Hardware Trojan Model For Attack And Detection Techniques

Ahmed Aliyu, Abdulaziz Bello, Usman Joda Mohammed, Ibrahim Hussaini Alhassan

Abstract: Today's integrated circuits (ICs) are vulnerable to hardware Trojans, which are malicious alterations to the circuit, either during design or fabrication. The interventions of human in production of Hardware resources have given room for possible modification of hardware components, so as to achieve some malicious aims. This modification help with possible loop holes in the hardware component for later attack. Due to the increase in popularity aim of attacks using embedded Trojan horse programs into chips, attacker are more likely to suppress them with malicious program, also notwithstanding the increase in disintegration of the design and manufacturing process of our microelectronic products (ICs), we should not only concern about inclusion of unplanned, undesirable hardware features ("bugs"), rather about inclusion of planned malicious hardware features: "Trojan Horses," which act as spies or guerrillas. This paper presents a Model of the fundamental attacks and possible detection techniques of Hardware Trojan. The result of the research has shown a great significance in education and for further researches.

Index Terms: Hardware Trojan, integrated circuits (Chips), Electronic design, techniques, Detection, bugs.

1. Introduction

Worldwide integration in the Production of integrated circuit (IC) has exposed numerous vulnerabilities in chip design and fabrication. Many complex ICs today contain multiple Intellectual Property (IP) blocks produced by third party vendors. Just few numbers of ICs were solely developed in-house, which leads to the minimization in likelihood as the circuit becomes more complexly secured. Due to this growth in IP by the third party vendors, an adversary can bargain the virtue of a design during any aspect in the supply chain. An IP block could provide satisfactory functionality for its designed specification but also contain malicious logic. Moreover, malicious logic can be embedded into the circuit such that it remains asleep until activated, but then cannot be forcefully conquered. Hardware threats referring to the three key aspects of information security: *Availability* (Denial-of-Service), *Confidentiality* (Information Leakage), and *Integrity* (Data Altering), must be reduced to ensure trustable formation of chips for required function. This means the chips should perform exactly within the range of its original designed specification (no more and no less). In the past two decades, security researches have focused on both network and information security and how to prevent cyber attacks. However, hardware Trojan Horses cause a deeper breach bypasses upper security layers and threatens all the entire critical infrastructures such as military infrastructure, financial systems and transportation vehicles. Hardware chips are becoming more vulnerable to malicious activities and alterations during both design and manufacturing phases.

- *Ahmed Aliyu and Mohammed Joda Usman are currently pursuing their masters degree program in currently pursuing masters degree program in Computer Science, Liaoning University of Technology , China. E-mail: ahmedaliyu8513@yahoo.com, umjoda@gmail.com*
- *Abdulaziz Bello and Ibrahim Hussaini Alhassan are currently pursuing masters degree program in Electrical Electronic Engineering Technology., Liaoning University of Technology, China. E-mail: abdulaziz.bello69@yahoo.com, ihalhssn@yahoo.com*

In general, hardware Trojans try to bypass or destroy the three major security concerns (CIA) of any system by: leaking confidential information and secret keys covertly to the adversary (Confidentiality attack); changing the value of a certain register (Integrity attack); disabling, deranging or destroying the entire hardware or components of it (Availability attack). Traditional Hardware testing strategies cannot effectively detect Trojans because the probability of triggering hardware Trojan during functional testing is extremely low. Plus, the small Trojan size with respect to chip overall size reduces the Trojan impact on side channels such as static and dynamic power. [1] Hardware Trojans can be a simple modification to the original circuit as shown in Fig. 1; Adversary inserts a simple two input AND gate between the original circuit output and logical one. If Trojan is inactive, circuit will produce its actual output, while if Trojan is triggered and becomes active, the input will logically be zero so circuit produces "Zero" output disregarding its original input value as explained in Equations. It is called SAZ Trojan (Stuck at Zero) as circuit output will stick at "Zero" if Trojan is activated. [1]

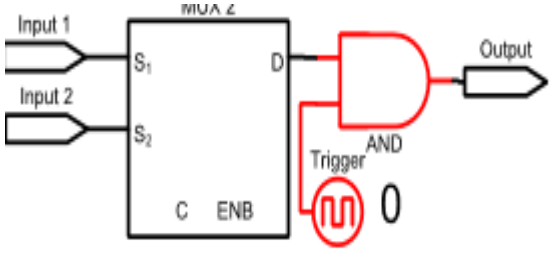

Figure: 1 "SAZ" hardware Trojan.

