Behavior Profiling of Power Distribution Networks for Runtime Hardware Trojan Detection

Faiq Khalid*, Syed Rafay Hasan†, Osman Hasan* and Falah Awwadl‡
*Department of Computer Engineering, Vienna University of Technology, Vienna, Austria
†Department of Electrical and Computer Engineering, Tennessee Tech University, Cookeville, TN, USA
‡School of Electrical Engineering & Computer Sciences, National University of Sciences & Technology, Islamabad, Pakistan
§College of Engineering, United Arab Emirates University, Al-Ain, UAE
Email: faiq.khalid@tuwien.ac.at, shasan@tntech.edu, osman.hasan@seecs.nust.edu.pk, f_awwad@uaeu.ac.ae

***Abstract*—Runtime hardware Trojan detection techniques are required in third party IP based SoCs as a last line of defense. Traditional techniques rely on golden data model or exotic signal processing techniques such as utilizing Choas theory or machine learning. Due to cumbersome implementation of such techniques, it is highly impractical to embed them on the hardware, which is a requirement in some mission critical applications. In this paper, we propose a methodology that generates a digital power profile during the manufacturing test phase of the circuit under test. A simple processing mechanism, which requires minimal computation of measured power signals, is proposed. For the proof of concept, we have applied the proposed methodology on a classical Advanced Encryption Standard circuit with 21 available Trojans. The experimental results show that the proposed methodology is able to detect 75% of the intrusions with the potential of implementing the detection mechanism on-chip with minimal overhead compared to the state-of-the-art techniques.**

I. Introduction

With the globalization of integrated-circuit chip design process, the chances of malicious hardware design intrusion, known as hardware Trojan, have grown tremendously [1], [2]. Hardware Trojans can lead to many unwanted activities, including leaking confidential information, changes in the timing characteristics of the circuits, malfunctioning, denial of service and counterfeiting [1]. Some of the prominent hardware Trojan detection techniques include micro-architecture modification to improve triggering of the potential Trojan payload during test [3] and the usage of inherent error detection of quasi delay insensitive (QDI) architectures to detect Trojans [4], [5], [6].

Various power signature techniques have been proposed that require multiple golden circuits to the extract power signatures [7], [8]. Recently researchers have introduced the power ports to produce the multiple power traces without using bulk of the integrated circuits. However, in these methodologies, all power ports should be exited at once but generally Trojans are presented in unknown small regions of integrated circuits. The main challenge of such techniques occurs in identifying the regions to embed the exit power ports [9]. The change in power can affect the other parameters, i.e., frequency and temperature. Based on the same concept, Ferraiuolo et. al. developed a technique that uses the power change effects on the frequency of ring oscillator networks to detect intrusions [10]. Since frequency is highly sensitive to power changes thus, this technique is vulnerable to variation in power supply.

Most of the power analysis based Trojan detection techniques can only detect the Trojans at the test stage but some Trojans may overshadow these techniques. For example, in a SoC design, some hard or firm IPs may hide Trojans depending on the aging of the chip. The possibility of detecting these Trojans during test phase is very low, and they may get activated once the chip is in use [11], [12], [13]. Runtime approaches, on the other hand, could monitor an IC for its entire operational lifetime, providing a last-line of defense. Therefore, specialized techniques have been developed to detect the Trojans at runtime. A promising run-time and low-overhead technique for hardware Trojan detection is temperature tracking. Apart from the correlation between power and temperature, a Trojan can cause a significant variant in the chip's power consumption. Thermal sensors required for detection are already embedded in many chips. A framework for temperature tracking consisting of design-time, test-time, and run-time phases is proposed in [14]. In the design phase, some statistical characteristics of switching activity, power consumption and thermal dynamics are collected and then the thermal sensors are placed. In the test-time phase, a calibration of the given chip due to the process variation takes place. Finally, the information from thermal sensors of the previous phases are used for runtime Trojan detection. Similarly, Bao et. al. have improved the temperature tracking by considering the temperature changes due to power leakage [15] but it requires a precise calibration over the environmental changes and process variations, and also relies on the premise that the triggering of payload will result in a substantially high current flow. Zhao et. al. exploited the dynamic thermal management techniques of integrated circuits to detect Trojans at runtime [16]. A key feature of this technique is to analyze the thermal profile of the integrated circuits to obtain the dynamic thermal/power parameters. The dynamic nature of these parameters is characterized using the Chaos theory, which transforms the dynamic behavior of the real-time signals into the deterministic wavelets (time and frequency) and then classified at runtime using the majority voting machine. This approach inherits the on-chip area and performance overhead of the classification algorithm and majority voting schemes. Similarly, Cao et. al. proposed a technique that uses active current sensors to extract power signatures in the term of delay signature [17]. At the design stage, this approach proposes to divide the IC into several regions, obtains the golden power of each region and designs the runtime monitor which consists of active current sensor, current

