC '19, June 2–6, 2019, Las Vegas, NV

ACM ISBN 978-1-4503-6725-7/19/06…$15.00
https://doi.org/10.1145/3316781.3323488

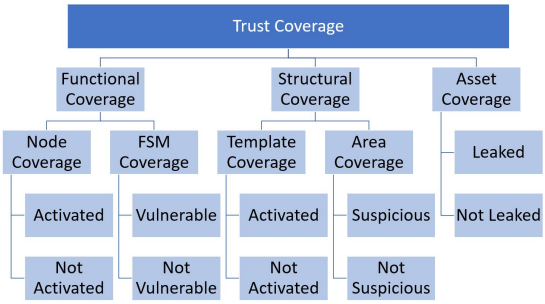

Figure 1: Components in Trust Coverage framework.

would enable seamless integration of other (future) integrity issues.

- We present a framework that employs a weighted sum of these metrics to form a comprehensive metric to quantify overall trust.
- We apply our framework on Trojan-free and Trojan-inserted IPs to evaluate trust and demonstrate the effectiveness of our approach.

The remainder of the paper is organized as follows. Related work on metrics is discussed in Section 2. Section 3 presents our proposed trust metric and describes the significance of its components. The experimental setup and results using Trojan-free and Trojan-inserted benchmarks are presented in Section 4. Finally, Section 5 concludes the paper.

2 RELATED WORK

Recent efforts have been made in quantitatively measuring trust in third-party IPs. A vulnerability measure is proposed in [10]. This approach identifies a unique set of suspicious nets in a design by evaluating power, structural, and timing characteristics of a gate-level netlist. FIGHT metric expands upon FANCI [5] by computing the controllability of internal nets to define trust in a gate-level design[6]. A metric for Trigger and Trojan coverage is first introduced in MERO [8]. Trigger and Trojan coverage are used to estimate the functional coverage of a gate-level netlist by using random sampling of a population of potential Trojans with a trigger activation probability less than a trigger threshold. The method proposed in [9] improves upon MERO by incorporating SAT solving and genetic algorithms to increase coverage at lower threshold values. Saha et al. proposed a metric based on an estimated detectability profile generated from signal probability estimations [13]. This metric quantifies a design's susceptibility to hardware Trojans triggered from a rare conditions observed at multiple nets [20]. However, all aforementioned metrics lack a comprehensive framework for considering all types of Trojans and their varied trigger structure, both triggered and always on.

A framework to quantitatively measure several potential vulnerabilities is proposed in [4]. For gate-level IP, DSeRC separately quantifies hard-to-control and observe nets, vulnerable FSMs, asset leakage and DFT integrity issues using existing metrics. While this work has pushed the need for a comprehensive metric to the forefront, there exist some drawbacks. For controllability and observability metrics, DSeRC uses the metric proposed in [10]. Not all possible potential Trojans may be captured by simply evaluating hard-to-control and hard-to-detect nets as no triggering templates or combination of rare nodes are considered when evaluating firm-IP. Therefore potential malicious insertions may go unnoticed. Because of these drawbacks, the DSeRC framework has limited applicability.

3 TRUST METRIC

To formally quantify integrity issues present in 3PIP, we propose a comprehensive metric that quantifies the overall trust. Table I outlines the types of Trojans at IP level, their effect, and the corresponding metric(s) that will be affected. Our metric framework evaluates three complementary coverage metrics to assess the structural and functional integrity of a 3PIP.

3.1 Functional Coverage

In an IP design, we define rare nodes as nodes with signal probability $\{sp_1, sp_0\} \leq \theta$ where θ is a threshold signal probability value. As mentioned before, rare nodes offer an adversary many opportunities to implant a rarely activated trigger conditions, ultimately reducing the Trojan's detectability. We quantify a test vector's rare node coverage using the following equation:

$$Node\ Coverage = \frac{R_A}{R_T} \tag{1}$$

where R_A, is the number of activated rare nodes, and R_T is the number of total rare nodes. Rare node coverage is important in assessing a design's susceptibility to Trojan insertion. A design with a large percentage of hard-to-control or unactivated rare nodes is generally less trustworthy than a design with a smaller percentage. As an example let us consider two functionally equivalent 3PIP ($D1$ and $D2$) with similar area received from different vendors. If $D1$ reports a higher rare node coverage than $D2$, we can conclude that there is a higher likelihood for the presence of a Trojan in $D2$, because there exist a higher number of uncovered nodes with difficult to control nets an attacker can exploit. Therefore, we can argue that a higher node coverage is associated with a higher trust value. Note, the value of θ is dependent upon the design and can be determined by analyzing random vector or practical workload simulations.