X.1=X (1)
X.0=0 (2)

A relatively new threat vector to networks and network endpoints is a HT appearing as a physical peripheral device that is designed to interact with the network endpoint using the approved peripheral device's communication protocol. For example, a USB keyboard that hides all malicious processing cycles from the target network endpoint to which it is attached by communicating with the target network endpoint using unintended USB channels. Once sensitive data is ex-filtrated from the target network endpoint to the HT, the HT can process the data and decide what to do with it: store it to

memory for later physical retrieval of the HT or possibly ex-filtrate it to the internet wirelessly or using the compromised network endpoint as a pivot. [2] [3]

2. An Overview

ARCHITECTURE-LEVEL TROJAN DETECTION

Majority voting technique can be used for protection with no need for a fully trusted chip as shown in Fig. 2.1. [1] H.A.M. Amin et al. aimed at producing a Trojan free output from infected IP cores. They used voting techniques for the output of odd number of multi- vendor IP cores trying to achieve negligible probability of infected output and report the infected IP core. Although the use of simple majority voting was suggested in other papers by Waksman and Sethumadhavan [4], it was not thoroughly evaluated using hardware implementation. H.A.M. Amin et al. also evaluated the protection method based on the probability of Trojans detection, probability of false positives, and probability of false negatives and also suggest an advanced voting technique based on giving a higher voting weight for trusted IP cores and they evaluated both the security properties and hardware overhead of both voting methods. Hardware overhead here means circuit area, circuit delay and Leaked power.

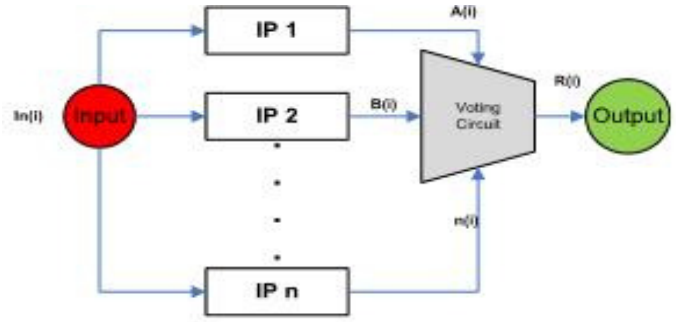

Figure: 2.1 Majority Voting Technique.