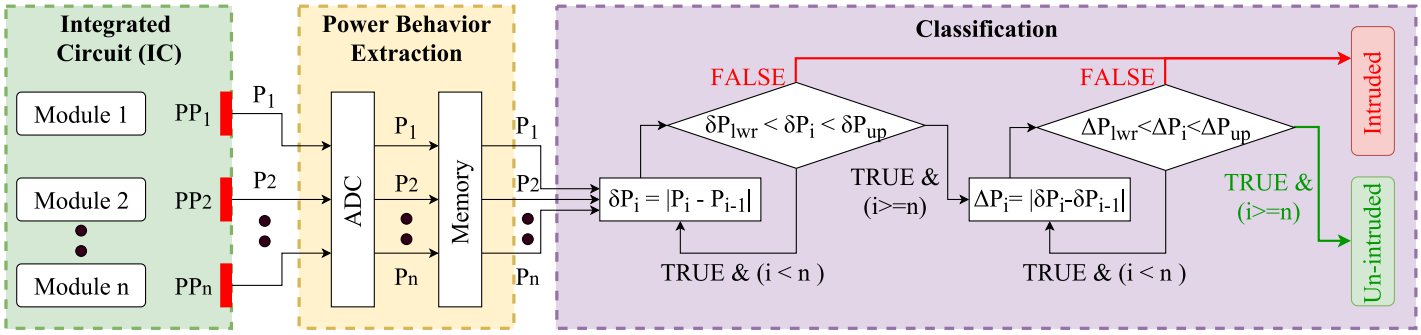


Figure 1: Proposed Methodology for Runtime Power Monitor

comparator and a scan chain to convert current into respective delay. During the runtime, the chip authentication is done by activating the targeted regions, extracting the timing information and then comparing it with the golden signatures. However, translation of power changes into delay may mask the small current changes of chip.

In this paper, we propose a low-overhead design approach, based on runtime current sensors, for security against hardware Trojans. This approach primarily takes advantage of the dynamic supply current analysis for identifying malicious hardware. The proposed approach consists of two main steps: The first step is to obtain the signature power behavior of the given integrated circuit that consists of sequentially connected modules at the pre-market test stage. In the next step, a current sensor measures the supply current for different modules and this information is used to create the modular power profile of integrated circuits at the runtime. These current values are used to obtain the change in current with respect to multiple possibilities of activation of modules. Finally, these changes in current are compared to the power behavior, obtained at the pre-market test stage, to identify the malicious power behavior. The main contributions of this paper are as follows:

1) A power profiling based off-chip runtime power analyzer is proposed for Trojan detection.
2) The computation requirement of the proposed approach is comparatively lower than all the existing state-of-the-art solutions.

II. Proposed Low-Overhead Run-time Power Monitoring

This section explains our proposed methodology for low-overhead runtime hardware Trojan detection through monitoring the modular power behavior of SoCs. This work is based on the premise that the defender is at the SoC integration phase and all the intellectual properties (IPs) provide access to activate different modules. Broadly, we divided our proposed methodology into following two phases.

A. Pre-Market Stage Power Behavior Extraction

The first phase of proposed methodology is to obtain the power behavior of different modules that are sequentially connected by utilizing the following steps: 1) In the first, we measure the power of each sequentially connected modules to extract the modular power behavior. This is obtained at pre-market test stage which is then used to develop the classification criteria. In the proposed methodology, we utilized rate power consumption in different sequentially connected blocks of IP modules to develop a two stage classification criteria. The first stage accepts difference between the power of two sequentially connected blocks:

$$\delta P = |P_i - P_{i-1}| \tag{1}$$

Where, δP is the difference between the power of two sequentially connected blocks (P_i, P_{i-1}). In order to establish this criteria, the first step is to identify the normal behavior zone, which is the range between the maximum and minimum values of δP. Thus, if the value of δP lies outside the normal behavior zone then it is considered as a detected intrusion. However, there are some intruded circuits that can not be detected using the first stage criterion. we have introduced a second stage criterion to increase the precision:

$$\Delta P = |\delta P_i - \delta P_{i-1}| \tag{2}$$

Where, ΔP is the change in difference between the power of the first two sequentially connected blocks (P_i, P_{i-1}). Similarly, its normal behavior zone is the range between the maximum and minimum values of ΔP. Finally, these classification rules are used to design a runtime power monitor. It obtains the current from sensors and generates the power profile at runtime, which is compared to the pre-market test stage power behavior to identify the abnormal power behavior, as shown in Fig. 1.