The second component considered in functional coverage is vulnerabilities present in FSMs. CAD tools or DFT structures can introduce or enable unwanted transitions in a protected FSM which an attacker can utilize. To account for such Trojan attacks in an FSM, we provide the following equation similar to [7]:

$$FSM\ Coverage = \frac{V_T - V_X}{V_T} \tag{2}$$

where V_T is the total number of state transitions and V_X is the number of don't care transitions. A higher FSM coverage is associated with a higher IP trust. If an IP has unused states, the CAD tool may insert several don't care states with corresponding transitions providing an attacker with several means to access protected FSM states during IP development or due to modification in field. Consequently, a higher number of don't cares reduces the FSM metric generated from Equation 2, resulting in a lower trust value.

By combining both node coverage and FSM coverage, we summarize the equation for functional coverage as:

$$F = w_n * Node + w_f * FSM \tag{3}$$

A highly trusted design will have a functional coverage of $F = 1$. This component of our trust metric mathematically models rare node coverage and FSM vulnerability addressing vulnerabilities

Table 1: Trojan Types and Integrity Issues

Trojan Type	Trust Issue	Metric
Triggered Combinational [1, 2]	Malfunction, Asset leakage, Parametric payload/Performance degradation	Functional, Structural, Asset
Triggered Sequential [1, 2]	Malfunction, Asset leakage, Parametric payload/Performance degradation	Functional, Structural, Asset
Always On Functional [2, 12]	Asset leakage, Performance degradation	Structural, Asset
Always On Side-Channel [11]	Asset leakage, Parametric payload/Performance degradation	Structural
Power or Temperature [2]	Parametric payload/Performance degradation	Structural

in net controllability and design implict or CAD/DFT introduced unwanted FSM transitions. The weights for individual functional components w_n, w_f can be altered based on the total number of rare nodes and state transitions.

3.2 Structural Coverage

Triggered Trojans are commonly created from a combination of rare nodes in the design shown in Figure 2 (a) & (b) which combines nodes to form an example triggering template[1]. For a q-triggered Trojan with m available rare nodes, the potential trigger conditions in the design can be found from $m!/(q!(m-q)!)$ potential trigger conditions in the design. Functional coverage does not directly account for this security issue as triggers can be constructed from a combination of not-so-rare nodes. Yet, not all combinations of rare nodes can be used as valid trigger conditions. Therefore, we propose the following equation for structural coverage considering up to j valid templates:

$$S = \sum_{i=1}^{j} w_i * TrojCov_i \tag{4}$$

$TrojCov$ is the percent of valid potential q-trigger templates that are both activated and propagated to an observable output when applied to a random payload [8, 9]. The weight w_i, is the ratio of potential q-trigger templates from $templ_i$ over the total number of valid potential q-trigger templates considered in $\{templ_0, templ_j\}$. From our analysis, higher $TrojCov$ values are associated with higher net observability. For example, if the 3-trigger template coverage of 3PIP $D1$ is larger than the 3-trigger template coverage of 3PIP $D2$, then we can conclude the presence of a Trojan in $D1$ is more likely to be activated *and* observed than in $D2$.

However, Equation 4 does not address the structure of always-on leakage or power and temperature Trojans. These Trojans are not triggered from the activation of rare nodes and can display similar structural properties to benign gate characteristics allowing them to evade detection. Oftentimes, an asset K or its complement $\bar{K}$ forms the only connection between the netlist and the always-on leakage Trojan structure. If we consider a netlist as a directed graph structure, always-on Trojan structures are generally $n-edge-connected$ subgraphs were n is the number of assets leaked. A bridge edge between two graphs is an edge when removed disconnects the two graphs. Figure 2(c) & (d) illustrate generic examples for always-on leakage and power Trojans. To address always-on side-channel Trojans, we include connectivity analysis to determine any potentially malicious subgraphs by identifying any asset that forms bridges between subgraphs of the netlist graph representation. Similarly, power and temperature Trojans are loosely connected to the netlist and can be detected using comparable analysis. Any subgraph area

[1]Valid templates can be automatically generated or provided by the designers.