Hardware chips fabrication process contains two major steps: design (including IP, models, tools, and designers); and fabrication (including mask generation and packaging). In an ASIC design process, the IP core blocks and standard model cells which are used by the designer during the design process are considered untrusted, also hardware fabrication step may be considered untrusted because an attacker may replace Trojan logic for original ones or inject a Trojan into chip silicon mask. The attacker is assumed to alter the design maliciously before or during fabrication, and detecting these alterations is extremely difficult, as detecting small malicious alteration is extremely harsh in today's high complex IP cores. Nano-meter physical inspection is very sophisticated and costs a lot. Trojans are activated under rare conditions so normal function testing is not sufficient to detect them. It is mandatory to provide methods that resolve the trust issues among fabrication facilities, designers, and end users. Designers need to assure that their designs are not altered while maintaining fabrication facilities technology secrets and third party IP core design properties. [1] Verbauwhede and Schaumont delved into trust issues at different levels of design abstraction (circuits, software, microarchitecture, and protocols). [5] At the most abstract level, the adversary can access the interpreter and perform scan-chain readout, software tempering or a fault attack. It is possible to use Side-channel information at the software-architecture level. At the hardware microarchitecture and circuit levels, the attacker takes into record power energy consumption or electromagnetic energy. Therefore, the authors anticipated a systematic countermeasure to protect the root of trust at different design abstractions. Tamper-proof techniques such as placing security parts into special casing with light, temperature, tampering, or motion sensors can provide protection at the physical level. Side-channel information such as power consumption should be separated from processing data or execution time to provide circuit level protection. To deal with power fluctuation, different technologies such as full-custom dynamic and differential logic styles should be used. In experiments conducted by the authors, advanced encryption standards employing wave dynamic and differential logic remained safely after 1.5 million power-differential attack measurements, whereas standard CMOS technology disclosed the key only after 2,000 attack measurements. To deal with side-channel attacks at the microarchitecture level, Verbauwhede and Schaumont suggested balancing if-and-else instructions to use the same amount of time and power during execution. The structure of microprocessors providing potential sources of side-channel information should be considered seriously. The authors also suggested using secure algorithm techniques, such as key and exponent blinding, to disable side-channel attacks at lower levels. [6] Suh, Deng, and Chan proposed authenticating the hardware by directly checking its implementation details at a low level. [7] The micro architecture features of a high-end secure microprocessor are complex and unique for each model. A secure processor is authenticated by a checksum response to a challenge within a time limit. The unique checksum is based on the cycle-to-cycle activities of the processor's specific internal microarchitectural mechanism. Privacy is not breached, because the checksum depends on the processor-manufactured model and not the specific processor. The authors showed that small differences in the crypto-architecture result in significant deviations in the checksum. Their work relied on the speed advantages of the actual processor rather than simulations that attempt to impersonate the processor. The time limit on the authentication ensures resiliency against simulation models attempting to compute the checksum. [6] Bloom, Narahari, and Simha introduced a runtime Trojan activity detection mechanism using a hardware guard circuit and operating-system support. [8] Trojan attacks can either be internally or externally activated, and they can cause denial of service, privilege escalation, or leakage of sensitive information. Trojans can be detected by failure analysis and hardware verification, ATPG, or side-channel analysis. Bloom, Narahari, and Simha's work concentrated on denial-of-service (DoS) and privilege escalation attacks. [8] They used a hardware guard circuit to efficiently perform the testing, while the operating system generated the checks. Their hardware circuit included a timer, a scratch RAM, a simple processor, and an optional content-addressable memory (CAM). Two tests were proposed: liveness checks and memory protection checks. Liveness checks are pseudorandom noncached-memory accesses that prevent simple prediction, delay, and replay attacks. Two solutions were provided for memory protection: a naïve solution and a solution using a real-time operating system (RTOS). The naive solution periodically schedules a process that continuously tries to read the kernel memory. However, the process is time-

consuming.