B. Runtime Power Analysis

In this phase, the runtime power monitors utilize the classification criteria to detect the malicious power activity. Fig. 1 shows the proposed off-chip runtime power monitors, which operates in two steps. In the first step, it obtains runtime power profile by measuring the current from the power ports of each module ($PP_1, PP_2, ..., PP_n$) of integrated circuits. These values are then converted into the respective digital values through an analog to digital converter and then stored into memory. In the second step, the extracted power behavior is used to calculate the changes in current with respect to the sequentially connected modules (first and second derivatives of power behavior) and then compared to the upper and lower bounds of the changes in current from the signature power profile. If it lies outside the bounds then the integrated circuit is considered as intruded, otherwise the IC is considered safe to use.

III. Case Study

Most of the integrated circuit components in modern SoCs are composed of a concatenation of sequentially connected modules. One of the examples is advanced encryption standard (AES) module, which consists of 10 sequentially connected modules. Therefore, to illustrate the effectiveness of the proposed methodology, we used AES modules and analyzed the behavior of some of its benchmark Trojans attack, available on trust-Hub.org [18]. The analysis of AES consists of the following steps:

Table I: Power Behavior (mW) of AES and AES-T100

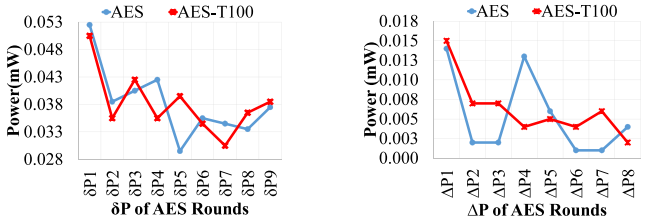
Round	Trojan Free			Trojan AES-T100		
	P	δP	ΔP	P	δP	ΔP
1	0.265			0.336		
2	0.317	0.052		0.386	0.05	
3	0.355	0.038	0.014	0.421	0.035	0.015
4	0.395	0.04	0.002	0.463	0.042	0.007
5	0.437	0.042	0.002	0.498	0.035	0.007
6	0.466	0.029	0.013	0.537	0.039	0.004
7	0.501	0.035	0.006	0.571	0.034	0.005
8	0.535	0.034	0.001	0.601	0.03	0.004
9	0.568	0.033	0.001	0.637	0.036	0.006
10	0.605	0.037	0.004	0.675	0.038	0.002
MAX	**0.605**	**0.052**	**0.014**	**0.675**	**0.039**	**0.015**
MIN	**0.265**	**0.029**	**0.001**	**0.336**	**0.03**	**0.002**

A. Signature obtained for the Power Behavior of AES

The first step is to obtain the signature power behavior of the AES, which is obtained by implementing AES modules without intrusion and with some of the benchmarks intrusions [18] in Verilog. The extraction of signature behavior of AES module is completed in two stages: In the first stage, the AES module is implemented in Verilog and its power rating is extracted using the Xillinx power (Xpower) analyzer for Virtex-5 (xc5vlx330) and Virtex-6 (Xc6vlx760). The highlighted columns of Table I show the dynamic power behavior signature of the AES in Xpower analyzer. In the next step, we calculated the first and second derivative, of the power behavior with respect to the sequentially connected modules using Equations 1 and 2. Table I shows the calculated values of first and second derivatives (δP and ΔP, respectively) of the AES power profile extracted from Xpower analyzer. The pre-market test stage bound values of δP for Xpower analyzer are 0.029 to 0.052 mW, as shown in Table I. Similarly, pre-market test stage bound values of ΔP in Table I for the Xpower analyzer are 0.001 to 0.014 mW. Finally, based on these bounds, we designed the runtime power monitor for AES, as shown in Fig. 1.