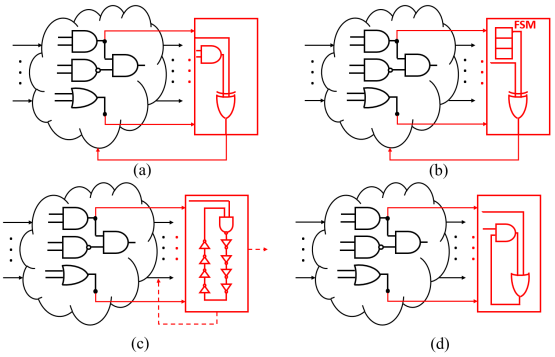

Figure 2: (a) Generic combinational trigger, (b) generic sequential trigger, (c) generic always-on leakage, and (d) generic power/temperature templates.

with a switching activity greater than a provided switching activity threshold is deemed suspicious. We combine this refinement to Equation 5 to form the final equation for structural coverage:

$$S = \left(\sum_{i=1}^{j} w_i * TrojCov_i\right) * \left(\frac{A_T - (A_{asset} + A_{power})}{A_T}\right) \tag{5}$$

where A_T is the total area, A_{asset} is the suspicious area of any subgraph component in which removing an asset edge disconnects it from the main netlist graph, and A_{power} is the suspicious area of any subgraph component with switching activity greater than a threshold value.

Adversaries often try to employ a minimum number of gates when inserting a Trojan to reduce parametric fingerprint and detectability. Attempts to disguise always-on Trojans by adding non-asset signals will impact designs area and power. Therefore, we assume any malicious insertion that is of a significant area or power will be detected through side-channel analysis. Although 3PIP do not have golden reference models, a golden chip free side-channel method can be used to supplement this metric [16].

3.3 Asset Coverage

Asset coverage is the final metric evaluated in our framework. Assets are user-defined internal signals that must remain confidential throughout the execution of the IP. From any given asset, there exist multiple paths to observable points throughout the design. However, a path from an asset K to and observable point O is potentially vulnerable if there exists a test vector which can propagate the value from K to O. We quantify this vulnerability in Equation 6:

$$L = \frac{w_a * (O_T - O_A)}{O_T} \tag{6}$$

O_A is the number of observable outputs that leaked an asset, and O_T is the total number of observable outputs. A weight w_a is an optional variable that can be used to rank the difficulty in leaking of O_A. The optional weighting is calculated by taking the ratio of unused input bits from the test vector over the total number of inputs and multiplying by the number of required vectors. An test vector requiring n input bits to control is harder to leak than an test vector requiring m input bits if $n > m$. By monitoring the number of activated observable points, this component of the metric quantifies how vulnerable a design is to leakage.

We address two integrity issues with asset coverage: 1) vulnerabilities in DFT hardware and CAD tools, and 2) vulnerabilities caused from malicious modification. Event traces can be reviewed to further distinguish if the source of the leakage is due to debug infrastructure or malicious modification. When comparing the asset coverage among 3PIPs, a higher asset coverage is associated with higher trust.

3.4 Aggregate Coverage

To calculate the final trust metric, we apply a weighted sum of the functional, structural, and asset coverages. The comprehensive trust metric is summarized below in Equation 7:

$$Trust = w_1F + w_2S + w_3L \tag{7}$$

where $w_1 + w_2 + w_3 = 1$. The weights in our model are variable as each design has its own security priorities. By default, the weights are assigned values $w_1 = w_2 = 0.25, w_3 = 0.5$. A trust value of 1 from Equation 7 is associated with highly trusted IP and a value of 0 is highly untrusted.

Our framework is largely tool independent. Test vectors are applied on gate level netlists using design and verification tools to extract the information required for analysis. Then after applying the weights, the final trust metric can be calculated. Comparing the resulting trust value for similar benchmarks can aid designers and verification engineers in selecting secure 3PIP and identifying vulnerabilities present at the IP level.

4 RESULTS

We apply the Trust Coverage framework on the arithmetic logic unit (ALU) of the OpenRISC 1200 [18]. The ALU is synthesized using Synopsys Design Compiler. We insert various hard-to-detect random Trojans using the tool in [19] to simulate the scenario in which an SoC integrator procures functionally equivalent IPs from several third-party vendors. The following Trojans are inserted: 1) functional combinational triggered Trojan (combTroj), 2) triggered Trojan that leaks asset through primary output (leakPO), and 3) always on side channel Trojan which leaks asset using ring oscillators (ROleak).

We calculate functional node coverage by applying a modification of MERO to activate rare signals ($\theta = 0.1$). For structural coverage, we again use a modification of MERO to identify valid instances of q-triggered structures, $2 \leq q \leq 4$. The connectivity analysis is performed using graph traversal to identify suspicious subgraph areas. Finally, asset coverage is calculated utilizing Cadence JasperGold. For simplicity, we consider the ALU opcode as a protected asset that should not be leaked. Note, because the ALU does not contain any flip -flops, we weight FSM coverage with 0.