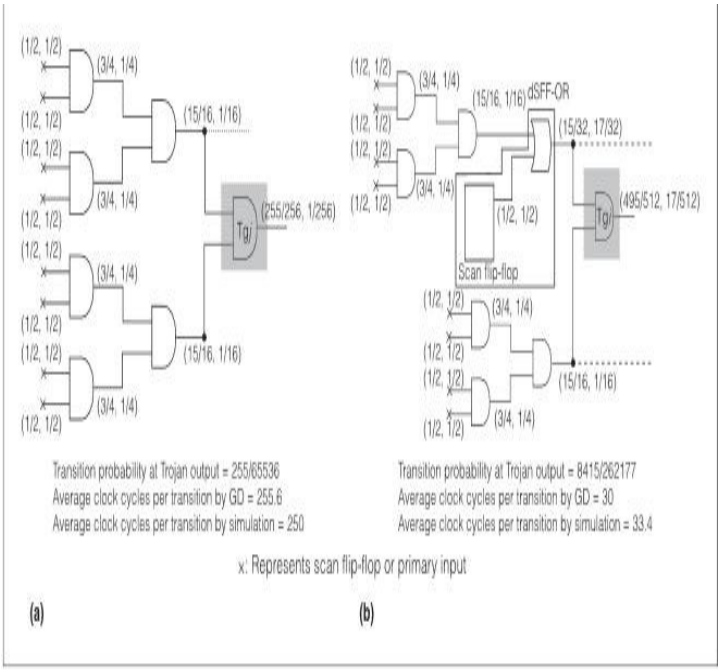

Figure 2.2: Analyzing Transition Probability in the Original Circuit (A) And After Dummy Scan Flip-Flop Insertion (B). (Source: Salmani et Al. [10])

RTOS support is needed to control the time of the checking process, which is created as a real-time task that is frequently required and consumes less time. The proposed solutions are evaluated on SPEC it 2006 benchmarks. The overhead for using RTOS support is approximately 2.2%. McIntyre et al. used hardware multicore systems, which permit simultaneous execution of the same functionality combined with verification. [9] Multicore systems are inherently redundant. Thus, as trust detection among the multiple cores is discovered, distributed software scheduling could be exploited to avoid low-trust cores. The distributed multicore task scheduler determines, over time and in the field, each core's hardware trust level.

POWER-BASED ANALYSIS

Agrawal et al. were the first to use side-channel information to detect Trojan contributions to circuit power consumption [11]. To obtain the power signature of Trojan-free (i.e., genuine) ICs, random patterns are applied and power measurement is performed. The data belonging to each power measurement consists of several elements, including power consumption of the circuit after applying inputs that are the same in all Trojan-free ICs; measurement noise, which can be removed by several measurements; process variations, which are random and cannot be removed; and Trojan contributions to the measured power consumption. After patterns are applied, a limited number of ICs are reverse engineered to ensure they are Trojan free. Once the reference signature is obtained, the same random patterns are applied to the IC under authentication (IUA). If the IUA's power signature differs from the reference signature, the IUA is considered suspicious and that it might contain a Trojan. Trojans of different sizes under different process variations are detected by applying random patterns and observing the signatures. If the Trojan is comparable in size with the circuit, its impact on the circuit-transient current will be significant and could be measured easily. However, process variations will mask the impact of very small Trojans on circuit power consumption. [6]

3. OUR MODEL ANALYSIS

In a given set of Integrated Circuit (chips) S_{IC} on a hardware, if there is an implant of Hardware Trojan H_T , hence, there exist changes; change in power, delay in time of processing and increase in memory size due to Hardware Trojan execution and processing in the integrated circuit. The following is deduced;

$$S_{IC} = \{H_T: \setminus \Delta C_P \cap \Delta T_D \cap \Delta M_S/I_M\} > \beta \qquad \text{where is } \beta > 0.$$

Once the above model occur, then there exist a tendency of Hardware Trojan exist in the integrated circuits. Trojan detection using side-channel signal analysis, there are namely; power characteristics, Timing (delay) and memory space occupied by Hardware Trojan. For timing a graph that signifies the change in power (current) due to presence of Trojan horse.

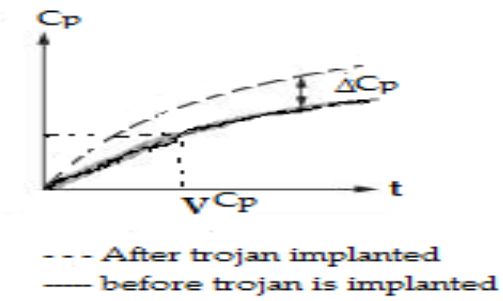

Figure 3.1 Current Variation

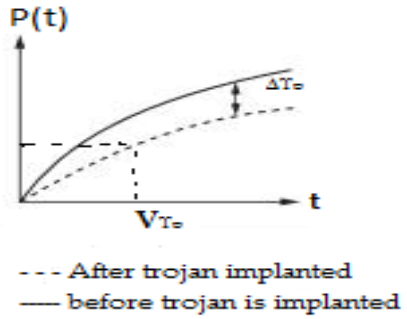

Figure 3.2 Time Delay Variation

Memory space; once a Hardware Trojan is in existence within the hardware chips it requires memory (cache) for execution.