IV. Results and Discussion

In order to validate the proposed runtime power analysis based hardware Trojan detection technique, we implemented multiple available AES intrusions [18]. To illustrate the effectiveness of our proposed methodology, we extracted the power profile of the intruded AES using multiple current mirrors. Since the change in current is very minute due to intrusion thus we boosted the current 50 times by taking the width ratio of current mirror equal to 50. However, boosting the current also increases the noise, but this methodology requires the change in current. Therefore, the noise boosting does not effect the change in power or current. The extracted power profile of intruded AES with Trojan AES-T100 is shown in Table I.



(a) Digital δP (mW) (b) Digital ΔP (mW)

Figure 2: Power Behavior of intruded AES and AES-T100

The next step of the proposed technique is to store this profile and for this purpose we converted the analog power profile into respective 16 bit digital power profile using analog to digital converter(ADC) because the change in power is almost 0.1 μW. The digital power profile of intruded AES is shown in Fig. 2. Fig. 2a represents the first derivative (δP) of the digital profile of the intruded AES and its analysis has shown that δP for this intrusion varies form 34 (100010) to 50 (110010), which is within the bound obtained from the generated signature. The effect on power behavior due to this intrusion AES-T100 is very minute and therefore parameter δP is unable to detect it. Therefore, the second derivative of the power behavior is calculated as shown in Fig. 2b and then it is compared with its respective bound obtained from the generated signature. This analysis shows that the value ΔP at point A2 is 15 (001111) μW, which is very well outside the lower bound of ΔP(14 (001110)) μW.

Table II: Effects of AES Benchmark Trojans on Power

Trojan	δP	ΔP	Trojan	δP	ΔP	Trojan	δP	ΔP
T100	No	Yes	**T1600**	Yes	Yes	**T300**	No	No
T1000	Yes	Yes	**T1700**	Yes	Yes	**T400**	No	No
T1100	Yes	Yes	**T1800**	Yes	Yes	**T500**	No	No
T1200	No	No	**T1900**	Yes	Yes	**T600**	Yes	Yes
T1300	Yes	Yes	**T200**	Yes	Yes	**T700**	Yes	Yes
T1400	Yes	Yes	**T2000**	Yes	Yes	**T800**	No	Yes
T1500	No	Yes	**T2100**	No	No	**T900**	No	Yes

Similarly, in order to check the robustness of our proposed methodology, we have analyzed the behavior of the different AES Trojans available on trust-HUB [18]. Table II shows that most of the Trojans can be detected by analyzing the first derivative of power δP. However, there are some intrusions that do not affect the first derivative of power δP therefore we analyzed its second derivative ΔP and Table II shows that it increases the detectable intrusions from 12 to 16 but there are still some Trojans that cannot be detected using this methodology, i.e, T1200, T2100, T300, T400 and T500 because the effect of these intrusions on power is very minimal.

V. Comparison

Table III shows the summary of the comparison with some of the state-of-the-art techniques. This comparative analysis is done using four parameters. The first parameter *Golden IC* means whether the golden IC is required for hardware Trojan detection

Table III: Comparison with state-of-the-art Techniques

Technique	Golden IC	Runtime	On-chip Area Overhead	Constraints
Proposed Methodology	No	Yes	No	1. Vulnerable to environmental and process variations 2. Only applicable to ICs which consists of sequentially connected modules
[14], [15]	Yes	Yes	n Temperature Sensors (n: no of temperature regions)	1. Precise calibration for environmental and process variations 2. High current flow due to triggering payload
[16]	Yes	Yes	Majority voting machine Transformation circuit n Temperature Sensors (n: no of temperature regions)	1. Huge power overhead of classification algorithm
[17]	Yes	Yes	1 current mirror 1 current comparator 1 Scan chain Register	1. Vulnerable to environmental and process variations 2. Only valid if Trojan has significant impact on the switching power

or not. The second parameter *Runtime* considers the online detection of hardware Trojans. Third parameter *On-chip Area overhead* evaluates the on-chip extra components required in case of runtime monitoring. The final parameter is *Constraints* which provides the limitations of a particular technique. The proposed technique is found to be better than other state-of-the-art alternatives in following ways:

1) Unlike [14], [15], [16], [17], our technique does not require the golden IC to extract the power based classification criteria, which is used for runtime Trojan detection.
2) The area overhead of the proposed technique is less than the other techniques as it has no on-chip monitoring setup but it requires an off-chip power monitoring analyzer. However, most of other alternatives have a large area overhead because they require on-chip temperature sensors [14], [15], majority voting machine and transformation circuits [16], and on-chip current mirrors and comparator [17].