Given the ALU is a purely combinational design, we place highest importance on structural integrity and use the following weight scheme: $w_1 = 0.2, w_2 = 0.6, w_3 = 0.2$, with equal weights elsewhere. From Table 2, we observe the golden design has the highest Trust Value. The low controllability and observability of the Trojan (combTroj) is reflected in both structural and functional coverage. Asset coverage calculations for the triggered leakage Trojan (leakPO) identify additional vulnerable paths to observable points introduced by the Trojan. For the side-channel leakage Trojan (ROleak), the ROs added less than 3% area overhead to the original ALU which is reflected in the connectivity analysis portion of the metric.

Table 2: Comparing Trust in Functionally Equivalent ALU

Benchmark	Func. Cov.	Struct. Cov.	Asset Cov.	Trust Value
golden	0.965	0.130	0.947	0.460
combTroj	0.955	0.120	0.947	0.452
leakPO	0.973	0.120	0.842	0.435
ROleak	0.965	0.120	0.947	0.454

5 CONCLUSION

In this paper, we proposed a comprehensive trust coverage metric by evaluating both structural and functional integrity in 3PIP. Our metrics quantify functional coverage, structural coverage, and asset coverage to address the threats present from a global supply chain. Furthermore, our framework can be used to identify potential design vulnerabilities and compare the trustworthiness of functionally equivalent IPs from different potentially untrusted vendors. We applied our metrics on Trojan-free and Trojan-inserted variants of an IP using state-of-the-art tools to generate the overall trust coverage. Our experimental results demonstrated that the inclusion of a Trojan at the IP level adversely affects one or more metrics reducing the overall trust when compared to a Trojan-free variant.

REFERENCES

[1] M. Tehranipoor and F. Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection", IEEE Design & Test 2010.
[2] S. Bhunia et al., "Hardware Trojan Attacks: Threat Analysis and Countermeasures", IEEE Special Issue on Trustworthy Hardware 2014.
[3] P. Mishra et al., *Hardware IP Security and Trust* Springer, 2016.
[4] K. Xiao, et al., "Security Rule Checking in IC Design", Computer, 2016.
[5] A. Waksman, et al., "FANCI: Identification of stealthy malicious logic using boolean functional analysis", CCS, 2013.
[6] D. Sullivan, et al., "FIGHT-Metric: Functional Identification of Gate-Level Hardware Trustworthiness", DAC, 2014.
[7] A. Nahiyan, et al.,"AVFSM: A Framework for Identifying and Mitigating Vulnerabilities in FSMs", DAC, 2016.
[8] R. Chakraborty et al., "MERO: A Statistical Approach for Hardware Trojan Detection". CHES 2009.
[9] S. Saha et al., "Improved Test Pattern Generation for Hardware Trojan Detection using Genetic Algorithm and Boolean Satisfiability", CHES, 2015.
[10] H. Salmani et al., "On design vulnerability analysis and trust benchmarks development", ICCD 2013.
[11] L. Lang et al., "MOLES: Malicious Off-Chip Leakage Enabled by Side-Channels." ICCAD, 2009.
[12] L. Lin et al., "Trojan Side Channels: Lightweight Hardware Trojans through Side Channel Engineering", CHES, 2009.
[13] S. Saha, et al., "Testability based Metric for Hardware Trojan Vulnerability Assessment", DSD, 2016.
[14] A. Nahiyan,et al., "Hardware Trojan Detection through Information Flow Security Verification", ITC 2017.
[15] G. K. Contreras, et al., "Security Vulnerability Analysis of Design-for-Test Exploits for Asset Protection in SoCs", ASP-DAC, 2017.
[16] T. Hoque, et al., "Golden-Free Hardware Trojan Detection with High Sensitivity Under Process Noise", Journal of Electronic Testing, 2017.
[17] Y. Huang, et al., "MERS: Statistical Test Generation for Side-Channel Analysis Based Trojan Detection", CCS, 2016.
[18] *OpenRISC 1200*, https://github.com/openrisc/or1200
[19] J. Cruz, et al., "An automated configurable Trojan insertion framework for dynamic trust benchmarks", DATE, 2018.
[20] J. Cruz et al., "Hardware Trojan Detection Using ATPG and Model Checking", VLSI Design, 2018.