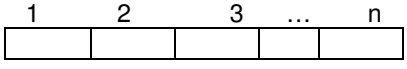

Fig 3.3 Memory location

The length of memory is 1…n, there is length increases after attack, which can be written as 1…n+m, where m is the change or increment in memory size (M_S) or memory consumption due to existence of Hardware Trojan. Even though some authors have argued extensively that the amount

memory, current and delay is very negligible but still hardware Trojan can be detected using the parameters stated.

5 CONCLUSIONS

The work in this paper has provided an overview of Hardware Trojan horse attacks and detection techniques. These attacks are carried-out due to design fault which may be intentional or unintentional. Through extensive review of several previous research papers, we have demonstrated that using this Model of hardware Trojan horse attack and detection techniques will increase or improve the attacks and detection sensitivity and understanding. To improve the sensitivity further, the paper only focuses on the Model of attacks and detection techniques. Further work can be done on automation/simulation of the attacks and detection Modeled system.

6 Acknowledgments

I wish to thanks my colleagues' for their kindness support and encouragement on this paper. More to this my sincere regards goes to the staff of the faculty of Electrical Electronic Engineering technology.

REFERENCES

[1]. Hany A.M. Amin, Yousra Alkabani and Gamal M.I. Selim. "System-level protection and hardware Trojan detection using weighted voting "Cairo University, journal of advanced research 2013, pp.1-7.

[2]. J. Clark, S. Leblanc, S. Knight, "Compromise through USB-based Hardware Trojan device, Future Generation Computer Systems" (2010) (In Press). dx.doi.org/10.1016/j.future.2010.04.008.

[3]. John Clark, Sylvain Leblanc, Scott Knight, "Hardware Trojan Device Based on Unintended USB Channels," Network and System Security, International Conference on, pp. 1-8, 2009 Third International Conference on Network and System Security,2009doi.ieeecomputersociety.org/10.1109/NSS.2009.48.

[4]. Waksman A, Sethumadhavan S. Silencing hardware backdoors. In: Proceedings of the 2011 IEEE symposium on security and privacy, SP '11. IEEE Computer Society; 2011. p. 49–63.

[5]. I. Verbauwhede and P. Schaumont, "Design Methods for Security and Trust," Proc. Design, Automation and Test in Europe Conf. (DATE 07), EDA Consortium, pp. 672-677.

[6]. Mohammad Tehranipoor and Farinaz Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection" IEEE Design & Test of Computers, 2010, pp. 10-25.

[7]. G.E. Suh, D. Deng, and A. Chan, "Hardware Authentication Leveraging Performance Limits in Detailed Simulations and Emulations," Proc. 46th Design Automation Conf. (DAC 09), ACM Press, 2009, pp. 682-687.

[8]. G. Bloom, B. Narahari, and R. Simha, "OS Support for Detecting Trojan Circuit Attacks," Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust (HOST 09), IEEE CS Press, 2009, pp. 100-103.

[9]. D. McIntyre et al., "Dynamic Evaluation of Hardware Trust," Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust (HOST 09), IEEE CS Press, 2009, pp. 108-111.

[10]. H. Salmani, M. Tehranipoor, and J. Plusquellic, "New Design Strategy for Improving Hardware Trojan Detection and Reducing Trojan Activation Time," Proc. IEEE Workshop Hardware-Oriented Security and Trust (HOST 09), IEEE CS Press, 2009, pp. 66-73.

[11]. D. Agrawal et al., "Trojan Detection Using IC Fingerprinting,"Proc. IEEE Symp. Security and Privacy (SP 07), IEEE CS Press, 2007, pp. 296-310.