Moreover, there are some other techniques [7], [8], [9], which directly measure the power through multiple power ports and manipulate it to detect the intrusions. These techniques may have no area and power overhead but they require multiple golden ICs to extract the golden model.

VI. Conclusion

In this paper, we presented a power profiling based methodology to detect the Trojans at runtime. In the proposed methodology, the modular power is extracted from power ports at runtime to generate a digital power profile, which is then compared with the power signature profile generated during test-phase, to identify the abnormalities. For the proof of concept we have applied it on a classical AES circuit with 21 available intrusions. The experimental results show that proposed methodology is able to detect 75% of the intrusions without any on-chip area overhead.

References

[1] M. Tehranipoor and F. Koushanfar, "A survey of hardware Trojan taxonomy and detection," *Design and Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.

[2] X. Zhang, K. Xiao, M. Tehranipoor, J. Rajendran, and R. Karri, "A study on the effectiveness of Trojan detection techniques using a red team blue team approach," in *VLSI Test Symposium*, 2013, pp. 1–3.

[3] M. Banga and M. Hsiao, "A novel sustained vector technique for the detection of hardware Trojans," in *International Conference on VLSI Design*, 2009, pp. 327–332.

[4] F. K. Lodhi, S. Hasan, O. Hasan, and F. Awwad, "Hardware Trojan detection in soft error tolerant macro synchronous micro asynchronous (msma) pipeline," in *Midwest Symposium onCircuits and Systems*, 2014, pp. 659–662.

[5] F. K. Lodhi, I. Abbasi, F. Khalid, O. Hasan, F. Awwad, and S. R. Hasan, "A self-learning framework to detect the intruded integrated circuits," in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1702–1705.

[6] N. Onizawa, W. Gross, and T. Hanyu, "A low-energy variation-tolerant asynchronous tcam for network intrusion detection systems," in *Asynchronous Circuits and Systems*, 2013, pp. 8–15.

[7] L. W. Wang and H. W. Luo, "A power analysis based approach to detect trojan circuits," in *International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering,*, 2011, pp. 380–384.

[8] L. Wang, H. Xie, and H. Luo, "Malicious circuitry detection using transient power analysis for ic security," in *Quality, Reliability, Risk, Maintenance, and Safety Engineering*, 2013, pp. 1164–1167.

[9] R. Rad, J. Plusquellic, and M. Tehranipoor, "A sensitivity analysis of power signal methods for detecting hardware trojans under real process and environmental conditions," *Transactions on Very Large Scale Integration Systems*, vol. 18, no. 12, pp. 1735–1744, 2010.

[10] A. Ferraiuolo, X. Zhang, and M. Tehranipoor, "Experimental analysis of a ring oscillator network for hardware Trojan detection in a 90nm asic," in *International Conference on Computer-Aided Design*, 2012, pp. 37–42.

[11] S. R. Hasan, S. F. Mossa, C. Perez, and F. Awwad, "Hardware trojans in asynchronous fifo-buffers: From clock domain crossing perspective," in *International Midwest Symposium on Circuits and Systems*, 2015, pp. 1–4.

[12] S. R. Hasan, S. F. Mossa, O. S. A. Elkeelany, and F. Awwad, "Tenacious hardware trojans due to high temperature in middle tiers of 3-d ics," in *International Midwest Symposium on Circuits and Systems*, 2015, pp. 1–4.

[13] F. K. Lodhi, S. R. Hasan, O. Hasan, and F. Awwad, "Power profiling of microcontrollers instruction set for runtime hardware trojans detection without golden circuit models," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2017, pp. 294–297.

[14] D. Forte, C. Bao, and A. Srivastava, "Temperature tracking: An innovative run-time approach for hardware Trojan detection," in *International Conference on Computer-Aided Design*, 2013, pp. 532–539.

[15] C. Bao, D. Forte, and A. Srivastava, "Temperature tracking: Toward robust run-time detection of hardware Trojans," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1577–1585, 2015.

[16] H. Zhao, K. Kwiat, C. Kamhoua, and M. Rodriguez, "Applying chaos theory for runtime hardware trojan detection," in *Computational Intelligence for Security and Defense Applications*, 2015, pp. 1–6.

[17] Y. Cao, C. H. Chang, and S. Chen, "Cluster-based distributed active current timer for hardware trojan detection," in *International Symposium on Circuits and Systems)*, 2013, pp. 1010–1013.

[18] H. Tehranipoor, Mohammad; Salamani, "trust-HUB," url = https://www.trust-hub.org/.